

<b>Document Title</b>	Specification of RESTful communication
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	876

<b>Document Status</b>	Final
<b>Part of AUTOSAR Standard</b>	Adaptive Platform
<b>Part of Standard Release</b>	18-10

<b>Document Change History</b>			
<b>Date</b>	<b>Release</b>	<b>Changed by</b>	<b>Description</b>
2018-10-31	18-10	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Updated APIs to use ara::core types</li> <li>• Minor editorial fixes</li> </ul>
2018-03-29	18-03	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Added HTTP/JSON network binding</li> <li>• Added support for payload compression</li> <li>• Adapted Event API</li> <li>• Added support for binary data</li> <li>• Minor extensions on API (e.g. helper functions)</li> </ul>
2017-10-27	17-10	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Initial release</li> </ul>

## Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

# Table of Contents

1	Introduction and functional overview	14
2	Acronyms and Abbreviations	15
3	Related documentation	16
3.1	Input documents	16
3.2	Related standards and norms	16
3.3	Related specification	16
4	Constraints and assumptions	17
4.1	Limitations	17
4.2	Applicability to car domains	17
5	Dependencies to other functional clusters	18
6	Requirements Tracing	19
7	Functional specification	35
7.1	General description	35
7.1.1	Architectural concepts	35
7.1.2	Design Scope	37
7.1.3	Design objectives	37
7.1.4	Basic Components	38
7.2	Support Functionality	39
7.3	URI	41
7.4	UUID	42
7.5	Endpoints	42
7.6	Client	45
7.7	Server	47
7.8	Routing	51
7.8.1	Patterns	51
7.8.2	Match	52
7.8.3	Matches	52
7.8.4	Route	52
7.8.5	Router	53
7.9	Object Graph Model	54
7.10	Network binding	56
7.10.1	Transport protocol	56
7.10.2	Serialization of payload	60
8	API specification	62
8.1	ara::rest::Allocator	62
8.1.1	Allocator	62
8.1.2	~Allocator	62
8.1.3	allocate	63

8.1.4	deallocate	63
8.1.5	is_equal	63
8.2	ara::rest::Client	64
8.2.1	NotificationHandlerType	64
8.2.2	SubscriptionStateHandlerType	64
8.2.3	Client	65
8.2.4	Client	65
8.2.5	operator=	66
8.2.6	Stop	66
8.2.7	Send	66
8.2.8	Subscribe	67
8.2.9	GetError	67
8.2.10	ObserveError	68
8.3	ara::rest::Event	68
8.3.1	Event	68
8.3.2	operator=	69
8.3.3	Unsubscribe	69
8.3.4	Resubscribe	70
8.3.5	GetUri	70
8.3.6	GetSubscriptionState	70
8.3.7	operator==	71
8.3.8	operator!=	71
8.3.9	operator<	72
8.4	ara::rest::IteratorRange	72
8.4.1	Iterator	72
8.4.2	IteratorRange	72
8.4.3	begin	73
8.4.4	end	73
8.4.5	begin	74
8.4.6	end	74
8.5	ara::rest::MoveIteratorRange	74
8.5.1	MoveIterator	75
8.5.2	MoveIteratorRange	75
8.5.3	begin	75
8.5.4	end	76
8.5.5	begin	76
8.5.6	end	76
8.6	ara::rest::Matches	77
8.6.1	MatchRange	77
8.6.2	Count	77
8.6.3	Get	78
8.6.4	Get	78
8.7	ara::rest::Match	79
8.7.1	Get	79
8.7.2	GetAs	79
8.8	ara::rest::ogm::Array	80

8.8.1	SelfType	80
8.8.2	ParentType	80
8.8.3	Iterator	80
8.8.4	ConstIterator	81
8.8.5	ValueRange	81
8.8.6	ConstValueRange	82
8.8.7	MoveRange	82
8.8.8	GetParent	82
8.8.9	GetParent	83
8.8.10	HasParent	83
8.8.11	GetSize	83
8.8.12	IsEmpty	84
8.8.13	GetValue	84
8.8.14	GetValue	85
8.8.15	GetValues	85
8.8.16	GetValues	85
8.8.17	Append	86
8.8.18	Insert	86
8.8.19	Remove	87
8.8.20	Release	87
8.8.21	Replace	88
8.8.22	Clear	88
8.8.23	Make	89
8.8.24	Make	89
8.8.25	Array	89
8.8.26	Array	90
8.9	ara::rest::ogm::Field	90
8.9.1	SelfType	91
8.9.2	ParentType	91
8.9.3	GetParent	91
8.9.4	GetParent	92
8.9.5	HasParent	92
8.9.6	GetName	92
8.9.7	GetValue	93
8.9.8	GetValue	93
8.9.9	SetValue	94
8.9.10	ReplaceValue	94
8.9.11	Make	94
8.9.12	Make	95
8.9.13	Field	95
8.9.14	Field	96
8.10	ara::rest::ogm::Int	96
8.10.1	SelfType	96
8.10.2	ParentType	97
8.10.3	ValueType	97
8.10.4	GetParent	97

8.10.5	GetParent	98
8.10.6	HasParent	98
8.10.7	GetValue	99
8.10.8	SetValue	99
8.10.9	Make	99
8.10.10	Make	100
8.10.11	Int	100
8.11	ara::rest::ogm::Node	101
8.11.1	SelfType	101
8.11.2	ParentType	101
8.11.3	GetParent	102
8.11.4	GetParent	102
8.11.5	HasParent	102
8.11.6	~Node	103
8.11.7	Node	103
8.11.8	operator=	104
8.11.9	GetAllocator	104
8.11.10	GetAllocator	104
8.11.11	Node	105
8.12	ara::rest::ogm::Object	105
8.12.1	SelfType	105
8.12.2	ParentType	106
8.12.3	Iterator	106
8.12.4	ConstIterator	106
8.12.5	FieldRange	107
8.12.6	ConstFieldRange	107
8.12.7	MoveFieldRange	108
8.12.8	GetParent	108
8.12.9	GetParent	108
8.12.10	HasParent	109
8.12.11	GetSize	109
8.12.12	IsEmpty	109
8.12.13	GetFields	110
8.12.14	GetFields	110
8.12.15	HasField	111
8.12.16	Find	111
8.12.17	Find	111
8.12.18	Insert	112
8.12.19	Remove	112
8.12.20	Release	113
8.12.21	Replace	113
8.12.22	Clear	114
8.12.23	Make	114
8.12.24	Make	115
8.12.25	Object	115
8.12.26	Object	116

8.13	ara::rest::ogm::Real	116
8.13.1	SelfType	116
8.13.2	ParentType	117
8.13.3	ValueType	117
8.13.4	GetParent	117
8.13.5	GetParent	118
8.13.6	HasParent	118
8.13.7	GetValue	118
8.13.8	SetValue	119
8.13.9	Make	119
8.13.10	Make	120
8.13.11	Real	120
8.14	ara::rest::ogm::String	120
8.14.1	SelfType	121
8.14.2	ParentType	121
8.14.3	ValueType	121
8.14.4	GetParent	122
8.14.5	GetParent	122
8.14.6	HasParent	122
8.14.7	GetValue	123
8.14.8	SetValue	123
8.14.9	Make	124
8.14.10	Make	124
8.14.11	String	125
8.14.12	String	125
8.15	ara::rest::ogm::Value	125
8.15.1	SelfType	126
8.15.2	ParentType	126
8.15.3	GetParent	126
8.15.4	GetParent	127
8.15.5	HasParent	127
8.15.6	Value	127
8.16	ara::rest::Pattern	128
8.16.1	Pattern	128
8.16.2	operator==	128
8.16.3	operator!=	129
8.16.4	operator<	129
8.17	ara::rest::ReplyHeader	130
8.17.1	GetStatus	130
8.17.2	SetStatus	130
8.17.3	GetUri	130
8.17.4	SetUri	131
8.17.5	HasField	131
8.17.6	InsertField	132
8.17.7	EraseField	132
8.17.8	GetField	132

8.17.9	SetField	133
8.17.10	NumFields	133
8.17.11	ClearFields	134
8.17.12	FieldIteratorRange	134
8.17.13	ConstFieldIteratorRange	134
8.17.14	FindField	135
8.17.15	GetFields	136
8.18	ara::rest::Reply	136
8.18.1	Reply	136
8.18.2	operator=	137
8.18.3	GetHeader	137
8.18.4	GetObject	138
8.18.5	ReleaseObject	138
8.18.6	ReleaseBinary	138
8.19	ara::rest::RequestHeader	139
8.19.1	GetMethod	139
8.19.2	SetMethod	139
8.19.3	GetUri	140
8.19.4	SetUri	140
8.19.5	HasField	140
8.19.6	InsertField	141
8.19.7	EraseField	141
8.19.8	GetField	142
8.19.9	SetField	142
8.19.10	NumFields	143
8.19.11	ClearFields	143
8.19.12	FieldIteratorRange	143
8.19.13	ConstFieldIteratorRange	144
8.19.14	FindField	144
8.19.15	GetFields	145
8.19.16	GetStatus	145
8.19.17	SetStatus	146
8.20	ara::rest::Request	146
8.20.1	Request	146
8.20.2	operator=	147
8.20.3	Request	147
8.20.4	Request	148
8.20.5	Request	148
8.20.6	Request	148
8.20.7	Request	149
8.20.8	Request	149
8.20.9	Request	150
8.21	ara::rest::Router	150
8.21.1	RouteHandlerType	150
8.21.2	RouteRange	151
8.21.3	ConstRouteRange	151



8.21.4	Router	151
8.21.5	Router	152
8.21.6	operator()	152
8.21.7	InsertRoute	153
8.21.8	EmplaceRoute	153
8.21.9	SetDefaultHandler	154
8.21.10	RouteCount	154
8.21.11	Routes	154
8.21.12	Routes	155
8.21.13	RemoveRoute	155
8.21.14	FindRoute	155
8.21.15	Clear	156
8.22	ara::rest::Route	156
8.22.1	Upshot	156
8.22.2	RouteHandlerType	157
8.22.3	Route	157
8.22.4	operator()	158
8.22.5	GetRequestMethod	158
8.22.6	GetPattern	159
8.22.7	operator==	159
8.22.8	operator!=	159
8.22.9	operator<	160
8.23	ara::rest::ServerEvent	160
8.23.1	ServerEvent	160
8.23.2	operator=	161
8.23.3	Notify	161
8.23.4	Notify	162
8.23.5	SetSubscriptionState	162
8.23.6	GetSubscriptionState	162
8.23.7	GetUri	163
8.23.8	SendError	163
8.23.9	operator==	164
8.23.10	operator!=	164
8.23.11	operator<	165
8.24	ara::rest::ServerReply	165
8.24.1	ServerReply	165
8.24.2	operator=	166
8.24.3	GetHeader	166
8.24.4	Send	166
8.24.5	Send	167
8.24.6	Send	167
8.24.7	Redirect	167
8.25	ara::rest::ServerRequest	168
8.25.1	ServerRequest	168
8.25.2	operator=	168
8.25.3	GetHeader	169

8.25.4	GetObject . . . . .	169
8.25.5	ReleaseObject . . . . .	170
8.25.6	ReleaseBinary . . . . .	170
8.26	ara::rest::Server . . . . .	171
8.26.1	RequestHandlerType . . . . .	171
8.26.2	SubscriptionHandlerType . . . . .	171
8.26.3	SubscriptionStateHandlerType . . . . .	171
8.26.4	Server . . . . .	172
8.26.5	operator= . . . . .	172
8.26.6	Server . . . . .	172
8.26.7	Start . . . . .	173
8.26.8	Stop . . . . .	173
8.26.9	ObserveSubscriptions . . . . .	174
8.26.10	GetError . . . . .	174
8.26.11	ObserveError . . . . .	175
8.27	ara::rest::StdAllocator . . . . .	175
8.27.1	value_type . . . . .	175
8.27.2	StdAllocator . . . . .	176
8.27.3	StdAllocator . . . . .	176
8.27.4	StdAllocator . . . . .	177
8.27.5	allocate . . . . .	177
8.27.6	deallocate . . . . .	177
8.27.7	select_on_container_copy_construction . . . . .	178
8.27.8	resource . . . . .	178
8.28	ara::rest::Uri::Builder . . . . .	179
8.28.1	Builder . . . . .	179
8.28.2	Builder . . . . .	179
8.28.3	Builder . . . . .	180
8.28.4	Builder . . . . .	180
8.28.5	Scheme . . . . .	181
8.28.6	UserInfo . . . . .	181
8.28.7	Host . . . . .	181
8.28.8	Port . . . . .	182
8.28.9	Path . . . . .	182
8.28.10	Path . . . . .	183
8.28.11	PathSegment . . . . .	183
8.28.12	PathSegments . . . . .	184
8.28.13	PathSegmentsFrom . . . . .	184
8.28.14	PathSegmentAt . . . . .	184
8.28.15	PathSegmentAt . . . . .	185
8.28.16	Query . . . . .	185
8.28.17	QueryParameter . . . . .	186
8.28.18	QueryParameter . . . . .	186
8.28.19	QueryParameterAt . . . . .	187
8.28.20	QueryParameterAt . . . . .	187
8.28.21	QueryParameterAt . . . . .	188

8.28.22	Fragment	188
8.28.23	Fragment	188
8.28.24	ToUri	189
8.28.25	ToPath	189
8.28.26	ToQuery	190
8.29	ara::rest::Uri::Path::Segment	190
8.29.1	Get	190
8.29.2	GetAs	191
8.29.3	operator==	191
8.29.4	operator!=	191
8.29.5	operator<	192
8.30	ara::rest::Uri::Path	192
8.30.1	IteratorRange	193
8.30.2	NumSegments	193
8.30.3	GetSegments	193
8.30.4	operator==	194
8.30.5	operator!=	194
8.30.6	operator<	194
8.31	ara::rest::Uri::Query::Parameter	195
8.31.1	GetKey	195
8.31.2	GetKeyAs	196
8.31.3	HasValue	196
8.31.4	GetValue	196
8.31.5	GetValueAs	197
8.32	ara::rest::Uri::Query	197
8.32.1	IteratorRange	198
8.32.2	NumParameters	198
8.32.3	GetParameters	198
8.32.4	GetParameter	199
8.32.5	Find	199
8.32.6	HasKey	200
8.33	ara::rest::Uri	200
8.33.1	Part	200
8.33.2	LENGTH_MAX	201
8.33.3	operator	201
8.33.4	Uri	202
8.33.5	HasScheme	202
8.33.6	GetScheme	203
8.33.7	HasUserInfo	203
8.33.8	GetUserinfo	203
8.33.9	HasHost	204
8.33.10	GetHost	204
8.33.11	HasPort	204
8.33.12	GetPort	205
8.33.13	HasPath	205
8.33.14	GetPath	205

8.33.15	HasQuery	206
8.33.16	GetQuery	206
8.33.17	HasFragment	207
8.33.18	GetFragment	207
8.33.19	GetFragmentAs	207
8.33.20	IsEmpty	208
8.33.21	IsRelative	208
8.33.22	IsOpaque	209
8.33.23	IsHierarchical	209
8.34	ara::rest::Uuid	209
8.34.1	MakeV1	210
8.34.2	MakeV3	210
8.34.3	MakeV4	210
8.34.4	MakeV5	211
8.34.5	Uuid	211
8.34.6	Uuid	212
8.34.7	Uuid	212
8.34.8	GetTimeLow	212
8.34.9	GetTimeMid	213
8.34.10	GetTimeHighAndVersion	213
8.34.11	GetClockSeq	214
8.34.12	GetNode	214
8.34.13	operator==	214
8.34.14	operator!=	215
8.34.15	operator<	215
8.35	ara::rest::ogm	215
8.35.1	Copy	215
8.35.2	Copy	216
8.35.3	Visit	216
8.35.4	Visit	217
8.35.5	Visit	217
8.35.6	Visit	218
8.35.7	VisitAll	218
8.35.8	VisitAll	219
8.35.9	VisitAll	219
8.35.10	VisitAll	219
8.35.11	Get	220
8.35.12	Get	220
8.35.13	GetValue	221
8.35.14	GetValue	221
8.35.15	Set	222
8.35.16	Set	222
8.35.17	Set	222
8.35.18	SetValue	223
8.35.19	SetValue	223
8.35.20	Cast	224

8.36	ara::rest	224
8.36.1	RequestMethod	224
8.36.2	SubscriptionState	225
8.36.3	EventPolicy	225
8.36.4	ShutdownPolicy	226
8.36.5	StartupPolicy	226
8.36.6	Function	227
8.36.7	Pointer	227
8.36.8	Task	227
8.36.9	duration_t	228
8.36.10	operator==	228
8.36.11	operator!=	229
8.36.12	NewDeleteAllocator	229
8.36.13	GetDefaultAllocator	229
8.36.14	SetDefaultAllocator	230
8.36.15	operator==	230
8.36.16	operator!=	230
8.36.17	operator	231
8.36.18	operator	231
8.36.19	MakeIteratorRange	232
8.36.20	MakeMoveIteratorRange	232
8.36.21	Resolve	233
8.36.22	Normalize	233
8.36.23	Relativize	233
8.36.24	ToString	234
8.36.25	ToString	234
8.36.26	ToString	235
8.36.27	ToString	235
8.36.28	ToString	236
8.36.29	ToString	236
8.36.30	ToString	237
8.36.31	InstanceIdentifier	237
A	Mentioned Class Tables	238

# 1 Introduction and functional overview

This document contains the requirements on the functionality, API and the configuration of the AUTOSAR Adaptive RESTful Communication as part of the Adaptive AUTOSAR platform foundation.

The Communication Management in general realizes functionality to establish communication paths between Adaptive AUTOSAR Applications. This document describes the RESTful Communication part (ara::rest API) whereas [1] covers Service Oriented Communication (ara::com API).

The API design of ara::rest is based on the REST paradigm introduced by [2]. The API is geared towards low and predictable resource usage which is often important in the automotive domain. This specification is focused on the low-level components to provide all features required to design RESTful APIs and deploy services on top.

## 2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations relevant to the Communication Management that are not included in the AUTOSAR glossary [3].

<b>Abbreviation / Acronym:</b>	<b>Description:</b>
REST	Representational State Transfer
HTTP	Hypertext Transfer Protocol
TLS	Transport Layer Security
MIME	Multipurpose Internet Mail Extensions

## 3 Related documentation

### 3.1 Input documents

- [1] Specification of Communication Management  
AUTOSAR\_SWS\_CommunicationManagement
- [2] REST: Architectural Styles and the Design of Network-based Software Architectures
- [3] Glossary  
AUTOSAR\_TR\_Glossary
- [4] Requirements on Communication Management  
AUTOSAR\_RS\_CommunicationManagement
- [5] RFC 3986, Uniform Resource Identifier (URI): Generic Syntax
- [6] SOME/IP Protocol Specification  
AUTOSAR\_PRS\_SOMEIPProtocol
- [7] RFC 4122, A Universally Unique IDentifier (UUID) URN Namespace
- [8] RFC 2616, Hypertext Transfer Protocol – HTTP/1.1
- [9] RFC 7159, The JavaScript Object Notation (JSON) Data Interchange Format
- [10] RFC 1951, DEFLATE Compressed Data Format Specification version 1.3
- [11] RFC 1952, GZIP file format specification version 4.3

### 3.2 Related standards and norms

See chapter [3.1](#).

### 3.3 Related specification

See chapter [3.1](#).



## 4 Constraints and assumptions

### 4.1 Limitations

The interfaces are only specified to the point to make semantics clear. To be precise this document does not yet fully specify the qualification C++ functions noexcept, overloading of functions to provide move semantics for optimization purposes nor does it claim to be const-correct. Move semantics in particular are specified where required for semantic correctness only. Also only HTTP network binding aspects of the AUTOSAR meta model are currently supported by the *SWS\_REST*. No modeling of the [Rest-ServiceInterface](#) internal structure is possible with the current *SWS\_REST*. The error handling for *RESTful* communication is currently limited due to the fact that errors are not reported in the context of a request transmission.

### 4.2 Applicability to car domains

No restrictions to applicability.

## 5 Dependencies to other functional clusters

There are currently no dependencies to other functional clusters.

## 6 Requirements Tracing

The following tables reference the requirements specified in the Requirements on Communication Management document [4] and links to the fulfillment of these.

Please note that if column “Satisfied by” is empty for a specific requirement this means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by
[RS_CM_00300]	The Communication Management shall provide a framework to support the RESTful communication paradigm introduced by [2] .	<a href="#">[SWS_REST_01101]</a> <a href="#">[SWS_REST_01102]</a> <a href="#">[SWS_REST_01103]</a> <a href="#">[SWS_REST_01104]</a> <a href="#">[SWS_REST_01105]</a> <a href="#">[SWS_REST_01106]</a> <a href="#">[SWS_REST_01107]</a> <a href="#">[SWS_REST_01108]</a> <a href="#">[SWS_REST_01109]</a> <a href="#">[SWS_REST_01110]</a> <a href="#">[SWS_REST_01111]</a> <a href="#">[SWS_REST_01201]</a> <a href="#">[SWS_REST_01203]</a> <a href="#">[SWS_REST_01301]</a> <a href="#">[SWS_REST_01302]</a> <a href="#">[SWS_REST_01304]</a> <a href="#">[SWS_REST_01305]</a> <a href="#">[SWS_REST_01306]</a> <a href="#">[SWS_REST_01307]</a> <a href="#">[SWS_REST_01308]</a> <a href="#">[SWS_REST_01312]</a> <a href="#">[SWS_REST_01313]</a> <a href="#">[SWS_REST_01314]</a> <a href="#">[SWS_REST_01315]</a> <a href="#">[SWS_REST_01316]</a> <a href="#">[SWS_REST_01317]</a> <a href="#">[SWS_REST_01318]</a> <a href="#">[SWS_REST_01401]</a> <a href="#">[SWS_REST_01402]</a> <a href="#">[SWS_REST_01403]</a> <a href="#">[SWS_REST_01404]</a> <a href="#">[SWS_REST_01405]</a> <a href="#">[SWS_REST_01406]</a> <a href="#">[SWS_REST_01407]</a> <a href="#">[SWS_REST_01408]</a> <a href="#">[SWS_REST_01409]</a> <a href="#">[SWS_REST_01410]</a> <a href="#">[SWS_REST_01411]</a> <a href="#">[SWS_REST_01412]</a> <a href="#">[SWS_REST_01413]</a> <a href="#">[SWS_REST_01414]</a> <a href="#">[SWS_REST_01415]</a> <a href="#">[SWS_REST_01416]</a> <a href="#">[SWS_REST_01417]</a> <a href="#">[SWS_REST_01418]</a> <a href="#">[SWS_REST_01419]</a> <a href="#">[SWS_REST_01420]</a> <a href="#">[SWS_REST_01421]</a> <a href="#">[SWS_REST_01422]</a> <a href="#">[SWS_REST_01501]</a> <a href="#">[SWS_REST_01502]</a> <a href="#">[SWS_REST_01503]</a> <a href="#">[SWS_REST_01504]</a> <a href="#">[SWS_REST_01505]</a> <a href="#">[SWS_REST_01506]</a> <a href="#">[SWS_REST_01507]</a> <a href="#">[SWS_REST_01508]</a> <a href="#">[SWS_REST_01509]</a> <a href="#">[SWS_REST_01510]</a> <a href="#">[SWS_REST_01511]</a> <a href="#">[SWS_REST_01512]</a> <a href="#">[SWS_REST_01513]</a> <a href="#">[SWS_REST_01514]</a> <a href="#">[SWS_REST_01515]</a> <a href="#">[SWS_REST_01516]</a> <a href="#">[SWS_REST_01517]</a> <a href="#">[SWS_REST_01518]</a> <a href="#">[SWS_REST_01519]</a> <a href="#">[SWS_REST_01522]</a> <a href="#">[SWS_REST_01523]</a> <a href="#">[SWS_REST_01524]</a> <a href="#">[SWS_REST_01525]</a>

Requirement	Description	Satisfied by
		[SWS_REST_01526] [SWS_REST_01527]
		[SWS_REST_01528] [SWS_REST_01529]
		[SWS_REST_01530] [SWS_REST_01531]
		[SWS_REST_01532] [SWS_REST_01533]
		[SWS_REST_01534] [SWS_REST_01535]
		[SWS_REST_01536] [SWS_REST_01537]
		[SWS_REST_01538] [SWS_REST_01601]
		[SWS_REST_01602] [SWS_REST_01603]
		[SWS_REST_01604] [SWS_REST_01605]
		[SWS_REST_01606] [SWS_REST_01607]
		[SWS_REST_01608] [SWS_REST_01609]
		[SWS_REST_01610] [SWS_REST_01611]
		[SWS_REST_01612] [SWS_REST_01613]
		[SWS_REST_01614] [SWS_REST_01615]
		[SWS_REST_01616] [SWS_REST_01617]
		[SWS_REST_01618] [SWS_REST_01619]
		[SWS_REST_01620] [SWS_REST_01621]
		[SWS_REST_01622] [SWS_REST_01623]
		[SWS_REST_01624] [SWS_REST_01625]
		[SWS_REST_01626] [SWS_REST_01627]
		[SWS_REST_01628] [SWS_REST_01629]
		[SWS_REST_01701] [SWS_REST_01702]
		[SWS_REST_01703] [SWS_REST_01704]
		[SWS_REST_01705] [SWS_REST_01706]
		[SWS_REST_01707] [SWS_REST_01708]
		[SWS_REST_01709] [SWS_REST_01710]
		[SWS_REST_01711] [SWS_REST_01712]
		[SWS_REST_01713] [SWS_REST_01714]
		[SWS_REST_01715] [SWS_REST_02000]
		[SWS_REST_02001] [SWS_REST_02002]
		[SWS_REST_02003] [SWS_REST_02004]
		[SWS_REST_02005] [SWS_REST_02006]
		[SWS_REST_02007] [SWS_REST_02008]
		[SWS_REST_02009] [SWS_REST_02010]
		[SWS_REST_02011] [SWS_REST_02012]
		[SWS_REST_02013] [SWS_REST_02014]
		[SWS_REST_02015] [SWS_REST_02016]
		[SWS_REST_02017] [SWS_REST_02018]
		[SWS_REST_02019] [SWS_REST_02020]
		[SWS_REST_02021] [SWS_REST_02022]
		[SWS_REST_02023] [SWS_REST_02024]
		[SWS_REST_02025] [SWS_REST_02026]
		[SWS_REST_02027] [SWS_REST_02028]
		[SWS_REST_02029] [SWS_REST_02030]
		[SWS_REST_02031] [SWS_REST_02033]
		[SWS_REST_02034] [SWS_REST_02035]
		[SWS_REST_02036] [SWS_REST_02037]
		[SWS_REST_02038] [SWS_REST_02039]

Requirement	Description	Satisfied by
		[SWS_REST_02040] [SWS_REST_02041]
		[SWS_REST_02042] [SWS_REST_02043]
		[SWS_REST_02044] [SWS_REST_02045]
		[SWS_REST_02046] [SWS_REST_02047]
		[SWS_REST_02048] [SWS_REST_02049]
		[SWS_REST_02050] [SWS_REST_02051]
		[SWS_REST_02052] [SWS_REST_02053]
		[SWS_REST_02054] [SWS_REST_02055]
		[SWS_REST_02056] [SWS_REST_02057]
		[SWS_REST_02058] [SWS_REST_02059]
		[SWS_REST_02060] [SWS_REST_02061]
		[SWS_REST_02062] [SWS_REST_02063]
		[SWS_REST_02064] [SWS_REST_02065]
		[SWS_REST_02066] [SWS_REST_02067]
		[SWS_REST_02068] [SWS_REST_02069]
		[SWS_REST_02070] [SWS_REST_02071]
		[SWS_REST_02072] [SWS_REST_02073]
		[SWS_REST_02074] [SWS_REST_02075]
		[SWS_REST_02076] [SWS_REST_02077]
		[SWS_REST_02078] [SWS_REST_02079]
		[SWS_REST_02080] [SWS_REST_02081]
		[SWS_REST_02082] [SWS_REST_02083]
		[SWS_REST_02084] [SWS_REST_02085]
		[SWS_REST_02086] [SWS_REST_02087]
		[SWS_REST_02088] [SWS_REST_02089]
		[SWS_REST_02090] [SWS_REST_02091]
		[SWS_REST_02092] [SWS_REST_02093]
		[SWS_REST_02094] [SWS_REST_02095]
		[SWS_REST_02096] [SWS_REST_02097]
		[SWS_REST_02098] [SWS_REST_02099]
		[SWS_REST_02100] [SWS_REST_02101]
		[SWS_REST_02102] [SWS_REST_02103]
		[SWS_REST_02104] [SWS_REST_02105]
		[SWS_REST_02106] [SWS_REST_02107]
		[SWS_REST_02108] [SWS_REST_02109]
		[SWS_REST_02110] [SWS_REST_02111]
		[SWS_REST_02112] [SWS_REST_02113]
		[SWS_REST_02114] [SWS_REST_02115]
		[SWS_REST_02116] [SWS_REST_02117]
		[SWS_REST_02118] [SWS_REST_02119]
		[SWS_REST_02120] [SWS_REST_02121]
		[SWS_REST_02122] [SWS_REST_02123]
		[SWS_REST_02124] [SWS_REST_02125]
		[SWS_REST_02126] [SWS_REST_02127]
		[SWS_REST_02128] [SWS_REST_02129]
		[SWS_REST_02130] [SWS_REST_02131]
		[SWS_REST_02132] [SWS_REST_02133]
		[SWS_REST_02134] [SWS_REST_02135]

Requirement	Description	Satisfied by
		[SWS_REST_02136] [SWS_REST_02137]
		[SWS_REST_02138] [SWS_REST_02139]
		[SWS_REST_02140] [SWS_REST_02141]
		[SWS_REST_02142] [SWS_REST_02143]
		[SWS_REST_02144] [SWS_REST_02145]
		[SWS_REST_02146] [SWS_REST_02147]
		[SWS_REST_02148] [SWS_REST_02149]
		[SWS_REST_02150] [SWS_REST_02151]
		[SWS_REST_02152] [SWS_REST_02153]
		[SWS_REST_02154] [SWS_REST_02155]
		[SWS_REST_02156] [SWS_REST_02157]
		[SWS_REST_02158] [SWS_REST_02159]
		[SWS_REST_02160] [SWS_REST_02161]
		[SWS_REST_02162] [SWS_REST_02163]
		[SWS_REST_02164] [SWS_REST_02165]
		[SWS_REST_02166] [SWS_REST_02167]
		[SWS_REST_02168] [SWS_REST_02169]
		[SWS_REST_02170] [SWS_REST_02171]
		[SWS_REST_02172] [SWS_REST_02173]
		[SWS_REST_02174] [SWS_REST_02175]
		[SWS_REST_02176] [SWS_REST_02177]
		[SWS_REST_02178] [SWS_REST_02179]
		[SWS_REST_02180] [SWS_REST_02181]
		[SWS_REST_02182] [SWS_REST_02183]
		[SWS_REST_02184] [SWS_REST_02185]
		[SWS_REST_02186] [SWS_REST_02187]
		[SWS_REST_02188] [SWS_REST_02189]
		[SWS_REST_02190] [SWS_REST_02191]
		[SWS_REST_02192] [SWS_REST_02193]
		[SWS_REST_02194] [SWS_REST_02195]
		[SWS_REST_02196] [SWS_REST_02197]
		[SWS_REST_02198] [SWS_REST_02199]
		[SWS_REST_02200] [SWS_REST_02201]
		[SWS_REST_02202] [SWS_REST_02203]
		[SWS_REST_02204] [SWS_REST_02205]
		[SWS_REST_02206] [SWS_REST_02207]
		[SWS_REST_02208] [SWS_REST_02209]
		[SWS_REST_02210] [SWS_REST_02211]
		[SWS_REST_02212] [SWS_REST_02213]
		[SWS_REST_02214] [SWS_REST_02215]
		[SWS_REST_02216] [SWS_REST_02217]
		[SWS_REST_02218] [SWS_REST_02219]
		[SWS_REST_02220] [SWS_REST_02221]
		[SWS_REST_02222] [SWS_REST_02223]
		[SWS_REST_02224] [SWS_REST_02225]
		[SWS_REST_02226] [SWS_REST_02227]
		[SWS_REST_02228] [SWS_REST_02229]
		[SWS_REST_02230] [SWS_REST_02231]

Requirement	Description	Satisfied by
		[SWS_REST_02232] [SWS_REST_02233]
		[SWS_REST_02234] [SWS_REST_02235]
		[SWS_REST_02236] [SWS_REST_02237]
		[SWS_REST_02238] [SWS_REST_02239]
		[SWS_REST_02240] [SWS_REST_02241]
		[SWS_REST_02242] [SWS_REST_02243]
		[SWS_REST_02244] [SWS_REST_02245]
		[SWS_REST_02246] [SWS_REST_02247]
		[SWS_REST_02248] [SWS_REST_02249]
		[SWS_REST_02250] [SWS_REST_02251]
		[SWS_REST_02252] [SWS_REST_02253]
		[SWS_REST_02254] [SWS_REST_02255]
		[SWS_REST_02256] [SWS_REST_02257]
		[SWS_REST_02258] [SWS_REST_02259]
		[SWS_REST_02260] [SWS_REST_02261]
		[SWS_REST_02262] [SWS_REST_02263]
		[SWS_REST_02264] [SWS_REST_02265]
		[SWS_REST_02266] [SWS_REST_02267]
		[SWS_REST_02268] [SWS_REST_02269]
		[SWS_REST_02270] [SWS_REST_02271]
		[SWS_REST_02272] [SWS_REST_02273]
		[SWS_REST_02274] [SWS_REST_02275]
		[SWS_REST_02276] [SWS_REST_02277]
		[SWS_REST_02278] [SWS_REST_02279]
		[SWS_REST_02280] [SWS_REST_02281]
		[SWS_REST_02282] [SWS_REST_02283]
		[SWS_REST_02284] [SWS_REST_02285]
		[SWS_REST_02286] [SWS_REST_02287]
		[SWS_REST_02288] [SWS_REST_02289]
		[SWS_REST_02290] [SWS_REST_02291]
		[SWS_REST_02292] [SWS_REST_02293]
		[SWS_REST_02294] [SWS_REST_02295]
		[SWS_REST_02296] [SWS_REST_02297]
		[SWS_REST_02298] [SWS_REST_02299]
		[SWS_REST_02300] [SWS_REST_02301]
		[SWS_REST_02302] [SWS_REST_02303]
		[SWS_REST_02304] [SWS_REST_02305]
		[SWS_REST_02306] [SWS_REST_02307]
		[SWS_REST_02308] [SWS_REST_02309]
		[SWS_REST_02310] [SWS_REST_02311]
		[SWS_REST_02312] [SWS_REST_02313]
		[SWS_REST_02314] [SWS_REST_02315]
		[SWS_REST_02316] [SWS_REST_02317]
		[SWS_REST_02318] [SWS_REST_02319]
		[SWS_REST_02320] [SWS_REST_02321]
		[SWS_REST_02322] [SWS_REST_02323]
		[SWS_REST_02324] [SWS_REST_02325]
		[SWS_REST_02326] [SWS_REST_02327]

Requirement	Description	Satisfied by
		[SWS_REST_02328] [SWS_REST_02329]
		[SWS_REST_02330] [SWS_REST_02331]
		[SWS_REST_02332] [SWS_REST_02333]
		[SWS_REST_02334] [SWS_REST_02335]
		[SWS_REST_02336] [SWS_REST_02337]
		[SWS_REST_02338] [SWS_REST_02339]
		[SWS_REST_02340] [SWS_REST_02341]
		[SWS_REST_02342] [SWS_REST_02343]
		[SWS_REST_02344] [SWS_REST_02345]
		[SWS_REST_02346] [SWS_REST_02347]
		[SWS_REST_02348] [SWS_REST_02349]
		[SWS_REST_02350] [SWS_REST_02351]
		[SWS_REST_02352] [SWS_REST_02353]
		[SWS_REST_02354] [SWS_REST_02355]
		[SWS_REST_02360] [SWS_REST_02361]
		[SWS_REST_02362] [SWS_REST_02363]
		[SWS_REST_02364] [SWS_REST_02365]
		[SWS_REST_02366] [SWS_REST_02367]
		[SWS_REST_02368] [SWS_REST_02369]
		[SWS_REST_02370] [SWS_REST_02371]
		[SWS_REST_02372] [SWS_REST_02373]
		[SWS_REST_02374] [SWS_REST_02375]
		[SWS_REST_02376] [SWS_REST_02377]
		[SWS_REST_02378] [SWS_REST_02379]
		[SWS_REST_02380] [SWS_REST_02381]
		[SWS_REST_02382] [SWS_REST_02383]
		[SWS_REST_02384] [SWS_REST_02385]
		[SWS_REST_02386] [SWS_REST_02387]
		[SWS_REST_02388] [SWS_REST_02389]
		[SWS_REST_02390] [SWS_REST_02391]
		[SWS_REST_02392] [SWS_REST_02393]
		[SWS_REST_02395] [SWS_REST_02396]
		[SWS_REST_02397] [SWS_REST_02398]
		[SWS_REST_02399] [SWS_REST_02400]
		[SWS_REST_02401] [SWS_REST_02402]
		[SWS_REST_02403] [SWS_REST_02404]
		[SWS_REST_02405] [SWS_REST_02406]
		[SWS_REST_02407] [SWS_REST_02409]
		[SWS_REST_02410] [SWS_REST_02411]
		[SWS_REST_02412] [SWS_REST_02413]
		[SWS_REST_02414] [SWS_REST_02415]
		[SWS_REST_02416] [SWS_REST_02417]
		[SWS_REST_02418] [SWS_REST_02419]
		[SWS_REST_02420] [SWS_REST_02421]
		[SWS_REST_02422] [SWS_REST_02423]
		[SWS_REST_02424] [SWS_REST_02425]
		[SWS_REST_02426] [SWS_REST_02427]
		[SWS_REST_02428] [SWS_REST_02489]



Requirement	Description	Satisfied by
		<a href="#">[SWS_REST_02490]</a> <a href="#">[SWS_REST_02492]</a> <a href="#">[SWS_REST_02493]</a> <a href="#">[SWS_REST_02494]</a> <a href="#">[SWS_REST_02496]</a> <a href="#">[SWS_REST_02497]</a> <a href="#">[SWS_REST_02498]</a> <a href="#">[SWS_REST_02499]</a> <a href="#">[SWS_REST_02501]</a> <a href="#">[SWS_REST_02502]</a> <a href="#">[SWS_REST_02503]</a> <a href="#">[SWS_REST_02505]</a> <a href="#">[SWS_REST_02506]</a> <a href="#">[SWS_REST_02507]</a> <a href="#">[SWS_REST_02508]</a> <a href="#">[SWS_REST_02511]</a> <a href="#">[SWS_REST_02512]</a> <a href="#">[SWS_REST_02513]</a> <a href="#">[SWS_REST_02514]</a> <a href="#">[SWS_REST_02515]</a> <a href="#">[SWS_REST_02516]</a> <a href="#">[SWS_REST_02517]</a> <a href="#">[SWS_REST_02518]</a> <a href="#">[SWS_REST_02519]</a> <a href="#">[SWS_REST_02520]</a> <a href="#">[SWS_REST_02528]</a> <a href="#">[SWS_REST_02529]</a> <a href="#">[SWS_REST_02805]</a> <a href="#">[SWS_REST_02889]</a> <a href="#">[SWS_REST_02932]</a> <a href="#">[SWS_REST_02973]</a> <a href="#">[SWS_REST_02989]</a> <a href="#">[SWS_REST_02991]</a> <a href="#">[SWS_REST_10902]</a>
<b>[RS_CM_00301]</b>	The Communication Management shall provide an abstraction of network protocols for RESTful services.	<a href="#">[SWS_REST_01301]</a> <a href="#">[SWS_REST_01302]</a> <a href="#">[SWS_REST_01304]</a> <a href="#">[SWS_REST_01305]</a> <a href="#">[SWS_REST_01306]</a> <a href="#">[SWS_REST_01307]</a> <a href="#">[SWS_REST_01308]</a> <a href="#">[SWS_REST_01312]</a> <a href="#">[SWS_REST_01313]</a> <a href="#">[SWS_REST_01314]</a> <a href="#">[SWS_REST_01315]</a> <a href="#">[SWS_REST_01316]</a> <a href="#">[SWS_REST_01317]</a> <a href="#">[SWS_REST_01318]</a> <a href="#">[SWS_REST_01401]</a> <a href="#">[SWS_REST_01402]</a> <a href="#">[SWS_REST_01403]</a> <a href="#">[SWS_REST_01404]</a> <a href="#">[SWS_REST_01405]</a> <a href="#">[SWS_REST_01406]</a> <a href="#">[SWS_REST_01407]</a> <a href="#">[SWS_REST_01408]</a> <a href="#">[SWS_REST_01409]</a> <a href="#">[SWS_REST_01410]</a> <a href="#">[SWS_REST_01411]</a> <a href="#">[SWS_REST_01412]</a> <a href="#">[SWS_REST_01413]</a> <a href="#">[SWS_REST_01414]</a> <a href="#">[SWS_REST_01415]</a> <a href="#">[SWS_REST_01416]</a> <a href="#">[SWS_REST_01417]</a> <a href="#">[SWS_REST_01418]</a> <a href="#">[SWS_REST_01419]</a> <a href="#">[SWS_REST_01420]</a> <a href="#">[SWS_REST_01421]</a> <a href="#">[SWS_REST_01422]</a> <a href="#">[SWS_REST_01501]</a> <a href="#">[SWS_REST_01502]</a> <a href="#">[SWS_REST_01503]</a> <a href="#">[SWS_REST_01504]</a> <a href="#">[SWS_REST_01505]</a> <a href="#">[SWS_REST_01506]</a> <a href="#">[SWS_REST_01507]</a> <a href="#">[SWS_REST_01508]</a> <a href="#">[SWS_REST_01509]</a> <a href="#">[SWS_REST_01510]</a> <a href="#">[SWS_REST_01511]</a> <a href="#">[SWS_REST_01512]</a> <a href="#">[SWS_REST_01513]</a> <a href="#">[SWS_REST_01514]</a> <a href="#">[SWS_REST_01515]</a> <a href="#">[SWS_REST_01516]</a> <a href="#">[SWS_REST_01517]</a> <a href="#">[SWS_REST_01518]</a> <a href="#">[SWS_REST_01519]</a> <a href="#">[SWS_REST_01522]</a> <a href="#">[SWS_REST_01523]</a> <a href="#">[SWS_REST_01524]</a> <a href="#">[SWS_REST_01525]</a> <a href="#">[SWS_REST_01526]</a> <a href="#">[SWS_REST_01527]</a> <a href="#">[SWS_REST_01528]</a> <a href="#">[SWS_REST_01529]</a> <a href="#">[SWS_REST_01530]</a> <a href="#">[SWS_REST_01531]</a> <a href="#">[SWS_REST_01532]</a> <a href="#">[SWS_REST_01533]</a> <a href="#">[SWS_REST_01534]</a> <a href="#">[SWS_REST_01535]</a> <a href="#">[SWS_REST_01536]</a> <a href="#">[SWS_REST_01537]</a> <a href="#">[SWS_REST_01538]</a>

Requirement	Description	Satisfied by
		<a href="#">[SWS_REST_02006]</a> <a href="#">[SWS_REST_02007]</a> <a href="#">[SWS_REST_02008]</a> <a href="#">[SWS_REST_02009]</a> <a href="#">[SWS_REST_02010]</a> <a href="#">[SWS_REST_02011]</a> <a href="#">[SWS_REST_02012]</a> <a href="#">[SWS_REST_02013]</a> <a href="#">[SWS_REST_02014]</a> <a href="#">[SWS_REST_02015]</a> <a href="#">[SWS_REST_02016]</a> <a href="#">[SWS_REST_02238]</a> <a href="#">[SWS_REST_02239]</a> <a href="#">[SWS_REST_02240]</a> <a href="#">[SWS_REST_02241]</a> <a href="#">[SWS_REST_02242]</a> <a href="#">[SWS_REST_02243]</a> <a href="#">[SWS_REST_02244]</a> <a href="#">[SWS_REST_02245]</a> <a href="#">[SWS_REST_02246]</a> <a href="#">[SWS_REST_02247]</a> <a href="#">[SWS_REST_02248]</a> <a href="#">[SWS_REST_02249]</a>
<b>[RS_CM_00304]</b>	The Communication Management shall support URIs according to RFC3986 [5] to identify data.	<a href="#">[SWS_REST_01101]</a> <a href="#">[SWS_REST_01102]</a> <a href="#">[SWS_REST_02022]</a> <a href="#">[SWS_REST_02167]</a> <a href="#">[SWS_REST_02168]</a> <a href="#">[SWS_REST_02178]</a> <a href="#">[SWS_REST_02179]</a> <a href="#">[SWS_REST_02259]</a> <a href="#">[SWS_REST_02260]</a> <a href="#">[SWS_REST_02261]</a> <a href="#">[SWS_REST_02262]</a> <a href="#">[SWS_REST_02263]</a> <a href="#">[SWS_REST_02264]</a> <a href="#">[SWS_REST_02265]</a> <a href="#">[SWS_REST_02266]</a> <a href="#">[SWS_REST_02267]</a> <a href="#">[SWS_REST_02268]</a> <a href="#">[SWS_REST_02269]</a> <a href="#">[SWS_REST_02270]</a> <a href="#">[SWS_REST_02271]</a> <a href="#">[SWS_REST_02272]</a> <a href="#">[SWS_REST_02273]</a> <a href="#">[SWS_REST_02274]</a> <a href="#">[SWS_REST_02275]</a> <a href="#">[SWS_REST_02276]</a> <a href="#">[SWS_REST_02277]</a> <a href="#">[SWS_REST_02278]</a> <a href="#">[SWS_REST_02279]</a> <a href="#">[SWS_REST_02280]</a> <a href="#">[SWS_REST_02281]</a> <a href="#">[SWS_REST_02282]</a> <a href="#">[SWS_REST_02283]</a> <a href="#">[SWS_REST_02284]</a> <a href="#">[SWS_REST_02285]</a> <a href="#">[SWS_REST_02286]</a> <a href="#">[SWS_REST_02287]</a> <a href="#">[SWS_REST_02288]</a> <a href="#">[SWS_REST_02289]</a> <a href="#">[SWS_REST_02290]</a> <a href="#">[SWS_REST_02291]</a> <a href="#">[SWS_REST_02292]</a> <a href="#">[SWS_REST_02293]</a> <a href="#">[SWS_REST_02294]</a> <a href="#">[SWS_REST_02295]</a> <a href="#">[SWS_REST_02296]</a> <a href="#">[SWS_REST_02297]</a> <a href="#">[SWS_REST_02298]</a> <a href="#">[SWS_REST_02299]</a> <a href="#">[SWS_REST_02300]</a> <a href="#">[SWS_REST_02301]</a> <a href="#">[SWS_REST_02302]</a> <a href="#">[SWS_REST_02303]</a> <a href="#">[SWS_REST_02304]</a> <a href="#">[SWS_REST_02305]</a> <a href="#">[SWS_REST_02306]</a> <a href="#">[SWS_REST_02307]</a> <a href="#">[SWS_REST_02308]</a> <a href="#">[SWS_REST_02309]</a> <a href="#">[SWS_REST_02310]</a> <a href="#">[SWS_REST_02311]</a> <a href="#">[SWS_REST_02312]</a> <a href="#">[SWS_REST_02313]</a> <a href="#">[SWS_REST_02314]</a> <a href="#">[SWS_REST_02315]</a> <a href="#">[SWS_REST_02316]</a> <a href="#">[SWS_REST_02317]</a> <a href="#">[SWS_REST_02318]</a> <a href="#">[SWS_REST_02319]</a> <a href="#">[SWS_REST_02320]</a> <a href="#">[SWS_REST_02321]</a> <a href="#">[SWS_REST_02322]</a> <a href="#">[SWS_REST_02323]</a>

Requirement	Description	Satisfied by
		<a href="#">[SWS_REST_02324]</a> <a href="#">[SWS_REST_02325]</a> <a href="#">[SWS_REST_02326]</a> <a href="#">[SWS_REST_02327]</a> <a href="#">[SWS_REST_02328]</a> <a href="#">[SWS_REST_02329]</a> <a href="#">[SWS_REST_02330]</a> <a href="#">[SWS_REST_02372]</a> <a href="#">[SWS_REST_02373]</a> <a href="#">[SWS_REST_02374]</a> <a href="#">[SWS_REST_02375]</a> <a href="#">[SWS_REST_02376]</a> <a href="#">[SWS_REST_02377]</a> <a href="#">[SWS_REST_02378]</a> <a href="#">[SWS_REST_02379]</a> <a href="#">[SWS_REST_02380]</a> <a href="#">[SWS_REST_02381]</a> <a href="#">[SWS_REST_02422]</a> <a href="#">[SWS_REST_02424]</a> <a href="#">[SWS_REST_02425]</a> <a href="#">[SWS_REST_02426]</a> <a href="#">[SWS_REST_02427]</a> <a href="#">[SWS_REST_02489]</a> <a href="#">[SWS_REST_02490]</a> <a href="#">[SWS_REST_02492]</a> <a href="#">[SWS_REST_02493]</a> <a href="#">[SWS_REST_02494]</a> <a href="#">[SWS_REST_02496]</a> <a href="#">[SWS_REST_02497]</a> <a href="#">[SWS_REST_02498]</a> <a href="#">[SWS_REST_02499]</a> <a href="#">[SWS_REST_02501]</a> <a href="#">[SWS_REST_02502]</a> <a href="#">[SWS_REST_02503]</a> <a href="#">[SWS_REST_02505]</a> <a href="#">[SWS_REST_02506]</a> <a href="#">[SWS_REST_02507]</a> <a href="#">[SWS_REST_02511]</a> <a href="#">[SWS_REST_02512]</a> <a href="#">[SWS_REST_02513]</a> <a href="#">[SWS_REST_02514]</a> <a href="#">[SWS_REST_02517]</a> <a href="#">[SWS_REST_02518]</a> <a href="#">[SWS_REST_02519]</a> <a href="#">[SWS_REST_02520]</a> <a href="#">[SWS_REST_10902]</a>
<b>[RS_CM_00305]</b>	The Communication Management shall represent data as an tree of objects.	<a href="#">[SWS_REST_01304]</a> <a href="#">[SWS_REST_01702]</a> <a href="#">[SWS_REST_01703]</a> <a href="#">[SWS_REST_01704]</a> <a href="#">[SWS_REST_01705]</a> <a href="#">[SWS_REST_01706]</a> <a href="#">[SWS_REST_01707]</a> <a href="#">[SWS_REST_01708]</a> <a href="#">[SWS_REST_01709]</a> <a href="#">[SWS_REST_01710]</a> <a href="#">[SWS_REST_01711]</a> <a href="#">[SWS_REST_01712]</a> <a href="#">[SWS_REST_01713]</a> <a href="#">[SWS_REST_01714]</a> <a href="#">[SWS_REST_01715]</a> <a href="#">[SWS_REST_02036]</a> <a href="#">[SWS_REST_02037]</a> <a href="#">[SWS_REST_02038]</a> <a href="#">[SWS_REST_02039]</a> <a href="#">[SWS_REST_02040]</a> <a href="#">[SWS_REST_02041]</a> <a href="#">[SWS_REST_02042]</a> <a href="#">[SWS_REST_02043]</a> <a href="#">[SWS_REST_02044]</a> <a href="#">[SWS_REST_02045]</a> <a href="#">[SWS_REST_02046]</a> <a href="#">[SWS_REST_02047]</a> <a href="#">[SWS_REST_02048]</a> <a href="#">[SWS_REST_02049]</a> <a href="#">[SWS_REST_02050]</a> <a href="#">[SWS_REST_02051]</a> <a href="#">[SWS_REST_02052]</a> <a href="#">[SWS_REST_02053]</a> <a href="#">[SWS_REST_02054]</a> <a href="#">[SWS_REST_02055]</a> <a href="#">[SWS_REST_02056]</a> <a href="#">[SWS_REST_02057]</a> <a href="#">[SWS_REST_02058]</a> <a href="#">[SWS_REST_02059]</a> <a href="#">[SWS_REST_02060]</a> <a href="#">[SWS_REST_02061]</a> <a href="#">[SWS_REST_02062]</a> <a href="#">[SWS_REST_02063]</a> <a href="#">[SWS_REST_02064]</a> <a href="#">[SWS_REST_02065]</a> <a href="#">[SWS_REST_02066]</a> <a href="#">[SWS_REST_02067]</a> <a href="#">[SWS_REST_02068]</a>

Requirement	Description	Satisfied by
		[SWS_REST_02069] [SWS_REST_02070]
		[SWS_REST_02071] [SWS_REST_02072]
		[SWS_REST_02073] [SWS_REST_02074]
		[SWS_REST_02075] [SWS_REST_02076]
		[SWS_REST_02077] [SWS_REST_02078]
		[SWS_REST_02079] [SWS_REST_02080]
		[SWS_REST_02081] [SWS_REST_02082]
		[SWS_REST_02083] [SWS_REST_02084]
		[SWS_REST_02085] [SWS_REST_02086]
		[SWS_REST_02087] [SWS_REST_02088]
		[SWS_REST_02089] [SWS_REST_02090]
		[SWS_REST_02091] [SWS_REST_02092]
		[SWS_REST_02093] [SWS_REST_02094]
		[SWS_REST_02095] [SWS_REST_02096]
		[SWS_REST_02097] [SWS_REST_02098]
		[SWS_REST_02099] [SWS_REST_02100]
		[SWS_REST_02101] [SWS_REST_02102]
		[SWS_REST_02103] [SWS_REST_02104]
		[SWS_REST_02105] [SWS_REST_02106]
		[SWS_REST_02107] [SWS_REST_02108]
		[SWS_REST_02109] [SWS_REST_02110]
		[SWS_REST_02111] [SWS_REST_02112]
		[SWS_REST_02113] [SWS_REST_02114]
		[SWS_REST_02115] [SWS_REST_02116]
		[SWS_REST_02117] [SWS_REST_02118]
		[SWS_REST_02119] [SWS_REST_02120]
		[SWS_REST_02121] [SWS_REST_02122]
		[SWS_REST_02123] [SWS_REST_02124]
		[SWS_REST_02125] [SWS_REST_02126]
		[SWS_REST_02127] [SWS_REST_02128]
		[SWS_REST_02129] [SWS_REST_02130]
		[SWS_REST_02131] [SWS_REST_02132]
		[SWS_REST_02133] [SWS_REST_02134]
		[SWS_REST_02135] [SWS_REST_02136]
		[SWS_REST_02137] [SWS_REST_02138]
		[SWS_REST_02139] [SWS_REST_02140]
		[SWS_REST_02141] [SWS_REST_02142]
		[SWS_REST_02143] [SWS_REST_02144]
		[SWS_REST_02145] [SWS_REST_02146]
		[SWS_REST_02147] [SWS_REST_02148]
		[SWS_REST_02149] [SWS_REST_02150]
		[SWS_REST_02151] [SWS_REST_02152]
		[SWS_REST_02153] [SWS_REST_02154]
		[SWS_REST_02155] [SWS_REST_02156]
		[SWS_REST_02157] [SWS_REST_02158]
		[SWS_REST_02343] [SWS_REST_02344]
		[SWS_REST_02345] [SWS_REST_02346]
		[SWS_REST_02347] [SWS_REST_02348]

Requirement	Description	Satisfied by
		<a href="#">[SWS_REST_02389]</a> <a href="#">[SWS_REST_02390]</a> <a href="#">[SWS_REST_02391]</a> <a href="#">[SWS_REST_02392]</a> <a href="#">[SWS_REST_02393]</a> <a href="#">[SWS_REST_02403]</a> <a href="#">[SWS_REST_02404]</a> <a href="#">[SWS_REST_02405]</a> <a href="#">[SWS_REST_02406]</a> <a href="#">[SWS_REST_02407]</a> <a href="#">[SWS_REST_02409]</a> <a href="#">[SWS_REST_02410]</a> <a href="#">[SWS_REST_02411]</a> <a href="#">[SWS_REST_02412]</a> <a href="#">[SWS_REST_02413]</a> <a href="#">[SWS_REST_02414]</a> <a href="#">[SWS_REST_02415]</a> <a href="#">[SWS_REST_02416]</a> <a href="#">[SWS_REST_02417]</a> <a href="#">[SWS_REST_02418]</a> <a href="#">[SWS_REST_02419]</a> <a href="#">[SWS_REST_02420]</a> <a href="#">[SWS_REST_02421]</a> <a href="#">[SWS_REST_02423]</a>
<b>[RS_CM_00306]</b>	The Communication Management shall provide an Object Graph which is independent of the used serialization format.	<a href="#">[SWS_REST_02036]</a> <a href="#">[SWS_REST_02037]</a> <a href="#">[SWS_REST_02038]</a> <a href="#">[SWS_REST_02039]</a> <a href="#">[SWS_REST_02040]</a> <a href="#">[SWS_REST_02041]</a> <a href="#">[SWS_REST_02042]</a> <a href="#">[SWS_REST_02043]</a> <a href="#">[SWS_REST_02044]</a> <a href="#">[SWS_REST_02045]</a> <a href="#">[SWS_REST_02046]</a> <a href="#">[SWS_REST_02047]</a> <a href="#">[SWS_REST_02048]</a> <a href="#">[SWS_REST_02049]</a> <a href="#">[SWS_REST_02050]</a> <a href="#">[SWS_REST_02051]</a> <a href="#">[SWS_REST_02052]</a> <a href="#">[SWS_REST_02053]</a> <a href="#">[SWS_REST_02054]</a> <a href="#">[SWS_REST_02055]</a> <a href="#">[SWS_REST_02056]</a> <a href="#">[SWS_REST_02057]</a> <a href="#">[SWS_REST_02058]</a> <a href="#">[SWS_REST_02059]</a> <a href="#">[SWS_REST_02060]</a> <a href="#">[SWS_REST_02061]</a> <a href="#">[SWS_REST_02062]</a> <a href="#">[SWS_REST_02063]</a> <a href="#">[SWS_REST_02064]</a> <a href="#">[SWS_REST_02065]</a> <a href="#">[SWS_REST_02066]</a> <a href="#">[SWS_REST_02067]</a> <a href="#">[SWS_REST_02068]</a> <a href="#">[SWS_REST_02069]</a> <a href="#">[SWS_REST_02070]</a> <a href="#">[SWS_REST_02071]</a> <a href="#">[SWS_REST_02072]</a> <a href="#">[SWS_REST_02073]</a> <a href="#">[SWS_REST_02074]</a> <a href="#">[SWS_REST_02075]</a> <a href="#">[SWS_REST_02076]</a> <a href="#">[SWS_REST_02077]</a> <a href="#">[SWS_REST_02078]</a> <a href="#">[SWS_REST_02079]</a> <a href="#">[SWS_REST_02080]</a> <a href="#">[SWS_REST_02081]</a> <a href="#">[SWS_REST_02082]</a> <a href="#">[SWS_REST_02083]</a> <a href="#">[SWS_REST_02084]</a> <a href="#">[SWS_REST_02085]</a> <a href="#">[SWS_REST_02086]</a> <a href="#">[SWS_REST_02087]</a> <a href="#">[SWS_REST_02088]</a> <a href="#">[SWS_REST_02089]</a> <a href="#">[SWS_REST_02090]</a> <a href="#">[SWS_REST_02091]</a> <a href="#">[SWS_REST_02092]</a> <a href="#">[SWS_REST_02093]</a> <a href="#">[SWS_REST_02094]</a> <a href="#">[SWS_REST_02095]</a> <a href="#">[SWS_REST_02096]</a> <a href="#">[SWS_REST_02097]</a> <a href="#">[SWS_REST_02098]</a> <a href="#">[SWS_REST_02099]</a> <a href="#">[SWS_REST_02100]</a> <a href="#">[SWS_REST_02101]</a> <a href="#">[SWS_REST_02102]</a> <a href="#">[SWS_REST_02103]</a> <a href="#">[SWS_REST_02104]</a> <a href="#">[SWS_REST_02105]</a> <a href="#">[SWS_REST_02106]</a> <a href="#">[SWS_REST_02107]</a>

Requirement	Description	Satisfied by
		<a href="#">[SWS_REST_02108]</a> <a href="#">[SWS_REST_02109]</a> <a href="#">[SWS_REST_02110]</a> <a href="#">[SWS_REST_02111]</a> <a href="#">[SWS_REST_02112]</a> <a href="#">[SWS_REST_02113]</a> <a href="#">[SWS_REST_02114]</a> <a href="#">[SWS_REST_02115]</a> <a href="#">[SWS_REST_02116]</a> <a href="#">[SWS_REST_02117]</a> <a href="#">[SWS_REST_02118]</a> <a href="#">[SWS_REST_02119]</a> <a href="#">[SWS_REST_02120]</a> <a href="#">[SWS_REST_02121]</a> <a href="#">[SWS_REST_02122]</a> <a href="#">[SWS_REST_02123]</a> <a href="#">[SWS_REST_02124]</a> <a href="#">[SWS_REST_02125]</a> <a href="#">[SWS_REST_02126]</a> <a href="#">[SWS_REST_02127]</a> <a href="#">[SWS_REST_02128]</a> <a href="#">[SWS_REST_02129]</a> <a href="#">[SWS_REST_02130]</a> <a href="#">[SWS_REST_02131]</a> <a href="#">[SWS_REST_02132]</a> <a href="#">[SWS_REST_02133]</a> <a href="#">[SWS_REST_02134]</a> <a href="#">[SWS_REST_02135]</a> <a href="#">[SWS_REST_02136]</a> <a href="#">[SWS_REST_02137]</a> <a href="#">[SWS_REST_02138]</a> <a href="#">[SWS_REST_02139]</a> <a href="#">[SWS_REST_02140]</a> <a href="#">[SWS_REST_02141]</a> <a href="#">[SWS_REST_02142]</a> <a href="#">[SWS_REST_02143]</a> <a href="#">[SWS_REST_02144]</a> <a href="#">[SWS_REST_02145]</a> <a href="#">[SWS_REST_02146]</a> <a href="#">[SWS_REST_02147]</a> <a href="#">[SWS_REST_02148]</a> <a href="#">[SWS_REST_02149]</a> <a href="#">[SWS_REST_02150]</a> <a href="#">[SWS_REST_02151]</a> <a href="#">[SWS_REST_02152]</a> <a href="#">[SWS_REST_02153]</a> <a href="#">[SWS_REST_02154]</a> <a href="#">[SWS_REST_02155]</a> <a href="#">[SWS_REST_02156]</a> <a href="#">[SWS_REST_02157]</a> <a href="#">[SWS_REST_02158]</a> <a href="#">[SWS_REST_02343]</a> <a href="#">[SWS_REST_02344]</a> <a href="#">[SWS_REST_02345]</a> <a href="#">[SWS_REST_02346]</a> <a href="#">[SWS_REST_02347]</a> <a href="#">[SWS_REST_02348]</a> <a href="#">[SWS_REST_02389]</a> <a href="#">[SWS_REST_02390]</a> <a href="#">[SWS_REST_02391]</a> <a href="#">[SWS_REST_02392]</a> <a href="#">[SWS_REST_02393]</a> <a href="#">[SWS_REST_02403]</a> <a href="#">[SWS_REST_02404]</a> <a href="#">[SWS_REST_02405]</a> <a href="#">[SWS_REST_02406]</a> <a href="#">[SWS_REST_02407]</a> <a href="#">[SWS_REST_02409]</a> <a href="#">[SWS_REST_02410]</a> <a href="#">[SWS_REST_02411]</a> <a href="#">[SWS_REST_02412]</a> <a href="#">[SWS_REST_02413]</a> <a href="#">[SWS_REST_02414]</a> <a href="#">[SWS_REST_02418]</a> <a href="#">[SWS_REST_02419]</a> <a href="#">[SWS_REST_02420]</a> <a href="#">[SWS_REST_02421]</a>
<b>[RS_CM_00307]</b>	The Communication Management shall provide an Object Graph where each Object is strongly typed.	<a href="#">[SWS_REST_02036]</a> <a href="#">[SWS_REST_02037]</a> <a href="#">[SWS_REST_02038]</a> <a href="#">[SWS_REST_02039]</a> <a href="#">[SWS_REST_02040]</a> <a href="#">[SWS_REST_02041]</a> <a href="#">[SWS_REST_02042]</a> <a href="#">[SWS_REST_02043]</a> <a href="#">[SWS_REST_02044]</a> <a href="#">[SWS_REST_02045]</a> <a href="#">[SWS_REST_02046]</a> <a href="#">[SWS_REST_02047]</a> <a href="#">[SWS_REST_02048]</a> <a href="#">[SWS_REST_02049]</a> <a href="#">[SWS_REST_02050]</a> <a href="#">[SWS_REST_02051]</a> <a href="#">[SWS_REST_02052]</a> <a href="#">[SWS_REST_02053]</a> <a href="#">[SWS_REST_02054]</a> <a href="#">[SWS_REST_02055]</a> <a href="#">[SWS_REST_02056]</a> <a href="#">[SWS_REST_02057]</a> <a href="#">[SWS_REST_02058]</a> <a href="#">[SWS_REST_02059]</a>

Requirement	Description	Satisfied by
		[SWS_REST_02060] [SWS_REST_02061]
		[SWS_REST_02062] [SWS_REST_02063]
		[SWS_REST_02064] [SWS_REST_02065]
		[SWS_REST_02066] [SWS_REST_02067]
		[SWS_REST_02068] [SWS_REST_02069]
		[SWS_REST_02070] [SWS_REST_02071]
		[SWS_REST_02072] [SWS_REST_02073]
		[SWS_REST_02074] [SWS_REST_02075]
		[SWS_REST_02076] [SWS_REST_02077]
		[SWS_REST_02078] [SWS_REST_02079]
		[SWS_REST_02080] [SWS_REST_02081]
		[SWS_REST_02082] [SWS_REST_02083]
		[SWS_REST_02084] [SWS_REST_02085]
		[SWS_REST_02086] [SWS_REST_02087]
		[SWS_REST_02088] [SWS_REST_02089]
		[SWS_REST_02090] [SWS_REST_02091]
		[SWS_REST_02092] [SWS_REST_02093]
		[SWS_REST_02094] [SWS_REST_02095]
		[SWS_REST_02096] [SWS_REST_02097]
		[SWS_REST_02098] [SWS_REST_02099]
		[SWS_REST_02100] [SWS_REST_02101]
		[SWS_REST_02102] [SWS_REST_02103]
		[SWS_REST_02104] [SWS_REST_02105]
		[SWS_REST_02106] [SWS_REST_02107]
		[SWS_REST_02108] [SWS_REST_02109]
		[SWS_REST_02110] [SWS_REST_02111]
		[SWS_REST_02112] [SWS_REST_02113]
		[SWS_REST_02114] [SWS_REST_02115]
		[SWS_REST_02116] [SWS_REST_02117]
		[SWS_REST_02118] [SWS_REST_02119]
		[SWS_REST_02120] [SWS_REST_02121]
		[SWS_REST_02122] [SWS_REST_02123]
		[SWS_REST_02124] [SWS_REST_02125]
		[SWS_REST_02126] [SWS_REST_02127]
		[SWS_REST_02128] [SWS_REST_02129]
		[SWS_REST_02130] [SWS_REST_02131]
		[SWS_REST_02132] [SWS_REST_02133]
		[SWS_REST_02134] [SWS_REST_02135]
		[SWS_REST_02136] [SWS_REST_02137]
		[SWS_REST_02138] [SWS_REST_02139]
		[SWS_REST_02140] [SWS_REST_02141]
		[SWS_REST_02142] [SWS_REST_02143]
		[SWS_REST_02144] [SWS_REST_02145]
		[SWS_REST_02146] [SWS_REST_02147]
		[SWS_REST_02148] [SWS_REST_02149]
		[SWS_REST_02150] [SWS_REST_02151]
		[SWS_REST_02152] [SWS_REST_02153]
		[SWS_REST_02154] [SWS_REST_02155]

Requirement	Description	Satisfied by
		<a href="#">[SWS_REST_02156]</a> <a href="#">[SWS_REST_02157]</a> <a href="#">[SWS_REST_02158]</a> <a href="#">[SWS_REST_02343]</a> <a href="#">[SWS_REST_02344]</a> <a href="#">[SWS_REST_02345]</a> <a href="#">[SWS_REST_02346]</a> <a href="#">[SWS_REST_02347]</a> <a href="#">[SWS_REST_02348]</a> <a href="#">[SWS_REST_02389]</a> <a href="#">[SWS_REST_02390]</a> <a href="#">[SWS_REST_02391]</a> <a href="#">[SWS_REST_02392]</a> <a href="#">[SWS_REST_02393]</a> <a href="#">[SWS_REST_02403]</a> <a href="#">[SWS_REST_02404]</a> <a href="#">[SWS_REST_02405]</a> <a href="#">[SWS_REST_02406]</a> <a href="#">[SWS_REST_02407]</a> <a href="#">[SWS_REST_02409]</a> <a href="#">[SWS_REST_02410]</a> <a href="#">[SWS_REST_02411]</a> <a href="#">[SWS_REST_02412]</a> <a href="#">[SWS_REST_02413]</a> <a href="#">[SWS_REST_02414]</a> <a href="#">[SWS_REST_02418]</a> <a href="#">[SWS_REST_02419]</a> <a href="#">[SWS_REST_02420]</a> <a href="#">[SWS_REST_02421]</a>
<b>[RS_CM_00308]</b>	The Communication Management shall provide methods to read and manipulate the Object Graph	<a href="#">[SWS_REST_02039]</a> <a href="#">[SWS_REST_02040]</a> <a href="#">[SWS_REST_02041]</a> <a href="#">[SWS_REST_02042]</a> <a href="#">[SWS_REST_02043]</a> <a href="#">[SWS_REST_02044]</a> <a href="#">[SWS_REST_02045]</a> <a href="#">[SWS_REST_02046]</a> <a href="#">[SWS_REST_02047]</a> <a href="#">[SWS_REST_02048]</a> <a href="#">[SWS_REST_02049]</a> <a href="#">[SWS_REST_02050]</a> <a href="#">[SWS_REST_02051]</a> <a href="#">[SWS_REST_02052]</a> <a href="#">[SWS_REST_02053]</a> <a href="#">[SWS_REST_02054]</a> <a href="#">[SWS_REST_02055]</a> <a href="#">[SWS_REST_02056]</a> <a href="#">[SWS_REST_02057]</a> <a href="#">[SWS_REST_02058]</a> <a href="#">[SWS_REST_02059]</a> <a href="#">[SWS_REST_02060]</a> <a href="#">[SWS_REST_02061]</a> <a href="#">[SWS_REST_02065]</a> <a href="#">[SWS_REST_02066]</a> <a href="#">[SWS_REST_02067]</a> <a href="#">[SWS_REST_02068]</a> <a href="#">[SWS_REST_02069]</a> <a href="#">[SWS_REST_02070]</a> <a href="#">[SWS_REST_02071]</a> <a href="#">[SWS_REST_02072]</a> <a href="#">[SWS_REST_02073]</a> <a href="#">[SWS_REST_02074]</a> <a href="#">[SWS_REST_02075]</a> <a href="#">[SWS_REST_02076]</a> <a href="#">[SWS_REST_02081]</a> <a href="#">[SWS_REST_02082]</a> <a href="#">[SWS_REST_02083]</a> <a href="#">[SWS_REST_02084]</a> <a href="#">[SWS_REST_02085]</a> <a href="#">[SWS_REST_02086]</a> <a href="#">[SWS_REST_02087]</a> <a href="#">[SWS_REST_02088]</a> <a href="#">[SWS_REST_02092]</a> <a href="#">[SWS_REST_02093]</a> <a href="#">[SWS_REST_02094]</a> <a href="#">[SWS_REST_02095]</a> <a href="#">[SWS_REST_02096]</a> <a href="#">[SWS_REST_02097]</a> <a href="#">[SWS_REST_02098]</a> <a href="#">[SWS_REST_02099]</a> <a href="#">[SWS_REST_02100]</a> <a href="#">[SWS_REST_02104]</a> <a href="#">[SWS_REST_02105]</a> <a href="#">[SWS_REST_02106]</a> <a href="#">[SWS_REST_02107]</a> <a href="#">[SWS_REST_02108]</a> <a href="#">[SWS_REST_02109]</a> <a href="#">[SWS_REST_02110]</a> <a href="#">[SWS_REST_02111]</a> <a href="#">[SWS_REST_02112]</a> <a href="#">[SWS_REST_02113]</a> <a href="#">[SWS_REST_02114]</a> <a href="#">[SWS_REST_02115]</a> <a href="#">[SWS_REST_02116]</a> <a href="#">[SWS_REST_02117]</a> <a href="#">[SWS_REST_02118]</a> <a href="#">[SWS_REST_02119]</a> <a href="#">[SWS_REST_02120]</a> <a href="#">[SWS_REST_02121]</a> <a href="#">[SWS_REST_02122]</a> <a href="#">[SWS_REST_02123]</a>



Requirement	Description	Satisfied by
		[SWS_REST_02124] [SWS_REST_02125] [SWS_REST_02126] [SWS_REST_02131] [SWS_REST_02132] [SWS_REST_02133] [SWS_REST_02134] [SWS_REST_02135] [SWS_REST_02136] [SWS_REST_02137] [SWS_REST_02138] [SWS_REST_02143] [SWS_REST_02144] [SWS_REST_02145] [SWS_REST_02146] [SWS_REST_02147] [SWS_REST_02148] [SWS_REST_02149] [SWS_REST_02150] [SWS_REST_02151] [SWS_REST_02155] [SWS_REST_02156] [SWS_REST_02157] [SWS_REST_02158] [SWS_REST_02343] [SWS_REST_02344] [SWS_REST_02345] [SWS_REST_02346] [SWS_REST_02347] [SWS_REST_02348] [SWS_REST_02389] [SWS_REST_02390] [SWS_REST_02391] [SWS_REST_02392] [SWS_REST_02393] [SWS_REST_02403] [SWS_REST_02404] [SWS_REST_02405] [SWS_REST_02406] [SWS_REST_02407] [SWS_REST_02409] [SWS_REST_02410] [SWS_REST_02411] [SWS_REST_02412] [SWS_REST_02413] [SWS_REST_02414] [SWS_REST_02418] [SWS_REST_02419] [SWS_REST_02420] [SWS_REST_02421]
[RS_CM_00309]	The Communication Management shall provide a way to match requests to corresponding server handlers and vice versa.	[SWS_REST_02027] [SWS_REST_02028] [SWS_REST_02029] [SWS_REST_02030] [SWS_REST_02031] [SWS_REST_02033] [SWS_REST_02034] [SWS_REST_02035] [SWS_REST_02159] [SWS_REST_02160] [SWS_REST_02161] [SWS_REST_02162] [SWS_REST_02163]
[RS_CM_00310]	The Communication Management shall provide an interface to install request handlers.	[SWS_REST_01616] [SWS_REST_01617] [SWS_REST_01618] [SWS_REST_01624] [SWS_REST_02244]
[RS_CM_00311]	The Communication Management shall provide type aliases for abstraction of standard C++ components.	[SWS_REST_01001] [SWS_REST_01002] [SWS_REST_01003] [SWS_REST_01004] [SWS_REST_01005] [SWS_REST_01007] [SWS_REST_01011] [SWS_REST_01013] [SWS_REST_01014] [SWS_REST_01015] [SWS_REST_01016] [SWS_REST_01017] [SWS_REST_01018] [SWS_REST_01019] [SWS_REST_01020] [SWS_REST_02354] [SWS_REST_02355] [SWS_REST_02360]

Requirement	Description	Satisfied by
[RS_CM_00312]	The Communication Management shall provide HTTP/1.1 to transport RESTful requests and responses.	[SWS_REST_01801] [SWS_REST_01802] [SWS_REST_01803] [SWS_REST_01804] [SWS_REST_01805] [SWS_REST_01806] [SWS_REST_01807] [SWS_REST_01808] [SWS_REST_01816] [SWS_REST_01817] [SWS_REST_01818] [SWS_REST_01819] [SWS_REST_01820] [SWS_REST_01821] [SWS_REST_01822] [SWS_REST_01823] [SWS_REST_01824] [SWS_REST_01825] [SWS_REST_01826] [SWS_REST_01827] [SWS_REST_01828] [SWS_REST_01829] [SWS_REST_01830] [SWS_REST_01831] [SWS_REST_01832] [SWS_REST_01833] [SWS_REST_01834] [SWS_REST_01852] [SWS_REST_01859]
[RS_CM_00313]	The Communication Management shall provide a JSON-based serialization for the payload of RESTful requests and responses.	[SWS_REST_01851] [SWS_REST_01852] [SWS_REST_01853] [SWS_REST_01854] [SWS_REST_01855] [SWS_REST_01856] [SWS_REST_01857] [SWS_REST_01858] [SWS_REST_01859] [SWS_REST_01899]
[RS_CM_00314]	The Communication Management shall provide Websockets to establish event communication.	[SWS_REST_01810] [SWS_REST_01811] [SWS_REST_01812] [SWS_REST_01813] [SWS_REST_01814] [SWS_REST_01815]

## 7 Functional specification

### 7.1 General description

This chapter and chapter 8 specify an API design for a RESTful application framework for Adaptive AUTOSAR. Traditionally RESTful services are applications of the mobile world where resource constraints are not as severe as in the automotive domain. Neither clients, servers, transport protocols, formats and last but not least RESTful applications on top of all this are usually particularly geared towards low and predictable resource usage. Providing RESTful services in an environment with strict quality requirements such as low and deterministic memory and processing time is therefore particularly challenging. `ara::rest` is specifically designed for this use-case.

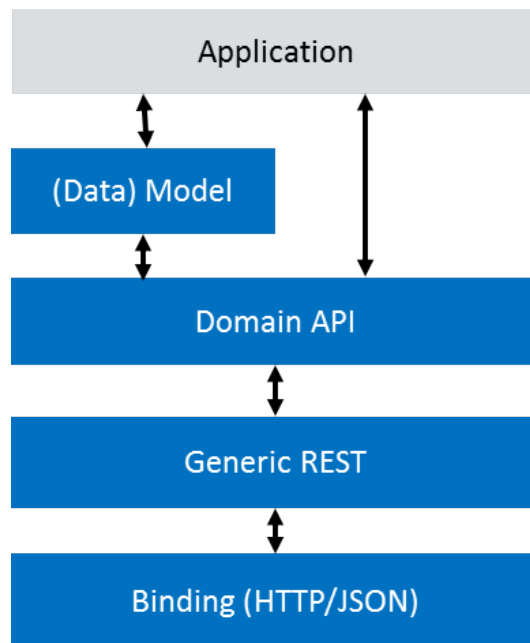


Figure 7.1: Typical RESTful service stack

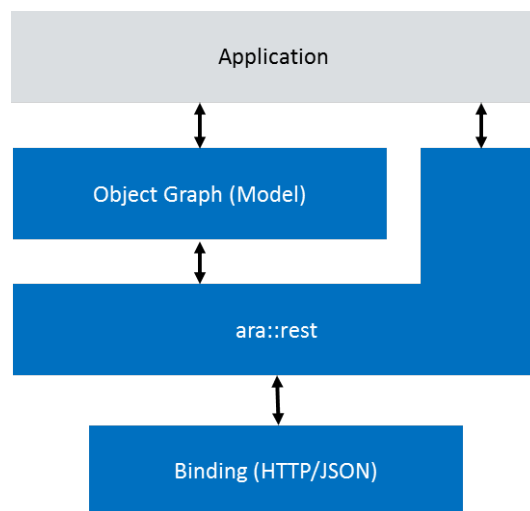
A typical RESTful API stack is depicted in figure 7.1. In this, `ara::rest` provides the lower to stack elements and a generic data representation from which elements like a domain-specific API and a domain-specific data model can be constructed.

#### 7.1.1 Architectural concepts

The `ara::rest` framework is modular in that it enables developers to access different layers involved in RESTful message transactions directly. This is in contrast to `ara::com` whose focus is to provide to the developer a traditional function call interface and to hide all details of the transactions beyond this point. This shifts some technical effort but also a lot of control away from the developer into a monolithic black box. This is a feasible design choice since there is a clear and very simple notion of what such an API constitutes: C-style functions.

As opposed to this RESTful APIs are an entirely unclear design target. Specific designs range from entirely data driven access up to simulation of RPC interfaces. In `ara::com` interfaces are precisely defined and message payloads are known to the byte. In RESTful APIs in general this is not necessarily the case. Control of a service is exercised entirely by means of (often dynamic) data. Specific functionality of a service is triggered by any combination of URI and data payload in the messages themselves and it is up to the service to interpret this data and turn this back into actual side-effects. To make the contrast clear, in `ara::com` a "meta contract" of the API exists which allows a high degree of optimization since the form of interaction is "function calls with mostly static data".

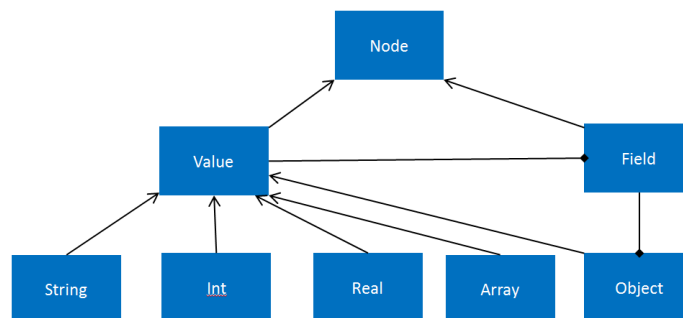
For RESTful APIs such a basic concept does not exist. Therefore RESTful API design is a two-step process of first designing a specific instance of an API which defines a class of services that agree on the general rules of interaction and second defining specific services by using the tools enabled by those rules. To make this more hands-on, there might exist a vendor-specific implementation of `ara::rest`, there might exist a OEM-specific RESTful API on top of this and there might exist a domain-specific service built by means of this RESTful API. `ara::rest` specifically addresses these requirements by incorporating a modular design which supports developers at the level of API as well as service design. The following diagram illustrates its general design. It depicts how a service application is composed in `ara::rest`:



**Figure 7.2: `ara::rest` component stack**

All `ara::rest` communication is performed through protocol-specific bindings. By default `ara::rest` abstracts from these protocol details. The generic REST layer of `ara::rest` only provides three fundamental abstractions: a tree-structured message payload (Object Graph), a URI and a request method (like GET or POST known from HTTP). From these basic primitives domain-specific RESTful APIs can be composed (such as W3C ViWi) which defines a concrete high-level protocol for interaction via object graphs, URIs and methods. Its purpose is to define the rules for access into a domain-specific data model and to provide an abstract (C++) API to an application.

The unified data representation, called Object Graph Model (OGM) comprises of a very limited set of data primitives which reflects the simplicity of usual RESTful communication artifacts (such as JSON message payloads).



**Figure 7.3: Class diagram of the Object Graph Model**

### 7.1.2 Design Scope

This specification covers the topics support-library, client/server endpoints and object graph data structure in detail. It does not yet cover topics related to mapping the AUTOSAR meta model onto `ara::rest`. Security features like a built-in support for authentication are not conclusively decided upon.

### 7.1.3 Design objectives

`ara::rest` has been carefully designed to enable implementations with deterministic timing, low resource footprint and low latency while being simple enough to reason about aspects of software safety. This sets it apart from existing C++ RESTful service frameworks. In the following a brief overview of its key design objectives are provided.

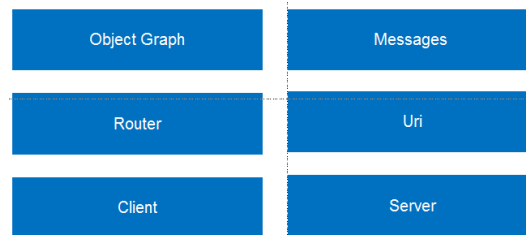
- **Component-based:** `ara::rest` is a construction kit to build RESTful APIs. It does not directly define a specific API upon which services can be realized. Instead it supports its construction by providing basic components that enable efficient implementations with a high degree of resource control to meet software safety requirements.
- **Standard compatibility:** `ara::rest` abstracts from standard C++ for some key components. On the one hand some critical features are not yet sufficiently mature, are not yet part of the standard or are incomplete. On the other hand some standard components inhibit safe and efficient implementations when it comes to resource control and asynchronous programming. Therefore `ara::rest` defines a small set of type aliases which allow for an implementation to provide custom components if the platform does not support them yet, or for enhanced safety and efficiency. `ara::rest` is carefully designed to enable the use of standard C++ com-

ponents for quick deployment (abstraction then boils down to simple type aliases) but also enables an implementation to replace all of them by custom components. It is not a wrapper.

- Asynchronous programming model: `ara::rest` is designed for asynchronous I/O as its default programming model. Consequently implementations can be highly responsive without the need for multi-threading. `ara::rest` does not enforce a single-thread execution model though. It is up to a concrete implementation how computing resources are being exploited.
- Task-based execution model: `ara::rest` is designed for low latency and low resource usage by defining a task-based execution model which complements the asynchronous programming model. `ara::rest` tasks abstract from traditional threading as they leave unspecified whether a task corresponds to a POSIX thread one-to-one, or whether it corresponds to a lightweight user-level thread abstraction or none of the above.
- User-defined memory management: To complement the stack-based resource model, all dynamic data structures of `ara::rest` support a custom allocator model to support safety and efficiency at all levels. (Allocators are compatible with C++17 PMR.) They complement the abstraction of certain critical standard C++ components.
- Unified data abstraction: The framework provides a unified API to handle message payloads and optionally the underlying data model of services. `ara::rest` communicates via tree-structured data called object graphs. It provides all necessary abstractions to build, traverse and dissect object graphs at C++ level. It reflects the capabilities of JSON to some extent for easy serialization. However it is designed with two additional objectives in mind. First, object graphs are designed for `ara::rest` specifically. Although they can be used independently of other `ara::rest` components, they fit into the general resource model. Second, object graphs can be specialized such that abstract objects can be derived from the AUTOSAR meta model. This lifts the handling of message payloads from the handling of primitives data types such as int, strings or "records" to the level of handling Doors, Windows and Batteries, for example. Nevertheless, a domain-specific `ara::rest` server that replies with a "Battery" object in its payload can communicate with a client which has no such notion and vice versa. This is key to enable communication with non-AUTOSAR RESTful clients and services. In addition this abstraction simplifies the mapping into Classic AUTOSAR since abstract object graphs precisely map to static struct representations required by SOME/IP [6].

#### 7.1.4 Basic Components

`ara::rest` can roughly be subdivided into functional blocks as depicted in figure 7.4.



**Figure 7.4: Basic components comprising ara::rest**

The Object Graph is a protocol-independent tree-like data structure which is the cornerstone of all ara::rest communication. Its purpose is to map to a protocol format such as JSON as well as to C structs. This maximizes compatibility with non-ARA communication peers and Classic AUTOSAR. Object graphs are transmitted in messages which abstract completely from a concrete underlying protocol binding.

Messages encapsulate the entire context of a request/reply communication cycle in the asynchronous programming model of ara::rest.

The routing concept provides a means to map requests (including request method and URI) onto user-defined handler functions. Routing is the cornerstone to lift abstraction from generic REST into a specific kind of RESTful API.

Uri is a generic RFC-compliant, memory-efficient and exception-safe URI representation. ara::rest provides so-called (network) endpoints for server and client communication which both provide a comparable degree of resource control.

The entire framework design is strictly geared towards maximal resource control. All computations and allocations can be strictly controlled and customized to the precise needs of an application (deployment).

## 7.2 Support Functionality

The ara::rest framework requires abstractions from some standard C++ components that are particularly critical in terms of resource control and safety. This provides much greater control of resources such as allocation of memory and computing time and allows to fix many of the existing problems with standard C++ components. However all of these abstractions indeed have an existing counterpart defined in the C++ standard (C++11 or C++17). In AUTOSAR some of these C++ components already has an abstraction in the Adaptive Core Types, [?, ]. In these cases the ara::core types shall be used.

**[SWS\_REST\_01001] Use of support type aliases** [ The support type aliases specified in this section may either alias standard C++ library types as available, or alias custom but API-compliant implementations, such as ara::core::types. ] ([RS\\_CM\\_00311](#))

**[SWS\_REST\_01002] Duration type** [ A type alias `ara::rest::duration_t` shall exist to express time spans of precision of at least microseconds. ] ([RS\\_CM\\_00311](#))

**[SWS\_REST\_01003] Pointer type** [ A type `ara::rest::Pointer` shall exist that models the precise semantics of `std::unique_ptr`. ]([RS\\_CM\\_00311](#))

**[SWS\_REST\_01004] Basic string type** [ `ara::core::String` shall be used for `std::basic_string` semantics. ]([RS\\_CM\\_00311](#))

**[SWS\_REST\_01005] String type** [ `ara::core::String` shall be used for `std::string` semantics. ]([RS\\_CM\\_00311](#))

**[SWS\_REST\_01007] Basic String View** [ `ara::core::StringView` shall be used for `std::basic_string_view` semantics. ]([RS\\_CM\\_00311](#))

**[SWS\_REST\_01011] String View type** [ `ara::core::StringView` shall be used for `std::string_view` semantics. ]([RS\\_CM\\_00311](#))

**[SWS\_REST\_01013] Function type** [ A type alias `template` shall exist that models the precise semantics of `std::function`. The following type alias template may be used:

```
1 template<typename T>
2 using Function = std::function<T>;
```

]([RS\\_CM\\_00311](#))

**[SWS\_REST\_01014] Task type** [ A type alias `template` shall exist that models the precise semantics of C++14 `std::future`. `ara::core::Future` shall be used. The following type alias template may be used:

```
1 template<typename T>
2 using Task = ara::core::Future<T>;
```

]([RS\\_CM\\_00311](#))

**[SWS\_REST\_01015] Function continuation-passing** [ In particular, `ara::rest::Task` shall support continuations by means of `ara::core::Future::then`. ]([RS\\_CM\\_00311](#))

**[SWS\_REST\_01016] Iterator ranges** [ A type `IteratorRange` shall be defined as a thin wrapper around a pair of iterators according to API `ara::rest::IteratorRange`. ]([RS\\_CM\\_00311](#))

**[SWS\_REST\_01017] Allocator** [ A type `ara::rest::Allocator` shall exist that models the precise semantics of C++17 `std::pmr::memory_resource`. The following type alias may be used, if available:

```
1 using Allocator = std::experimental::pmr::memory_resource;
```

]([RS\\_CM\\_00311](#))

**[SWS\_REST\_01018] Allocator Adapter** [ For compatibility, a standard allocator adapter type shall be provided that models the precise semantics of C++17 `std::pmr::polymorphic_allocator`. The following type alias template may be used, if available:

```
1 template<typename T>
```



```
2 using StdAllocator = std::experimental::pmr::polymorphic_allocator<T>;
```

](RS\_CM\_00311)

**[SWS\_REST\_01019] NewDelete Allocator** [ For compatibility, a default allocator type shall be provided that models the precise semantics of C++17 `std::pmr::new_delete_resource`. This type shall allocate and free memory via default new and delete operators internally. The following type alias template may be used, if available:

```
1 template<typename T>
2 using NewDeleteAllocator = std::experimental::pmr::new_delete_resource;
```

](RS\_CM\_00311)

**[SWS\_REST\_01020] Get and Set Default Allocator** [ There shall exist two functions to request () and reset () a global default allocator instance. The following type alias template may be used, if available:

```
1 Allocator* GetDefaultAllocator() noexcept;
2 Allocator* SetDefaultAllocator(Allocator*) noexcept;
```

](RS\_CM\_00311)

## 7.3 URI

`ara::rest::Uri` is a universal container for RFC 3986 [5] compliant identifiers and is specifically designed with resource control in mind. `ara::rest` messaging is essentially based on two kinds of payloads: object graph and URI. In simple transactions, URIs potentially yield larger memory footprints than the message payloads themselves. An efficient URI type is therefore required.

**[SWS\_REST\_01101] URI API** [ This type shall at least provide the interface along with its functional description as specified in `ara::rest::Uri`. ](RS\_CM\_00300, RS\_CM\_00304)

**[SWS\_REST\_01102] URI UTF-8** [ `ara::rest::Uri` shall support UTF-8 in percent-encoded form (see RFC 3986 [5]). Otherwise no awareness of a particular character encoding is required. ](RS\_CM\_00300, RS\_CM\_00304)

**[SWS\_REST\_01103] URI String conversion** [ All `ara::rest::Uri` member functions that return string objects shall return a textual representation in non-percent-encoded form. All forms of `ara::rest::ToString` shall return a percent or non-percent encoded form, depending the flag specified as its function argument. ](RS\_CM\_00300)

**[SWS\_REST\_01104] URI Mutability** [ `ara::rest::Uri` shall be immutable. New versions of URI objects can only be created via `ara::rest::Uri::Builder`. ](RS\_CM\_00300)

**[SWS\_REST\_01105] URI Exceptions** [ `ara::rest::Uri` shall not throw. ]  
([RS\\_CM\\_00300](#))

**[SWS\_REST\_01106] URI Maximal Length** [ `ara::rest::Uri` shall keep a constant length limit of 2048 bytes including potential string terminators and shall be provided via `ara::rest::Uri::LENGTH_MAX`. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01107] Builder API** [ `ara::rest::Uri::Builder` shall at least provide the API defined in its API specification. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01108] Builder Exceptions** [ `ara::rest::Uri::Builder` may throw. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01109] URI Normalization** [ `ara::rest::Normalize` shall normalize `ara::rest::Uri`s according to RFC 3986 [5]. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01110] URI Resolve** [ `ara::rest::Resolve` shall resolve `ara::rest::Uri`s according to RFC 3986 [5]. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01111] URI Relativize** [ `ara::rest::Relativize` shall relativize `ara::rest::Uri`s according to RFC 3986 [5]. ]([RS\\_CM\\_00300](#))

## 7.4 UUID

**[SWS\_REST\_01201] UUID type** [ To represent UUIDs according to RFC 4122 [7], `ara::rest::Uuid` shall be defined according to its API specification. ]  
([RS\\_CM\\_00300](#))

**[SWS\_REST\_01203] UUID exceptions** [ `ara::rest::Uuid` shall not throw exceptions during construction. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_02428] UUID Construction Semantics** [ UUID constructor shall construct an UUID with zero values. UUID generation is provided by static helper functions `ara::rest::Uuid::MakeV1`, `ara::rest::Uuid::MakeV3`, `ara::rest::Uuid::MakeV4`, `ara::rest::Uuid::MakeV5`. ]([RS\\_CM\\_00300](#))

## 7.5 Endpoints

`ara::rest` provides two (network) endpoints - `ara::rest::Client` and `ara::rest::Server` - that manage I/O and resources of RESTful communication. This section specifies the common requirements for both client and server. Specific requirements for the `ara::rest::Client` and the `ara::rest::Server` can be found in Chapter 7.6 and 7.7.

**[SWS\_REST\_01301] Endpoints** [ `ara::rest` endpoints shall have no notion of the underlying communication protocol. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01302] Endpoint RESTful communication paradigm** [ `ara::rest` endpoints may violate the REST-principle by maintaining state about their respective peers internally, depending on the requirements of the underlying transport protocol used in peer communication. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01304] Endpoint I/O abstraction** [ All communication of an application with a remote peer shall be performed via `ara::rest::RequestMethod`, `ara::rest::Uri` and the Object Graph Model. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_01305] Request Methods** [ All accesses into an application data model shall be performed with a small, predefined set of request methods that denote whether an access shall be reading (GET), creating (POST), creating or changing (PUT), deleting (DELETE) or introspection (OPTIONS). Their precise semantics is application-defined. An enumeration `ara::rest::RequestMethod` shall exist which represents these respective access methods as well as `ara::rest::operator|` for recombination. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01306] SubscriptionState** [ To facilitate events, an implementation shall provide an enumeration `ara::rest::SubscriptionState` whose elements shall have the following semantics: An event subscription "subscribed" if client and server successfully performed a subscription handshake. It is "canceled" if both peers agreed to cancel the subscription. If the state is neither "subscribed" nor "canceled" it shall be "invalid". ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01307] EventPolicy** [ Events in `ara::rest` can be subscribed to in different modes. An implementation shall provide an enumeration `ara::rest::EventPolicy` that shall provide the following semantics for event subscriptions:

- `ara::rest::EventPolicy::kTriggered`: An event is only triggered upon explicit request through `ara::rest::ServerEvent::Notify`. The given time bound limits the number of requests per time frame to at most once.
- `ara::rest::EventPolicy::kPeriodic`: An event is triggered in constant time frames. Explicit triggers of events have not effect.
- `EventPolicy::kTriggered|EventPolicy::kPeriodic`: An event is triggered the end of each given time frame, if and only if a trigger has been issued within this frame at least once. Additional triggers shall have no effect.

]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01308] ShutdownPolicy** [ To facilitate a well-defined shutdown of an endpoint, an implementation shall provide an enumeration `ara::rest::ShutdownPolicy` which shall provide the general semantics as described subsequently. Specific semantic differences between client and server are specified below.

- `ara::rest::ShutdownPolicy::kForced`: A forced shutdown shall cancel (terminate) all transactions as fast as possible and does not block the caller for

"unreasonably" (implementation-defined) long period of time. During a forced shutdown, further network I/O is not allowed. A forced shutdown shall not allocate new memory resources. Apart from this semantics of these policies are implementation defined.

- `ara::rest::ShutdownPolicy::kGraceful`: Endpoints may shut down "gracefully", which shall allow all ongoing transactions to finish while blocking the caller. Precise semantics of these policies are implementation defined.

]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01312] Reply Header** [ A `ara::rest::ReplyHeader` shall exist which shall provide mutable access to `ara::rest::RequestMethod` and `ara::rest::Uri` of a `ara::rest::Reply` or `ara::rest::ServerReply`. The precise semantics of `ara::rest::ReplyHeader::GetStatus` and `ara::rest::ReplyHeader::SetStatus` as well as `ara::rest::ReplyHeader::GetUri` and `ara::rest::ReplyHeader::SetUri` is protocol-dependent. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01313] Request Header** [ A `ara::rest::RequestHeader` shall exist which shall provide mutable access to `ara::rest::RequestMethod` and `ara::rest::Uri` of a `ara::rest::Request` or `ara::rest::ServerRequest`. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01314] Instance identifier** [ A `ara::rest::InstanceIdentifier` shall relate to the corresponding port in the meta model of the `SoftwareComponentType` over the `InstanceRef`. Note that the port name of the meta model alone is not sufficient to clearly identify it in its final instantiation, where the same component implementation might be instantiated multiple times in the code and then eventually started multiple times in different processes. This is a generic problem which also exists on `ara::com`. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01315] RestServiceInterface** [ The `RestServiceInterface` shall contain the model of a specific RESTful service. Note that further modeling of how such a service looks like is already present in `TPS_Manifest` but it is currently not applied in this SWS document. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01316] Network binding** [ `RestServiceInterface` implemented by `ara::rest::Client` and `ara::rest::Server` shall be bound to a specific network binding. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01317] TLS secure channel for RESTful communication** [ A TLS secure channel shall be used for RESTful communication if a `TlsSecureComProps` instance is referenced by `RestHttpPortPrototypeMapping`. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01318] TLS secure channel for event communication** [ A TLS secure channel shall be used for event-based communication if a `TlsSecureComProps` instance is referenced by `RestHttpPortPrototypeMapping`. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

## 7.6 Client

`ara::rest` requires the existence of a type `ara::rest::Client` which represents the client network endpoint for RESTful communication. A client is not bound to a particular remote endpoint.

**[SWS\_REST\_01401] Client Resources** [ `ara::rest::Client` maintains all resources related to communication with a peer. Upon destruction, all such resources shall be released. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01402] Client Interface** [ An `ara::rest::Client` shall at least provide the interface as specified in its API specification and according to the detailed semantics specified there. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01403] Client Instances** [ `ara::rest::Client` shall not be copyable. It shall be movable. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01404] Client Multiplicity** [ `ara::rest::Client` shall not be bound to one particular peer. As such, a client is not bound to a particular server instance. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01405] Client Asynchronicity** [ `ara::rest::Client` shall be able to operate according to its semantic requirements without employing multi-threading. Consequently, a user shall be aware of the fact that invoking blocking functions such as `ara::rest::Task<T>::wait()` may degrade service quality. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01406] Client Concurrency** [ It is implementation-defined whether `ara::rest::Client` is operating concurrently. Consequently, a user shall be aware of the fact that invoking blocking functions may degrade service quality. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01407] Client Construction** [ `ara::rest::Client` shall be provided with a unique identifier that selects an implementation-defined configuration record. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01408] Client Startup** [ `ara::rest::Client` shall be ready for transmission to remote hosts if construction succeeded. In particular, a client may throw if construction fails. Otherwise, an error shall be indicated via `ara::rest::Client::GetError`. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01409] Client Shutdown** [ `ara::rest::Client` shall be stopped via `ara::rest::Client::Stop` according to the semantics specified in its API specification and SWS\_REST\_01308. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01410] Client Stop** [ `ara::rest::Client::Stop` shall be thread-safe. After it has been invoked, calling `ara::rest::Client::Send`, `ara::rest::Client::Subscribe` or `ara::rest::Client::Stop` shall have no effect. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01411] Client Stop Task** [ `ara::rest::Client::Stop` returns a `Task` which shall only complete once shutdown succeeded. It may throw otherwise. ] (*RS\_CM\_00300, RS\_CM\_00301*)

**[SWS\_REST\_01412] Client Sending** [ `ara::rest::Client::Send` shall transmit an `ara::rest::Request` to a peer specified via the given request URI. The call shall never block even for multiple requests to the same peer. ] (*RS\_CM\_00300, RS\_CM\_00301*)

**[SWS\_REST\_01413] Client Peer Addressing** [ `ara::rest::Client::Send` shall transmit a request to the peer specified by the uri authority part via `ara::rest::Uri::GetHost` and `ara::rest::Uri::GetPort`. If not such information is available a client shall fall back to the host address and port given by the network binding configuration. ] (*RS\_CM\_00300, RS\_CM\_00301*)

**[SWS\_REST\_01414] Client Events** [ An implementation shall provide `ara::rest::Client::Subscribe` to create `ara::rest::Event` instances which represent a single event subscription. `ara::rest::Client::Subscribe` shall never block. ] (*RS\_CM\_00300, RS\_CM\_00301*)

**[SWS\_REST\_01415] Client Event Uri** [ The URI passed to `ara::rest::Client::Subscribe` denotes an entity to subscribe to. URI query parameters shall be taken into account. ] (*RS\_CM\_00300, RS\_CM\_00301*)

**[SWS\_REST\_01416] Client Event Policy** [ An implementation shall honor the event policy passed to according to [SWS\_REST\_01307] along with the time bound specified. ] (*RS\_CM\_00300, RS\_CM\_00301*)

**[SWS\_REST\_01417] Client Event Notification** [ An instance of `ara::rest::Client::NotificationHandlerType` shall be passed to `ara::rest::Client::Subscribe` for the asynchronous reception of event notification. ] (*RS\_CM\_00300, RS\_CM\_00301*)

**[SWS\_REST\_01418] Client Event Status** [ An instance of `ara::rest::Client::SubscriptionStateHandlerType` may be passed to `ara::rest::Client::Subscribe` for the asynchronous changes to the event status. ] (*RS\_CM\_00300, RS\_CM\_00301*)

**[SWS\_REST\_01419] Client Exception** [ `ara::rest::Client` itself may throw during construction and during regular operation only for errors that leave the client in an undefined state. ] (*RS\_CM\_00300, RS\_CM\_00301*)

**[SWS\_REST\_01420] Client Error** [ All errors not leaving a client in an undefined state shall be indicated via `ara::rest::Client::GetError` and `ara::rest::Client::ObserveError`. In particular this concerns all I/O related errors. The type `ara::core::ErrorCode` shall be used for error codes. ] (*RS\_CM\_00300, RS\_CM\_00301*)

**[SWS\_REST\_01421] Client Request** [ `ara::rest::Request` shall be non-copyable and shall maintain all resources related to issue a data request to a peer. In particular, the lifetime of a request object and the lifetime of the objects passed to it for

transmission shall be independent. This implies copying or moving. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01422] Client Reply** [ `ara::rest::Reply` shall be non-copyable and shall maintain all resources related to reception of data from a peer. The lifetime of all resources referenced by a reply object is bound to the lifetime of the reply itself. This implies copying or moving. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

## 7.7 Server

`ara::rest` requires the existence of a type `ara::rest::Server` which represents the server network endpoint for RESTful communication along with its components.

**[SWS\_REST\_01501] Server Resources** [ `ara::rest::Server` maintains all resources related to communication with a peer. Upon destruction, all such resources shall be released. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01502] Server Interface** [ An `ara::rest::Server` shall at least provide the interface as specified in the API specification and according to the detailed semantics specified there. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01503] Server Instances** [ `ara::rest::Server` shall not be copyable. It shall be movable. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01504] Server Multibinding** [ `ara::rest::Server` shall be able to handle multiple transport protocol bindings concurrently. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01505] Server Multiplicity** [ `ara::rest::Server` shall be able to handle multiple peers concurrently over potentially multiple transport protocols. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01506] Server Asynchronicity** [ `ara::rest::Server` shall be able to operate according to its semantic requirements without employing multi-threading. Consequently, a user shall be aware of the fact that invoking blocking functions such as `ara::rest::Task<T>::wait()` may degrade service quality. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01507] Server Concurrency** [ It is implementation-defined whether `ara::rest::Server` is operating concurrently. Consequently, an application developer shall be aware of the fact that invoking blocking functions may degrade service quality. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01508] Server Construction** [ `ara::rest::Server` shall be provided with a unique identifier that selects an implementation-defined configuration record. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01509] Server Startup** [ `ara::rest::Server` shall not be operational until `ara::rest::Server::Start` is invoked explicitly. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01510] Server Startup Policies** [ To facilitate a well-defined startup of a server endpoint, an implementation shall provide an enumeration `ara::rest::StartupPolicy` which shall provide the following general semantics:

- `StartupPolicy::kDetached`: the server endpoint shall not block its calling context during startup and regular operation.
- `StartupPolicy::kAttached`: the server endpoint shall block its calling context during startup and regular operation as long as it is explicitly shut down.

] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01511] Server Startup Task** [ `ara::rest::Server::Start` returns a `Task` which shall only complete once startup succeeded. It may throw otherwise. ] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01512] Server Shutdown** [ `ara::rest::Server` shall be stopped via `ara::rest::Server::Stop` according to the semantics specified in its API specification and [\[SWS\\_REST\\_01308\]](#). ] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01513] Server Restart** [ `ara::rest::Server` shall leave its instances in a well-defined state after shutdown such that a subsequent startup remains feasible. ] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01514] Server Stop** [ `ara::rest::Server::Stop` shall be thread-safe. After it has been invoked, no new requests shall be accepted. ] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01515] Server Stop Task** [ `ara::rest::Server::Stop` returns a `Task` which shall only complete once shutdown succeeded. It may throw otherwise. ] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01516] Server User Message Notification** [ `ara::rest::Server` shall invoke the user-defined handler function specified in its destructor at least as early as a valid message header has been received. It is implementation-defined whether the handler is activated only when the entire message has been received. ] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01517] Server Event Subscriptions** [ `ara::rest::Server` shall accept event subscriptions without user-intervention. It shall inform the application of new subscriptions or subscription state changes via `ara::rest::Server::ObserveSubscriptions`. ] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01518] Server Event Data Request** [ Depending on the subscription parameters, event notifications shall be issued at certain points in time. `ara::rest::Server` shall issue a regular GET request to the application for the URI and the respective query parameters as supplied by the subscriber. To the application



such a request shall be indistinguishable from regular requests. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01519] Server Event Object** [ Each subscription shall be represented by a unique `ara::rest::ServerEvent` object which shall maintain all resources related to this subscription. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01522] Server Event Object Destruction** [ Upon destruction of a `ara::rest::ServerEvent` object the corresponding event subscription shall be terminated instantly, unilaterally and all managed server-side resources shall be released. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01523] Server Event Object Subscription Cancellation** [ `ara::rest::ServerEvent::SetSubscriptionState` called with parameter `ara::rest::SubscriptionState::kCanceled` shall cancel the subscription in accordance to the rules of the underlying protocols implementing the event mechanism. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01524] Server Event Object Re-Subscription** [ Once `ara::rest::ServerEvent::SetSubscriptionState` with parameter `ara::rest::SubscriptionState::kCanceled` is called, resubscription on the same object shall not be allowed. Each subscription shall yield to a unique event object. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01525] Server Event Object Notify** [ `ara::rest::ServerEvent::Notify` shall notify its corresponding `ara::rest::Server` instance of potential updates. The notifications are only triggered for URIs with an active subscription. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01526] Server Event Object Notify Semantics** [ Following a call to `ara::rest::ServerEvent::Notify`, it is the responsibility of `ara::rest::Server` to decide whether an actual event notification message shall be transmitted to the subscriber. The actual data that should be transmitted can either be received in the `ara::rest::ServerEvent::Notify` call, or obtained from the related URI. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01527] Server Event Observation** [ `ara::rest::Server::ObserveSubscriptions` shall be used by an application to register user-defined handler functions to be called on new subscriptions and state changed to existing subscriptions. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01528] Server Event Creation** [ An instance of `ara::rest::Server::SubscriptionHandlerType` that is registered via `ara::rest::Server::ObserveSubscriptions` is called by `ara::rest::Server` for each newly accepted subscription. It is the responsibility of the server object to create an instance of `ara::rest::ServerEvent` and pass it to the user via the handler function. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

**[SWS\_REST\_01529] Server Event Status Changes** [ An instance of `ara::rest::Server::SubscriptionStateHandlerType` that is regis-

tered via `ara::rest::Server::ObserveSubscriptions` is called by `ara::rest::Server` for each detected state change of a subscription according to `ara::rest::SubscriptionState`. [\]\(RS\\_CM\\_00300, RS\\_CM\\_00301\)](#)

**[SWS\_REST\_01530] Server Exception** `ara::rest::Server` itself may throw during construction and during regular operation only for errors that leave the server in an undefined state. [\]\(RS\\_CM\\_00300, RS\\_CM\\_00301\)](#)

**[SWS\_REST\_01531] Server Error** `ara::rest::Server::GetError` and `ara::rest::Server::ObserveError`. In particular this concerns all I/O related errors. The type `ara::core::ErrorCode` shall be used for error codes. [\]\(RS\\_CM\\_00300, RS\\_CM\\_00301\)](#)

**[SWS\_REST\_01532] Server Requests and Responses** `ara::rest::ServerRequest` and `ara::rest::ServerReply` shall be instantiated by `ara::rest::Server` for each received request message. Both objects shall manage all resource related to the current transaction. An application shall neither construct or destroy these objects. [\]\(RS\\_CM\\_00300, RS\\_CM\\_00301\)](#)

**[SWS\_REST\_01533] Server Request Semantics** A reference of `ara::rest::ServerRequest` shall be passed to the user-defined handler function as soon as a valid message header has been accepted. Access to message payloads shall be asynchronous via `ara::rest::ServerRequest::GetObject` or `ara::rest::ServerRequest::ReleaseObject` respectively. [\]\(RS\\_CM\\_00300, RS\\_CM\\_00301\)](#)

**[SWS\_REST\_01534] Server Reply Semantics** A reference of `ara::rest::ServerReply` may be passed to the user-defined handler function along with a reference to its corresponding `ara::rest::ServerRequest`. The instance shall be initialized such that it represents an empty reply message. [\]\(RS\\_CM\\_00300, RS\\_CM\\_00301\)](#)

**[SWS\_REST\_01535] Server Reply Send Semantics** `ara::rest::ServerReply::Send` shall trigger the transmission of a message to sender of the respective request. Upon destruction a `ara::rest::ServerReply` shall call `Send` implicitly. [\]\(RS\\_CM\\_00300, RS\\_CM\\_00301\)](#)

**[SWS\_REST\_01536] Server Reply Multiple Send Semantics** `ara::rest::ServerReply::Send` shall not be called multiple times. [\]\(RS\\_CM\\_00300, RS\\_CM\\_00301\)](#)

**[SWS\_REST\_01537] Server Reply Redirect Semantics** `ara::rest::ServerReply::Redirect` shall issue a protocol-dependent client redirection which shall cause a client to repeat the request to the endpoint indicated via `ara::rest::ReplyHeader::SetUri`. [\]\(RS\\_CM\\_00300, RS\\_CM\\_00301\)](#)

**[SWS\_REST\_01538] Server Reply Send/Redirect Interaction** `ara::rest::ServerReply::Send` shall not be called after

`ara::rest::ServerReply::Redirect`, and vice versa. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

## 7.8 Routing

Routing is multiplexing of server requests depending on the valuation of `ara::rest::RequestMethod` and `ara::rest::Uri`. It connects RESTful API accesses with an underlying execution or data model. In the following the components for routing are specified bottom-up.

### 7.8.1 Patterns

**[SWS\_REST\_01601] Pattern API** [ An implementation shall provide a type `ara::rest::Pattern` that satisfies at least the interface and basic semantics as defined in the API specification below. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01602] Pattern Syntax** [ `ara::rest::Pattern` represents a pattern string to match against `ara::rest::Uri::Path` instances. A pattern string may be composed of all valid URI path characters as well as characters “\*” and “\*\*” (wildcards). ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01603] Pattern General Wildcard semantics** [ Wildcards shall match URI path segments, not general string characters in an URI. In other words, URI path segment delimiters “/” restrict matching. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01604] Pattern Single Wildcard semantics** [ Wildcard character “\*” shall match exactly a single URI path segment.

Example: Pattern `/foo/*/bar` shall match URI path `/foo/baz/bar`. It shall not match URI path `/foo/baz/bab/baz` nor `/foo/bar`. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01605] Pattern Double Wildcard semantics** [ Wildcard characters “\*\*” shall match any number of path segments (including none).

Example: Pattern `/foo/**` shall match URI path `/foo`, `/foo/baz` and `/foo/baz/bar`. It shall not match URI paths that do not begin with `/foo`. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01606] Pattern Comparability** [ `ara::rest::Pattern` shall be equal-to, unequal-to and less-than comparable. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01607] Pattern Order Criterion** [ `ara::rest::Pattern` shall be less-than comparable and form a lexicographic order in which URI path segments are considered “characters” in left-to-right order.

Example: It holds that “car” < “car/window” which means “car” before “/car/window”. For wildcards the order “\*\*” < “\*” < “anything else” holds. ]([RS\\_CM\\_00300](#))

## 7.8.2 Match

**[SWS\_REST\_01608] Match API** [ An implementation shall provide a type `ara::rest::Match` that satisfies at least the interface and basic semantics as defined in the API specification below. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01609] Match Creation** [ When matching URIs against patterns, then for every path segment matched against either single wildcard "\*" or double wildcard "\*\*" an instance of `ara::rest::Match` shall be created. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01610] Match Access As String** [ `ara::rest::Match::Get` shall return a view of the matched URI segment. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01611] Match Access As Type** [ `ara::rest::Match::GetAs` shall convert the match into type T specified by the template parameter. Type T shall be `InputStreamable`: there shall exist a global function `std::istream& operator>>(std::istream&, const T&)` that performs lexical conversion. If conversion fails, `GetAs()` shall throw `std::invalid_argument`. Function overload `GetAs(T&&)` shall perform the same conversion but instead of throwing it returns the function argument if conversion fails. ]([RS\\_CM\\_00300](#))

## 7.8.3 Matches

**[SWS\_REST\_01612] Matches API** [ An implementation shall provide a type `ara::rest::Matches` that satisfies at least the interface and basic semantics as defined in the API specification below. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01613] Matches Creation** [ When matching URIs against patterns, then a `ara::rest::Matches` instance shall be created that shall contain all `ara::rest::Match` objects created. Matches shall own the Match instances and release them upon destruction. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01614] Matches of General Path Segments** [ Matches shall not represent non-wildcard path segments. ]([RS\\_CM\\_00300](#))

## 7.8.4 Route

**[SWS\_REST\_01615] Route API** [ An implementation shall provide a type `ara::rest::Route` that satisfies at least the interface and basic semantics as defined in the API specification below. ]([RS\\_CM\\_00300](#))

**[SWS\_REST\_01616] Route Semantics** [ `ara::rest::Route` shall call the user-defined function of type `ara::rest::Route::RouteHandlerType` provided as a constructor argument if `ara::rest::RequestMethod` and `ara::rest::Pattern` provided as constructor arguments match the request method and URI of the `ara::rest::ServerRequest` passed to `ara::rest::Route::operator()`. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00310](#))

**[SWS\_REST\_01617] Route Match** [ If a route matches, the `ara::rest::Route::RouteHandlerType` specified via its constructor shall be called along with the respective request and reply objects, and a set of `ara::rest::Matches` that represent wildcard matches of the request URI. ] (*RS\_CM\_00300*, *RS\_CM\_00310*)

**[SWS\_REST\_01618] Route Return Values** [ `ara::rest::Route::operator()` returns values of type `ara::rest::Route::Upshot`. The respective return values shall have the following effect on the routing described later:

- `ara::rest::Route::Upshot::kAccept`: `ara::rest::Router` shall not search for further matches.
- `ara::rest::Route::Upshot::kYield`: `ara::rest::Router` shall select the next matching route (multiple routes may match) or the default handler function in case of no next matching route.
- `ara::rest::Route::Upshot::kDefault`: `ara::rest::Router` shall execute its default handler function (specified below).

] (*RS\_CM\_00300*, *RS\_CM\_00310*)

**[SWS\_REST\_01619] Route Comparability** [ `ara::rest::Route` shall be equal-to, unequal-to and less-than comparable. ] (*RS\_CM\_00300*)

**[SWS\_REST\_01620] Route Order Criterion** [ Routes shall compare less-than in lexicographic order such that the given `ara::rest::Uri::Path` compare first, the given `ara::rest::RequestMethod` compare last. ] (*RS\_CM\_00300*)

**[SWS\_REST\_01621] Route Order Criterion for RequestMethod** [ While `ara::rest::Uri` is ordered lexicographically, `ara::rest::RequestMethod` shall be less-than comparable for route matching according to the following rule: The order of request methods is lexicographic with each enumerator representing a character of a string concatenated by operator `|`. Therefore, for single "digits" it holds that  $kGET < kPOST < \dots$  etc according to their underlying numeric values. For multiple "digits" it holds that - for example -  $kGET < kGET | kPOST < kGET | kPUT < \dots < kAny$ . But note that  $kGET | kPUT == kPUT | kGET$ . In words, the most precise specifiers (singleton request methods) have precedence over the less precise specification of sets of enumerators. This is not the same as simply taking the underlying numeric value of the OR-combined enumerators. ] (*RS\_CM\_00300*)

## 7.8.5 Router

**[SWS\_REST\_01622] Router API** [ An implementation shall provide a type `ara::rest::Router` that satisfies at least the interface and basic semantics as defined in the API specification below. ] (*RS\_CM\_00300*)

**[SWS\_REST\_01623] Router Semantics** [ `ara::rest::Router` shall maintain an ordered set of routes and it shall find a matching route by comparing the request

method and URI components of a given `ara::rest::ServerRequest` against each given `ara::rest::Route` in the set. [\]\(RS\\_CM\\_00300\)](#)

**[SWS\_REST\_01624] Router Usage** [ A router is a pre-defined request handler type which may be passed to the constructor of `ara::rest::Server` in order to de-multiplex messaging. [\]\(RS\\_CM\\_00300, RS\\_CM\\_00310\)](#)

**[SWS\_REST\_01625] Router Route Order** [ Routes shall be matched according to their order criterion as defined above from “smallest” to “greatest”. Incomparable routes shall be matched in the order of insertion into a router. [\]\(RS\\_CM\\_00300\)](#)

**[SWS\_REST\_01626] Router Route Match** [ If a route matches, its `ara::rest::Route::operator()` shall be called. [\]\(RS\\_CM\\_00300\)](#)

**[SWS\_REST\_01627] Router Route Skipping** [ If a `ara::rest::Route::operator()` returns an `ara::rest::Route::Upshot` value of “kYield”, then a router shall call the next match. [\]\(RS\\_CM\\_00300\)](#)

**[SWS\_REST\_01628] Router Default Route** [ If no route matches, the request handler function specified via `ara::rest::Router::SetDefaultHandler` shall be called, if it has been set. If no such handler is set, the router shall silently ignore the request currently under inspection. [\]\(RS\\_CM\\_00300\)](#)

**[SWS\_REST\_01629] Router Route Defaulting** [ If a `ara::rest::Route::operator()` returns an `ara::rest::Route::Upshot` value of “Default”, then the request handler function specified via `ara::rest::Router::SetDefaultHandler` shall be called, if it has been set. If no such handler is set, the router shall silently ignore the request currently under inspection. [\]\(RS\\_CM\\_00300\)](#)

## 7.9 Object Graph Model

`ara::rest` message payloads are always represented as object graph data-structures (object graph models; OGM). OGM serve as a universal exchange format.

**[SWS\_REST\_01701] OGM Representation** [ All transport-specific message payloads shall be converted to and from OGM for communication and interaction with a service application. [\]\(RS\\_CM\\_00300\)](#)

**[SWS\_REST\_01702] OGM Syntax** [ An implementation shall provide a basic set of data types whose interfaces shall at least satisfy interfaces and basic functionalities according to their API specifications provided by `ara::rest::ogm::Node`, `ara::rest::ogm::Field`, `ara::rest::ogm::Value`, `ara::rest::ogm::String`, `ara::rest::ogm::Int`, `ara::rest::ogm::Real`, `ara::rest::ogm::Array` and `ara::rest::ogm::Object`. [\]\(RS\\_CM\\_00300, RS\\_CM\\_00305\)](#)

**[SWS\_REST\_01703] OGM Semantics: Internal Ownership** [ OGM instances are always trees. Each parent node shall uniquely own its children. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_01704] OGM Semantics: External Ownership** [ Ownership of an OGM shall never be shared within an application. The lifetime of an OGM shall strictly be bound to the lifetime of the `Pointer` instance owning an OGM after construction. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_01705] OGM Construction Semantics** [ OGM node constructors are non-public. To construct OGM nodes their respective static `Make()` member functions shall be used which return a `Pointer` instance that own the OGM just created. Only leaf-types in the type hierarchy may be instantiated. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_01706] OGM Destruction Semantics** [ If an owner (a parent node or a `Pointer` holding a reference to an OGM) is destroyed, all owned objects shall be destroyed too. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_01707] OGM Copy Semantics** [ An OGM cannot be copied directly or implicitly. To copy an OGM `ara::rest::ogm::Copy` shall be used. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_01708] OGM Move Semantics For Owners** [ To move an OGM, its owning `Pointer` instance shall be moved. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_01709] OGM Move Semantics For Non-owners: Release** [ To obtain ownership of subtrees, member functions with prefix “Release” shall be called on the owning OGM node. Only node types that represent sets (`ara::rest::ogm::Array` and `ara::rest::ogm::Object`) may be empty and therefore have a “Release” functionality. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_01710] OGM Move Semantics For Non-owners: Replace** [ For consistency, some node types have “Replace” functions instead of “Release”. To take on ownership of a sub-tree, it shall be replaced by a suitable replacement object. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_01711] OGM Iterator Semantics** [ Some OGM node types are iterable. Iterators shall not expose any internal data management. The respective iterator types shall provide C++ references directly to the referenced set elements ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_01712] OGM String encoding** [ `ara::rest::ogm::String` shall support UTF-8. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_01713] OGM Int precision** [ `ara::rest::ogm::Int` is signed and shall be at least as precise as a C++ `std::int64_t` integer type. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_01714] OGM Real precision** [ `ara::rest::ogm::Real` shall be at least as precise as a C++ `double` floating point type. ]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_02423] OGM Thread safety** [ Setting and getting values of OGM nodes shall be thread safe. ] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_01715] OGM Visit** [ [ara::rest::ogm::Visit](#) implements the visit pattern and shall expose the actual type of an OGM node from a reference to any of its parents in the OGM node type hierarchy. ] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_02415] OGM VisitAll** [ [ara::rest::ogm::VisitAll](#) implements the recursive visit pattern and shall expose the actual types of OGM nodes from a reference traversing all values of the given node. ] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_02416] OGM Utility** [ [ara::rest::ogm::Get](#), [ara::rest::ogm::GetValue](#), [ara::rest::ogm::Set](#) and [ara::rest::ogm::SetValue](#) implement OGM helper functions for getting and setting primitive values in OGM nodes. ] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

**[SWS\_REST\_02417] OGM Cast** [ [ara::rest::ogm::Cast](#) implements functionality for casting [ara::rest::ogm::Node](#) objects to concrete OGM node types. ] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#))

## 7.10 Network binding

The following chapters describe the requirements according to specific bus protocol bindings and the serialization. In the current version, only HTTP/1.1 [8] with JSON [9] payload is supported.

### 7.10.1 Transport protocol

**[SWS\_REST\_01801] Transport protocol** [ An implementation shall implement HTTP/1.1 [8] to transport the payload over the network. ] ([RS\\_CM\\_00312](#))

**[SWS\_REST\_01802] Mapping of [ara::rest::RequestMethod](#) kGet** [ [ara::rest::Requests](#) with the [ara::rest::RequestMethod](#) kGet shall be transported over the HTTP/1.1 GET method. ] ([RS\\_CM\\_00312](#))

**[SWS\_REST\_01803] Mapping of [ara::rest::RequestMethod](#) kPost** [ [ara::rest::Requests](#) with the [ara::rest::RequestMethod](#) kPost shall be transported over the HTTP/1.1 POST method. ] ([RS\\_CM\\_00312](#))

**[SWS\_REST\_01804] Mapping of [ara::rest::RequestMethod](#) kPut** [ [ara::rest::Requests](#) with the [ara::rest::RequestMethod](#) kPut shall be transported over the HTTP/1.1 PUT method. ] ([RS\\_CM\\_00312](#))

**[SWS\_REST\_01805] Mapping of [ara::rest::RequestMethod](#) kHead** [ [ara::rest::Requests](#) with the [ara::rest::RequestMethod](#) kHead shall be transported over the HTTP/1.1 HEAD method. ] ([RS\\_CM\\_00312](#))



**[SWS\_REST\_01806] Mapping of `ara::rest::RequestMethod` `kDelete`** [ `ara::rest::Requests` with the `ara::rest::RequestMethod` `kDelete` shall be transported over the HTTP/1.1 DELETE method. ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01807] Mapping of `ara::rest::RequestMethod` `kOptions`** [ `ara::rest::Requests` with the `ara::rest::RequestMethod` `kOptions` shall be transported over the HTTP/1.1 OPTIONS method. ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01808] HTTP status code handling** [ An implementation shall follow the HTTP/1.1 status code specification [8]. ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01810] Websocket handling** [ Websocket channel shall be opened during the first `ara::rest::Event` subscription. Websocket channel shall be kept open until all `ara::rest::Events` and `ara::rest::ServerEvents` on the channel have been canceled or invalidated. There shall be one Websocket channel between `ara::rest::Client` and `ara::rest::Server` for all events. In case the channel is closed without canceling all the events they shall be immediately invalidated by changing the event subscription states to `ara::rest::SubscriptionState::kInvalid`. ]([RS\\_CM\\_00314](#))

**[SWS\_REST\_01811] Event subscription message** [ `ara::rest::Event` subscription message shall be sent as JSON over Websocket channel. Subscription message shall always be sent by the `ara::rest::Client::Subscribe` API call. After successful response to the subscription message on the Client side, the event subscription state shall be changed to `ara::rest::SubscriptionState::kSubscribed`. Subscription message shall be in the following format:

```

1 {
2     "type": "subscribe",      // Message type as string
3     "event": <entity_uri>,   // Entity URI to subscribe as string
4     "interval": <interval>, // Interval of periodic events in milliseconds
                           as integer
5     "updateLimit": <limit>  // Limit of updates for triggered events in
                           milliseconds as integer
6 }
7

```

]([RS\\_CM\\_00314](#))

**[SWS\_REST\_01812] Event cancellation message** [ Event cancellation message shall be sent as JSON over Websocket channel when either `ara::rest::Event::Unsubscribe` or `ara::rest::ServerEvent::SetSubscriptionState` with parameter `ara::rest::SubscriptionState::kCanceled` is called. No other payload is allowed on the channel before or after the response to the cancellation message. After successful response to the cancellation message on the Client side, the event subscription state shall be changed to `ara::rest::SubscriptionState::kCanceled`. Cancellation message shall be in the following format:

```

1 {
2     "type": "unsubscribe", // Message type as string
3     "event": <entity_uri> // Entity URI to subscribe as string

```

4 }  
5

](RS\_CM\_00314)

**[SWS\_REST\_01813] Event state responses** [ Subscription and cancellation messages shall be responded as JSON over the Websocket channel. The type field shall be either "subscribe", "unsubscribe" or "resubscribe" depending on the responded message set with `ara::rest::ServerEvent::SetSubscriptionState`. The response shall be sent immediately after call to `ara::rest::ServerEvent::SetSubscriptionState`. The response shall be in the following format:

```
1 {
2     "type": <type>,           // Message type as string
3     "event": <entity_uri>,    // Entity URI as string
4     "status": "ok"           // Response status as string
5 }
6
```

](RS\_CM\_00314)

**[SWS\_REST\_01814] Event error message** [ Event error messages shall be transmitted as JSON over the Websocket channel. Errors shall lead to immediate invalidation of the `ara::rest::Event` and corresponding `ara::rest::ServerEvent`. Therefore the `ara::rest::Server` application shall call `ara::rest::ServerEvent::SetSubscriptionState` with parameter `ara::rest::SubscriptionState::kInvalid`. The event error message is issued by `ara::rest::ServerEvent::SendError` where the parameter `errorCode` matches to the JSON key "code" and `errorMessage` to "data". The error shall be sent immediately after call to the `ara::rest::ServerEvent::SendError`. The message shall be in the following format:

```
1 {
2     "type": "error",          // Message type as string
3     "code": <error_code>,    // error code as integer
4     "event": <entity_uri>,    // Entity URI as string
5     "data": <error_msg>      // error message as string
6 }
7
```

](RS\_CM\_00314)

**[SWS\_REST\_01815] Event data response** [ Event notifications sent by `ara::rest::Server` shall be transported as JSON over the Websocket channel. Event notifications shall only be sent if the subscription state of the `ara::rest::ServerEvent` is `ara::rest::SubscriptionState::kSubscribed`. The response shall be in the following format:

```
1 {
2     "type": "data",          // Message type as string
3     "event": <entity_uri>,    // Entity URI as string
```

```
4     "data": <payload>           // Payload of event, represented on ara::rest
      level by ara::rest::ogm
5 }
6
```

]([RS\\_CM\\_00314](#))

**[SWS\_REST\_01816] Compression support** [ An implementation shall provide data compression support to improve transfer speed and network bandwidth utilization. ] ([RS\\_CM\\_00312](#))

**[SWS\_REST\_01817] Compression support DEFLATE** [ An implementation shall provide data compression support with the DEFLATE algorithm according to [10] and as listed in [HttpAcceptEncodingEnum](#). ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01818] Compression support GZIP** [ An implementation shall provide data compression support with the GZIP algorithm according to [11] and as listed in [HttpAcceptEncodingEnum](#). ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01819] Compression support** [ Data compression shall be hidden from the application context. ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01820] Compression support** [ The accepted data compression of a [ara::rest::Client](#) shall be as configured in [acceptsEncoding](#) of the [RestHttpPortPrototypeMapping](#). ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01833] Compression of small payloads** [ An implementation may skip compression of payloads which is smaller than 1400 bytes to optimize CPU usage. ] ([RS\\_CM\\_00312](#))

**[SWS\_REST\_01834] Compression of small payloads** [ If there is a multi selection of [HttpAcceptEncodingEnum](#), the GZIP algorithm shall be preferred from DEFLATE by the [ara::rest::Server](#) when responding to requests. ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01821] Default host for [ara::rest::Client](#)** [ Host address given by [host](#) shall be used if no information is given by [ara::rest::Uri::GetHost](#) for a [ara::rest::Request](#). ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01822] Default TCP port for [ara::rest::Client](#)** [ TCP port given by [tcpPort](#) shall be used if no information is given by [ara::rest::Uri::GetPort](#) for a [ara::rest::Request](#). ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01823] IP address configuration for [ara::rest::Server](#)** [ [ara::rest::Server](#) shall bind to the IP address given by the [Ipv4Configuration/Ipv6Configuration](#) attribute of the [NetworkEndpoint](#) that is referenced (in role [networkEndpointAddress](#)) by [host](#). ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01824] TCP port configuration for [ara::rest::Server](#)** [ The [ara::rest::Server](#) shall bind to the [tcpPort](#) of the [RestHttpPortPrototypeMapping](#). ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01825] Accept content type HTTP field for binary data requests** [ `ara::rest::Client` and `ara::rest::Server` applications shall provide the requested MIME-Type in the HTTP accept header for binary data. ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01826] `ara::rest::Client` set accept content type of HTTP/1.1 GET requests for binary data** [ HTTP accept header shall be set with the function `ara::rest::RequestHeader::SetField` before sending the HTTP/1.1 GET request. ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01827] `ara::rest::Client` get binary HTTP response data** [ The function `ara::rest::Reply::ReleaseBinary` shall be used to get the binary data of a `ara::rest::Reply` transmitted with HTTP. ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01828] `ara::rest::Server` set binary HTTP/1.1 GET response payload** [ The function `ara::rest::ServerReply::Send2` shall be used to provide the binary data content of the HTTP response. ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01829] `ara::rest::Server` set HTTP content type of response containing binary data** [ HTTP content type of the response shall be set with the function `ara::rest::ReplyHeader::SetField` before sending the response. ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01830] `ara::rest::Client` set content type type of HTTP/1.1 PUT and POST requests for sending binary data** [ HTTP content type header shall be set with the function `ara::rest::RequestHeader::SetField` before sending the HTTP/1.1 PUT or POST request. ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01831] `ara::rest::Client` set binary payload for HTTP/1.1 PUT and POST requests** [ The function `ara::rest::Request::Request7` shall be used to provide the binary data content of the HTTP/1.1 PUT or POST request. ]([RS\\_CM\\_00312](#))

**[SWS\_REST\_01832] `ara::rest::Server` get binary payload for HTTP/1.1 PUT and POST requests** [ The function `ara::rest::ServerRequest::ReleaseBinary` shall be used to get the binary data content of the HTTP/1.1 PUT or POST request. ]([RS\\_CM\\_00312](#))

## 7.10.2 Serialization of payload

On application level the `ara::rest` message payload is (with the exception of binary data) represented as object graph data-structures. This data representation needs to be serialized first before it can be transmitted. The necessary mapping of the object graph data-structures to the `ara::rest` message payload is specified in this chapter.

**[SWS\_REST\_01851] Serialization format** [ An implementation shall serialize the `ara::rest` message payload with JSON [9]. ]([RS\\_CM\\_00313](#))

**[SWS\_REST\_01852] Default HTTP content type** [ An implementation shall set the HTTP content type header field to the MIME "application/json" as default for

`ara::rest::RequestMethod` `kPost` and `kPut` requests. Note that the application can still override the content type with `ara::rest::ReplyHeader::SetField`. ]  
([RS\\_CM\\_00313](#), [RS\\_CM\\_00312](#))

**[SWS\_REST\_01859] Default HTTP accept type** [ An implementation shall set the HTTP accept header field to the MIME "application/json" as default for `ara::rest::RequestMethod` `kGet` request. Note that the application can still override the accept header field with `ara::rest::RequestHeader::SetField`. ]([RS\\_CM\\_00313](#), [RS\\_CM\\_00312](#))

**[SWS\_REST\_01853] Serialization of `ara::rest::ogm::Object`** [ The `ara::rest::ogm::Object` shall be serialized as a JSON root object. ]  
([RS\\_CM\\_00313](#))

**[SWS\_REST\_01854] Serialization of `ara::rest::ogm::Field`** [ The `ara::rest::ogm::Field` shall be serialized as a JSON object. Note that an `ara::rest::ogm::Field` can contain further `ara::rest::ogm::Fields` and `ara::rest::ogm::Arrays` which have to be serialized accordingly. ]  
([RS\\_CM\\_00313](#))

**[SWS\_REST\_01855] Serialization of `ara::rest::ogm::Array`** [ The `ara::rest::ogm::Array` shall be serialized as a JSON array. ]([RS\\_CM\\_00313](#))

**[SWS\_REST\_01856] Serialization of `ara::rest::ogm::Int`** [ The `ara::rest::ogm::Int` shall be serialized as a JSON integer value. ]  
([RS\\_CM\\_00313](#))

**[SWS\_REST\_01857] Serialization of `ara::rest::ogm::Real`** [ The `ara::rest::ogm::Real` shall be serialized as a JSON floating point values. ]([RS\\_CM\\_00313](#))

**[SWS\_REST\_01858] Serialization of `ara::rest::ogm::String`** [ The `ara::rest::ogm::String` shall be serialized as a JSON string. ]([RS\\_CM\\_00313](#))

**[SWS\_REST\_01899] Serialization of other MIME-types** [ Serialization of other MIME-types then "application/json" (e.g. binary data) is implementation-defined. ]  
([RS\\_CM\\_00313](#))

## 8 API specification

This chapter contains the formal API documentation of `ara::rest`.

### 8.1 `ara::rest::Allocator`

[SWS\_REST\_02000] [`ara::rest::Allocator` class shall be declared in the `ara/rest/allocator.h` header file:

```
1         class ara::rest::Allocator;
```

]([RS\\_CM\\_00300](#))

#### 8.1.1 Allocator

<b>Service name:</b>	<code>ara::rest::Allocator::Allocator</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>ara::rest::Allocator::Allocator()=default</code>
<b>Function param:</b>	None
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/allocator.h</code>
<b>Class:</b>	<code>ara::rest::Allocator</code>
<b>Description:</b>	Constructs this object.

**Table 8.1: `ara::rest::Allocator::Allocator`**

[SWS\_REST\_02001] `ara::rest::Allocator::Allocator` [[Table 8.1](#) describes the interface `ara::rest::Allocator::Allocator`.]([RS\\_CM\\_00300](#))

#### 8.1.2 `~Allocator`

<b>Service name:</b>	<code>ara::rest::Allocator::~~Allocator</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>virtual ara::rest::Allocator::~~Allocator()</code>
<b>Function param:</b>	None
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/allocator.h</code>
<b>Class:</b>	<code>ara::rest::Allocator</code>
<b>Description:</b>	Destroys this object.

**Table 8.2: `ara::rest::Allocator::~~Allocator`**

[SWS\_REST\_02002] `ara::rest::Allocator::~~Allocator` [Table 8.2 describes the interface `ara::rest::Allocator::~~Allocator`.] ([RS\\_CM\\_00300](#))

### 8.1.3 allocate

<b>Service name:</b>	<code>ara::rest::Allocator::allocate</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>void* ara::rest::Allocator::allocate(std::size_t bytes, std::size_t alignment=alignof(std::max_align_t))</code>	
<b>Function param:</b>	<code>bytes</code>	desired size of the memory area to be allocated
<b>Function param:</b>	<code>alignment</code>	alignment of the memory area
<b>Return value:</b>	a pointer to the allocated memory area	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	<code>ara/rest/allocator.h</code>	
<b>Class:</b>	<code>ara::rest::Allocator</code>	
<b>Description:</b>	Allocates a memory area.	

**Table 8.3: `ara::rest::Allocator::allocate`**

[SWS\_REST\_02003] `ara::rest::Allocator::allocate` [Table 8.3 describes the interface `ara::rest::Allocator::allocate`.] ([RS\\_CM\\_00300](#))

### 8.1.4 deallocate

<b>Service name:</b>	<code>ara::rest::Allocator::deallocate</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>void ara::rest::Allocator::deallocate(void *p, std::size_t bytes, std::size_t alignment=alignof(std::max_align_t))</code>	
<b>Function param:</b>	<code>p</code>	pointer to the allocated memory area
<b>Function param:</b>	<code>bytes</code>	size of the allocated memory area
<b>Function param:</b>	<code>alignment</code>	alignment of allocated memory area
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	<code>ara/rest/allocator.h</code>	
<b>Class:</b>	<code>ara::rest::Allocator</code>	
<b>Description:</b>	Releases a memory area.	

**Table 8.4: `ara::rest::Allocator::deallocate`**

[SWS\_REST\_02004] `ara::rest::Allocator::deallocate` [Table 8.4 describes the interface `ara::rest::Allocator::deallocate`.] ([RS\\_CM\\_00300](#))

### 8.1.5 is\_equal

<b>Service name:</b>	ara::rest::Allocator::is_equal	
<b>Type:</b>	Member function	
<b>Syntax:</b>	bool ara::rest::Allocator::is_equal(const Allocator &alloc) const	
<b>Function param:</b>	alloc	an allocator to compare against
<b>Return value:</b>	true if the two allocators compare equal	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/allocator.h	
<b>Class:</b>	ara::rest::Allocator	
<b>Description:</b>	Tests whether two allocators are equal. Allocators are equal if memory allocated by one can be deallocated by the other.	

**Table 8.5: ara::rest::Allocator::is\_equal**

[SWS\_REST\_02005] `ara::rest::Allocator::is_equal` [Table 8.5 describes the interface `ara::rest::Allocator::is_equal`.] (RS\_CM\_00300)

## 8.2 ara::rest::Client

[SWS\_REST\_02006] [ara::rest::Client class shall be declared in the ara/rest/client.h header file:

```
1     class ara::rest::Client;
```

] (RS\_CM\_00300, RS\_CM\_00301)

### 8.2.1 NotificationHandlerType

<b>Name:</b>	NotificationHandlerType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::Client::NotificationHandlerType = void(const ogm::Object&)
<b>Header file:</b>	ara/rest/client.h
<b>Class:</b>	ara::rest::Client
<b>Description:</b>	Denotes a callback function for notifications.

**Table 8.6: ara::rest::Client::NotificationHandlerType**

[SWS\_REST\_02007] `NotificationHandlerType` [Table 8.6 describes the type alias `ara::rest::Client::NotificationHandlerType`.] (RS\_CM\_00300, RS\_CM\_00301)

### 8.2.2 SubscriptionStateHandlerType

<b>Name:</b>	SubscriptionStateHandlerType
--------------	------------------------------



<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::Client::SubscriptionStateHandlerType = void(const Event&, SubscriptionState)
<b>Header file:</b>	ara/rest/client.h
<b>Class:</b>	ara::rest::Client
<b>Description:</b>	Denotes a callback to call if subscription status changes.

**Table 8.7: ara::rest::Client::SubscriptionStateHandlerType**

[SWS\_REST\_02008] **SubscriptionStateHandlerType** [Table 8.7 describes the type alias [ara::rest::Client::SubscriptionStateHandlerType](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

### 8.2.3 Client

<b>Service name:</b>	ara::rest::Client::Client	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ara::rest::Client::Client(const ara::rest::InstanceIdentifier &inst_id, Allocator *alloc=GetDefaultAllocator())	
<b>Function param:</b>	inst_id	<a href="#">ara::rest::InstanceIdentifier</a> identifies concrete service instace
<b>Function param:</b>	alloc	allocator for dynamic memory
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/client.h	
<b>Class:</b>	ara::rest::Client	
<b>Description:</b>	Constructs a client.	

**Table 8.8: ara::rest::Client::Client**

[SWS\_REST\_02009] **ara::rest::Client::Client** [Table 8.8 describes the interface [ara::rest::Client::Client](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

### 8.2.4 Client

<b>Service name:</b>	ara::rest::Client::Client
<b>Type:</b>	Member function
<b>Syntax:</b>	ara::rest::Client::Client(const Client &)=delete
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/client.h
<b>Class:</b>	ara::rest::Client
<b>Description:</b>	non-copy-constructible

**Table 8.9: ara::rest::Client::Client**

[SWS\_REST\_02010] `ara::rest::Client::Client` [Table 8.9 describes the interface `ara::rest::Client::Client`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

### 8.2.5 operator=

<b>Service name:</b>	<code>ara::rest::Client::operator=</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Client&amp; ara::rest::Client::operator=(const Client &amp;)=delete</code>
<b>Return value:</b>	a value of type <code>Client &amp;</code>
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/client.h</code>
<b>Class:</b>	<code>ara::rest::Client</code>
<b>Description:</b>	non-copy-assignable

**Table 8.10: `ara::rest::Client::operator=`**

[SWS\_REST\_02011] `ara::rest::Client::operator=` [Table 8.10 describes the interface `ara::rest::Client::operator=`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

### 8.2.6 Stop

<b>Service name:</b>	<code>ara::rest::Client::Stop</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Task&lt;void&gt; ara::rest::Client::Stop(ShutdownPolicy policy=ShutdownPolicy::kGraceful)</code>
<b>Function param:</b>	policy   shutdown policy
<b>Return value:</b>	a task waiting for shutdown to complete
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/client.h</code>
<b>Class:</b>	<code>ara::rest::Client</code>
<b>Description:</b>	Requests a client shutdown. If shutting down gracefully, the client waits for all transactions to finish. If not, then all connections must be terminated instantly.

**Table 8.11: `ara::rest::Client::Stop`**

[SWS\_REST\_02012] `ara::rest::Client::Stop` [Table 8.11 describes the interface `ara::rest::Client::Stop`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

### 8.2.7 Send

<b>Service name:</b>	<code>ara::rest::Client::Send</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Task&lt;Pointer&lt;Reply&gt; &gt; ara::rest::Client::Send(const Request &amp;req)</code>

<b>Function param:</b>	req	a request message
<b>Return value:</b>	a task waiting for the corresponding reply	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/client.h	
<b>Class:</b>	ara::rest::Client	
<b>Description:</b>	Issues a request to a peer. Issues a request to the peer either specified in the client configuration record or the URI of the request. The configuration record is identified by the id specified in the Client constructor. If Uri::Authority is set, it overwrites the configuration record.	

**Table 8.12: ara::rest::Client::Send**

[SWS\_REST\_02013] **ara::rest::Client::Send** [Table 8.12 describes the interface `ara::rest::Client::Send`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

## 8.2.8 Subscribe

<b>Service name:</b>	ara::rest::Client::Subscribe	
<b>Type:</b>	Member function	
<b>Syntax:</b>	Task<Event> ara::rest::Client::Subscribe(const Uri &uri, EventPolicy policy, duration_t time, const Function< NotificationHandlerType > &notify, const Function< SubscriptionStateHandlerType > &state={})	
<b>Function param:</b>	uri	the event to subscribe to
<b>Function param:</b>	policy	the notification policy
<b>Function param:</b>	time	time bound as a parameter of the notification policy
<b>Function param:</b>	notify	user-defined event notification handler function
<b>Function param:</b>	state	user-define subscription state observer function
<b>Return value:</b>	a task waiting for the Event construction and subscription Reply.	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/client.h	
<b>Class:</b>	ara::rest::Client	
<b>Description:</b>	Performs an event subscription. An event is uniquely identified by its Uri. A subscription to an event means that if preconditions are met a notification is issued whose message payload is identical to the result set obtained by issuing a GET request on the Uri.	

**Table 8.13: ara::rest::Client::Subscribe**

[SWS\_REST\_02014] **ara::rest::Client::Subscribe** [Table 8.13 describes the interface `ara::rest::Client::Subscribe`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

## 8.2.9 GetError

<b>Service name:</b>	ara::rest::Client::GetError	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ara::core::ErrorCode ara::rest::Client::GetError() const	

<b>Function param:</b>	None
<b>Return value:</b>	status of the client
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/client.h
<b>Class:</b>	ara::rest::Client
<b>Description:</b>	Obtain client status.

**Table 8.14: ara::rest::Client::GetError**

[SWS\_REST\_02015] **ara::rest::Client::GetError** [Table 8.14 describes the interface `ara::rest::Client::GetError`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

### 8.2.10 ObserveError

<b>Service name:</b>	ara::rest::Client::ObserveError	
<b>Type:</b>	Member function	
<b>Syntax:</b>	void ara::rest::Client::ObserveError(const Function< void(ara::core::ErrorCode)> &hnd)	
<b>Function param:</b>	hnd	user-defined handler function to called on status changes
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/client.h	
<b>Class:</b>	ara::rest::Client	
<b>Description:</b>	Observe status changes.	

**Table 8.15: ara::rest::Client::ObserveError**

[SWS\_REST\_02016] **ara::rest::Client::ObserveError** [Table 8.15 describes the interface `ara::rest::Client::ObserveError`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

## 8.3 ara::rest::Event

[SWS\_REST\_02017] [ara::rest::Event class shall be declared in the `ara/rest/client.h` header file:

```
1     class ara::rest::Event;
```

] ([RS\\_CM\\_00300](#))

### 8.3.1 Event

<b>Service name:</b>	ara::rest::Event::Event
<b>Type:</b>	Member function
<b>Syntax:</b>	ara::rest::Event::Event(const Event &)=delete

<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/client.h
<b>Class:</b>	ara::rest::Event
<b>Description:</b>	Non-copyable.

**Table 8.16: ara::rest::Event::Event**

[SWS\_REST\_02018] **ara::rest::Event::Event** [Table 8.16 describes the interface `ara::rest::Event::Event`.] ([RS\\_CM\\_00300](#))

### 8.3.2 operator=

<b>Service name:</b>	ara::rest::Event::operator=
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Event&amp; ara::rest::Event::operator=(const Event &amp;)=delete</code>
<b>Return value:</b>	a value of type <code>Event &amp;</code>
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/client.h
<b>Class:</b>	ara::rest::Event
<b>Description:</b>	Non-copy-assignable.

**Table 8.17: ara::rest::Event::operator=**

[SWS\_REST\_02019] **ara::rest::Event::operator=** [Table 8.17 describes the interface `ara::rest::Event::operator=`.] ([RS\\_CM\\_00300](#))

### 8.3.3 Unsubscribe

<b>Service name:</b>	ara::rest::Event::Unsubscribe
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Task&lt;bool&gt; ara::rest::Event::Unsubscribe()</code>
<b>Function param:</b>	None
<b>Return value:</b>	a task waiting for cancellation which returns true on success.
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/client.h
<b>Class:</b>	ara::rest::Event
<b>Description:</b>	Cancels an event subscription by issuing a cancelation request. A subscription can also be terminated (but not canceled) by destroying the correspond Event object.

**Table 8.18: ara::rest::Event::Unsubscribe**

[SWS\_REST\_02020] **ara::rest::Event::Unsubscribe** [Table 8.18 describes the interface `ara::rest::Event::Unsubscribe`.] ([RS\\_CM\\_00300](#))

### 8.3.4 Resubscribe

<b>Service name:</b>	ara::rest::Event::Resubscribe
<b>Type:</b>	Member function
<b>Syntax:</b>	Task<bool> ara::rest::Event::Resubscribe()
<b>Function param:</b>	None
<b>Return value:</b>	a task waiting for re-subscription to be finished which returns true on success
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/client.h
<b>Class:</b>	ara::rest::Event
<b>Description:</b>	Re-subscribes to an event. Resubscription to an already subscribed event is valid but has not user-visible effect.

**Table 8.19: ara::rest::Event::Resubscribe**

[SWS\_REST\_02021] **ara::rest::Event::Resubscribe** [Table 8.19 describes the interface `ara::rest::Event::Resubscribe`.] ([RS\\_CM\\_00300](#))

### 8.3.5 GetUri

<b>Service name:</b>	ara::rest::Event::GetUri
<b>Type:</b>	Member function
<b>Syntax:</b>	const Uri& ara::rest::Event::GetUri() const
<b>Function param:</b>	None
<b>Return value:</b>	the Uri corresponding to this event subscription
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/client.h
<b>Class:</b>	ara::rest::Event
<b>Description:</b>	Returns the event Uri.

**Table 8.20: ara::rest::Event::GetUri**

[SWS\_REST\_02022] **ara::rest::Event::GetUri** [Table 8.20 describes the interface `ara::rest::Event::GetUri`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.3.6 GetSubscriptionState

<b>Service name:</b>	ara::rest::Event::GetSubscriptionState
<b>Type:</b>	Member function
<b>Syntax:</b>	SubscriptionState ara::rest::Event::GetSubscriptionState() const
<b>Function param:</b>	None
<b>Return value:</b>	the current subscription state as perceived by the client
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/client.h
<b>Class:</b>	ara::rest::Event
<b>Description:</b>	Returns the current subscription state.

**Table 8.21: ara::rest::Event::GetSubscriptionState**

[SWS\_REST\_02023] **ara::rest::Event::GetSubscriptionState** [Table 8.21 describes the interface `ara::rest::Event::GetSubscriptionState`.] (*RS\_CM\_00300*)

### 8.3.7 operator==

<b>Service name:</b>	ara::rest::Event::operator==	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	friend bool operator==(const Event &a, const Event &b)	
<b>Function param:</b>	a	an event
<b>Function param:</b>	b	an event
<b>Return value:</b>	true if a and b are equal	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/client.h	
<b>Namespace:</b>	ara::rest::Event	
<b>Description:</b>	Tests events for equality.	

**Table 8.22: ara::rest::Event::operator==**

[SWS\_REST\_02024] **ara::rest::Event::operator==** [Table 8.22 describes the interface `ara::rest::Event::operator==`.] (*RS\_CM\_00300*)

### 8.3.8 operator!=

<b>Service name:</b>	ara::rest::Event::operator!=	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	friend bool operator!=(const Event &a, const Event &b)	
<b>Function param:</b>	a	an event
<b>Function param:</b>	b	an event
<b>Return value:</b>	true if a and b are unequal	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/client.h	
<b>Namespace:</b>	ara::rest::Event	
<b>Description:</b>	Tests events for inequality.	

**Table 8.23: ara::rest::Event::operator!=**

[SWS\_REST\_02025] **ara::rest::Event::operator!=** [Table 8.23 describes the interface `ara::rest::Event::operator!=`.] (*RS\_CM\_00300*)

### 8.3.9 operator<

<b>Service name:</b>	ara::rest::Event::operator<	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	friend bool operator<(const Event &a, const Event &b)	
<b>Function param:</b>	a	an event
<b>Function param:</b>	b	an event
<b>Return value:</b>	true if a less-than b	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/client.h	
<b>Namespace:</b>	ara::rest::Event	
<b>Description:</b>	Tests events for their partial order Order criterion is implementation-defined.	

**Table 8.24: ara::rest::Event::operator<**

[SWS\_REST\_02026] `ara::rest::Event::operator<` [Table 8.24 describes the interface `ara::rest::Event::operator<`.] (RS\_CM\_00300)

## 8.4 ara::rest::IteratorRange

[SWS\_REST\_02382] [ara::rest::IteratorRange class shall be declared in the `ara/rest/iterator.h` header file:

```
1     template <typename IterT >
2     class ara::rest::IteratorRange;
```

] (RS\_CM\_00300)

### 8.4.1 Iterator

<b>Name:</b>	Iterator
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::IteratorRange< IterT >::Iterator = IterT
<b>Header file:</b>	ara/rest/iterator.h
<b>Class:</b>	ara::rest::IteratorRange
<b>Description:</b>	Type of the underlying pair of iterators.

**Table 8.25: ara::rest::IteratorRange::Iterator**

[SWS\_REST\_02383] `Iterator` [Table 8.25 describes the type alias `ara::rest::IteratorRange::Iterator`.] (RS\_CM\_00300)

### 8.4.2 IteratorRange

<b>Service name:</b>	ara::rest::IteratorRange::IteratorRange
----------------------	---



<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>ara::rest::IteratorRange&lt; IterT                  &gt;::IteratorRange(Iterator first, Iterator last)</code>	
<b>Function param:</b>	first	an iterator denoting the start of the sequence
<b>Function param:</b>	last	an iterator denoting the end of the sequence
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/iterator.h	
<b>Class:</b>	ara::rest::IteratorRange	
<b>Description:</b>	Constructs an IteratorRange from a pair of iterators. For convenient construction, see MakeIteratorRange().	

**Table 8.26: ara::rest::IteratorRange::IteratorRange**

[SWS\_REST\_02384] `ara::rest::IteratorRange::IteratorRange` [Table 8.26 describes the interface `ara::rest::IteratorRange::IteratorRange`.] (RS\_CM\_00300)

### 8.4.3 begin

<b>Service name:</b>	<code>ara::rest::IteratorRange::begin</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>Iterator ara::rest::IteratorRange&lt; IterT                  &gt;::begin()                  const</code>	
<b>Function param:</b>	None	
<b>Return value:</b>	an iterator	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/iterator.h	
<b>Class:</b>	ara::rest::IteratorRange	
<b>Description:</b>	Returns the start of the sequence.	

**Table 8.27: ara::rest::IteratorRange::begin**

[SWS\_REST\_02385] `ara::rest::IteratorRange::begin` [Table 8.27 describes the interface `ara::rest::IteratorRange::begin`.] (RS\_CM\_00300)

### 8.4.4 end

<b>Service name:</b>	<code>ara::rest::IteratorRange::end</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>Iterator ara::rest::IteratorRange&lt; IterT                  &gt;::end()                  const</code>	
<b>Function param:</b>	None	
<b>Return value:</b>	an iterator	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/iterator.h	
<b>Class:</b>	ara::rest::IteratorRange	
<b>Description:</b>	Returns the end of the sequence.	

**Table 8.28: ara::rest::IteratorRange::end**

[SWS\_REST\_02386] `ara::rest::IteratorRange::end` [Table 8.28 describes the interface `ara::rest::IteratorRange::end.`] ([RS\\_CM\\_00300](#))

### 8.4.5 begin

<b>Service name:</b>	<code>ara::rest::IteratorRange::begin</code>
<b>Type:</b>	Non-member function
<b>Syntax:</b>	<code>friend Iterator begin(const IteratorRange &amp;r)</code>
<b>Function param:</b>	<code>r</code>   an <code>IteratorRange</code>
<b>Return value:</b>	the start of the sequence
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/iterator.h</code>
<b>Namespace:</b>	<code>ara::rest::IteratorRange</code>
<b>Description:</b>	Non-member equivalent of <code>IteratorRange::begin()</code>

**Table 8.29: `ara::rest::IteratorRange::begin`**

[SWS\_REST\_02387] `ara::rest::IteratorRange::begin` [Table 8.29 describes the interface `ara::rest::IteratorRange::begin.`] ([RS\\_CM\\_00300](#))

### 8.4.6 end

<b>Service name:</b>	<code>ara::rest::IteratorRange::end</code>
<b>Type:</b>	Non-member function
<b>Syntax:</b>	<code>friend Iterator end(const IteratorRange &amp;r)</code>
<b>Function param:</b>	<code>r</code>   an <code>IteratorRange</code>
<b>Return value:</b>	the end of the sequence
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/iterator.h</code>
<b>Namespace:</b>	<code>ara::rest::IteratorRange</code>
<b>Description:</b>	Non-member equivalent of <code>IteratorRange::end()</code>

**Table 8.30: `ara::rest::IteratorRange::end`**

[SWS\_REST\_02388] `ara::rest::IteratorRange::end` [Table 8.30 describes the interface `ara::rest::IteratorRange::end.`] ([RS\\_CM\\_00300](#))

## 8.5 `ara::rest::MoveIteratorRange`

[SWS\_REST\_02395] [`ara::rest::MoveIteratorRange` class shall be declared in the `ara/rest/iterator.h` header file:

```
1     template <typename IterT >
2     class ara::rest::MoveIteratorRange;
```

] ([RS\\_CM\\_00300](#))

### 8.5.1 MoveIterator

<b>Name:</b>	MoveIterator
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::MoveIteratorRange< IterT >::MoveIterator = IterT
<b>Header file:</b>	ara/rest/iterator.h
<b>Class:</b>	ara::rest::MoveIteratorRange
<b>Description:</b>	Type of the underlying pair of movable iterators.

**Table 8.31: ara::rest::MoveIteratorRange::MoveIterator**

[SWS\_REST\_02397] **Iterator** [Table 8.31 describes the type alias `ara::rest::MoveIteratorRange::MoveIterator`.] ([RS\\_CM\\_00300](#))

### 8.5.2 MoveIteratorRange

<b>Service name:</b>	ara::rest::MoveIteratorRange::MoveIteratorRange	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ara::rest::MoveIteratorRange< IterT >::MoveIteratorRange(MoveIterator first, MoveIterator last)	
<b>Function param:</b>	first	an iterator denoting the start of the sequence
<b>Function param:</b>	last	an iterator denoting the end of the sequence
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/iterator.h	
<b>Class:</b>	ara::rest::MoveIteratorRange	
<b>Description:</b>	Constructs an MoveIteratorRange from a pair of movable iterators. For convenient construction, see MakeMoveIteratorRange().	

**Table 8.32: ara::rest::MoveIteratorRange::MoveIteratorRange**

[SWS\_REST\_02398] **ara::rest::MoveIteratorRange::IteratorRange** [Table 8.32 describes the interface `ara::rest::MoveIteratorRange::MoveIteratorRange`.] ([RS\\_CM\\_00300](#))

### 8.5.3 begin

<b>Service name:</b>	ara::rest::MoveIteratorRange::begin
<b>Type:</b>	Member function
<b>Syntax:</b>	MoveIterator ara::rest::MoveIteratorRange< IterT >::begin() const
<b>Function param:</b>	None
<b>Return value:</b>	a movable iterator
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/iterator.h
<b>Class:</b>	ara::rest::MoveIteratorRange

<b>Description:</b>	Returns the start of the sequence.
---------------------	------------------------------------

**Table 8.33: ara::rest::MoveIteratorRange::begin**

[SWS\_REST\_02399] **ara::rest::MoveIteratorRange::begin** [Table 8.33 describes the interface `ara::rest::MoveIteratorRange::begin.`] ([RS\\_CM\\_00300](#))

#### 8.5.4 end

<b>Service name:</b>	ara::rest::MoveIteratorRange::end
<b>Type:</b>	Member function
<b>Syntax:</b>	MoveIterator ara::rest::MoveIteratorRange< IterT >::end() const
<b>Function param:</b>	None
<b>Return value:</b>	a movable iterator
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/iterator.h
<b>Class:</b>	ara::rest::MoveIteratorRange
<b>Description:</b>	Returns the end of the sequence.

**Table 8.34: ara::rest::MoveIteratorRange::end**

[SWS\_REST\_02400] **ara::rest::MoveIteratorRange::end** [Table 8.34 describes the interface `ara::rest::MoveIteratorRange::end.`] ([RS\\_CM\\_00300](#))

#### 8.5.5 begin

<b>Service name:</b>	ara::rest::MoveIteratorRange::begin
<b>Type:</b>	Non-member function
<b>Syntax:</b>	friend MoveIterator begin(const MoveIteratorRange &r)
<b>Function param:</b>	r                                    a MoveIteratorRange
<b>Return value:</b>	the start of the sequence
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/iterator.h
<b>Namespace:</b>	ara::rest::MoveIteratorRange
<b>Description:</b>	Non-member equivalent of MoveIteratorRange::begin()

**Table 8.35: ara::rest::MoveIteratorRange::begin**

[SWS\_REST\_02401] **ara::rest::MoveIteratorRange::begin** [Table 8.35 describes the interface `ara::rest::MoveIteratorRange::begin.`] ([RS\\_CM\\_00300](#))

#### 8.5.6 end

<b>Service name:</b>	ara::rest::MoveIteratorRange::end
----------------------	-----------------------------------

<b>Type:</b>	Non-member function
<b>Syntax:</b>	friend MoveIterator end(const MoveIteratorRange &r)
<b>Function param:</b>	r                      a MoveIteratorRange
<b>Return value:</b>	the end of the sequence
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/iterator.h
<b>Namespace:</b>	ara::rest::MoveIteratorRange
<b>Description:</b>	Non-member equivalent of MoveIteratorRange::end()

**Table 8.36: ara::rest::MoveIteratorRange::end**

[SWS\_REST\_02402] `ara::rest::MoveIteratorRange::end` [Table 8.36 describes the interface `ara::rest::MoveIteratorRange::end`.] ([RS\\_CM\\_00300](#))

## 8.6 ara::rest::Matches

[SWS\_REST\_02027] [`ara::rest::Matches` class shall be declared in the `ara/rest/routing.h` header file:

```
1           class ara::rest::Matches;
```

] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00309](#))

### 8.6.1 MatchRange

<b>Name:</b>	MatchRange
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::Matches::MatchRange = IteratorRange<unspecified_iterator_type>
<b>Header file:</b>	ara/rest/routing.h
<b>Class:</b>	ara::rest::Matches
<b>Description:</b>	An IteratorRange of all pattern matches for this Route.

**Table 8.37: ara::rest::Matches::MatchRange**

[SWS\_REST\_02028] `MatchRange` [Table 8.37 describes the type alias `ara::rest::Matches::MatchRange`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00309](#))

### 8.6.2 Count

<b>Service name:</b>	ara::rest::Matches::Count
<b>Type:</b>	Member function
<b>Syntax:</b>	std::size_t ara::rest::Matches::Count() const
<b>Function param:</b>	None
<b>Return value:</b>	the number of URI matches

<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/routing.h
<b>Class:</b>	ara::rest::Matches
<b>Description:</b>	Provides the number of URI wildcard matches after applying a pattern to a route.

**Table 8.38: ara::rest::Matches::Count**

[SWS\_REST\_02029] **ara::rest::Matches::Count** [Table 8.38 describes the interface [ara::rest::Matches::Count](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00309](#))

### 8.6.3 Get

<b>Service name:</b>	ara::rest::Matches::Get	
<b>Type:</b>	Member function	
<b>Syntax:</b>	const Match& ara::rest::Matches::Get(std::size_t i) const	
<b>Function param:</b>	i	index to the i'th URI wildcard match
<b>Return value:</b>	return type	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/routing.h	
<b>Class:</b>	ara::rest::Matches	
<b>Description:</b>	Provides access to a specific URI match.	

**Table 8.39: ara::rest::Matches::Get**

[SWS\_REST\_02030] **ara::rest::Matches::Get** [Table 8.39 describes the interface [ara::rest::Matches::Get](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00309](#))

### 8.6.4 Get

<b>Service name:</b>	ara::rest::Matches::Get	
<b>Type:</b>	Member function	
<b>Syntax:</b>	MatchRange ara::rest::Matches::Get() const	
<b>Function param:</b>	None	
<b>Return value:</b>	a range of URI matches	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/routing.h	
<b>Class:</b>	ara::rest::Matches	
<b>Description:</b>	Provides access to the sequence of URI (wildcard) matches. After this route has been matched against a given request, all wildcard URI matches are accessible with this range.	

**Table 8.40: ara::rest::Matches::Get**

[SWS\_REST\_02031] **ara::rest::Matches::Get** [Table 8.40 describes the interface [ara::rest::Matches::Get](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00309](#))

## 8.7 ara::rest::Match

[SWS\_REST\_02033] [ara::rest::Match class shall be declared in the ara/rest/routing.h header file:

```
1         class ara::rest::Match;
```

](RS\_CM\_00300, RS\_CM\_00309)

### 8.7.1 Get

<b>Service name:</b>	ara::rest::Match::Get
<b>Type:</b>	Member function
<b>Syntax:</b>	StringView ara::rest::Match::Get() const
<b>Function param:</b>	None
<b>Return value:</b>	a string of the matches path segment
<b>Exceptions:</b>	noexcept (std::is_nothrow_constructible< StringView >::value)
<b>Header file:</b>	ara/rest/routing.h
<b>Class:</b>	ara::rest::Match
<b>Description:</b>	Returns a path segment as a string.

**Table 8.41: ara::rest::Match::Get**

[SWS\_REST\_02034] ara::rest::Match::Get [Table 8.41 describes the interface ara::rest::Match::Get.](RS\_CM\_00300, RS\_CM\_00309)

### 8.7.2 GetAs

<b>Service name:</b>	ara::rest::Match::GetAs
<b>Type:</b>	Member function template
<b>Syntax:</b>	template <typename T > T ara::rest::Match::GetAs(T &&def={})
<b>Function param:</b>	def   if conversion fails,
<b>Return value:</b>	The converted value of conversion succeeded, otherwise it returns the function argument.
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/routing.h
<b>Class:</b>	ara::rest::Match
<b>Description:</b>	Returns a type-converted path segment. Applies a type conversion on the matched path segment. The conversion result is assigned to the function argument which is subsequently returned. If conversion fails the function argument is returned unchanged. So either form is valid: GetAs<string>(), GetAs(string{my_allocator}), GetAs<string>("conversion failed")

**Table 8.42: ara::rest::Match::GetAs**

[SWS\_REST\_02035] `ara::rest::Match::GetAs` [Table 8.42 describes the interface `ara::rest::Match::GetAs`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00309](#))

## 8.8 `ara::rest::ogm::Array`

[SWS\_REST\_02036] [`ara::rest::ogm::Array` class shall be declared in the `ara/rest/ogm/array.h` header file:

```
1      class ara::rest::ogm::Array : public ara::rest::ogm::Value;
    ] (RS\_CM\_00300, RS\_CM\_00305, RS\_CM\_00306, RS\_CM\_00307)
```

### 8.8.1 SelfType

<b>Name:</b>	SelfType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	<code>using ara::rest::ogm::Array::SelfType = Array</code>
<b>Header file:</b>	<code>ara/rest/ogm/array.h</code>
<b>Class:</b>	<code>ara::rest::ogm::Array</code>
<b>Description:</b>	Its own type.

**Table 8.43: `ara::rest::ogm::Array::SelfType`**

[SWS\_REST\_02037] **SelfType** [Table 8.43 describes the type alias `ara::rest::ogm::Array::SelfType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.8.2 ParentType

<b>Name:</b>	ParentType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	<code>using ara::rest::ogm::Array::ParentType = Value</code>
<b>Header file:</b>	<code>ara/rest/ogm/array.h</code>
<b>Class:</b>	<code>ara::rest::ogm::Array</code>
<b>Description:</b>	Type of its parent in the OGM type hierarchy.

**Table 8.44: `ara::rest::ogm::Array::ParentType`**

[SWS\_REST\_02038] **ParentType** [Table 8.44 describes the type alias `ara::rest::ogm::Array::ParentType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.8.3 Iterator



<b>Name:</b>	Iterator
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Array::Iterator = unspecified_iterator_type
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	A forward iterator of the represented set values.

**Table 8.45: ara::rest::ogm::Array::Iterator**

[SWS\_REST\_02039] **Iterator** [Table 8.45 describes the type alias `ara::rest::ogm::Array::Iterator`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.4 ConstIterator

<b>Name:</b>	ConstIterator
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Array::ConstIterator = unspecified_iterator_type
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	Value iterator.

**Table 8.46: ara::rest::ogm::Array::ConstIterator**

[SWS\_REST\_02040] **ConstIterator** [Table 8.46 describes the type alias `ara::rest::ogm::Array::ConstIterator`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.5 ValueRange

<b>Name:</b>	ValueRange
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Array::ValueRange = IteratorRange<Iterator>
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	Iterator range.

**Table 8.47: ara::rest::ogm::Array::ValueRange**

[SWS\_REST\_02041] **ValueRange** [Table 8.47 describes the type alias `ara::rest::ogm::Array::ValueRange`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.6 ConstValueRange

<b>Name:</b>	ConstValueRange
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Array::ConstValueRange = IteratorRange<ConstIterator>
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	Iterator range.

**Table 8.48: ara::rest::ogm::Array::ConstValueRange**

[SWS\_REST\_02042] **ConstValueRange** [Table 8.48 describes the type alias [ara::rest::ogm::Array::ConstValueRange.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.7 MoveRange

<b>Name:</b>	MoveRange
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Array::MoveRange = IteratorRange<MoveIterator>
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	MoveIterator range.

**Table 8.49: ara::rest::ogm::Array::MoveRange**

[SWS\_REST\_02403] **MoveRange** [Table 8.49 describes the type alias [ara::rest::ogm::Array::MoveRange.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.8 GetParent

<b>Service name:</b>	ara::rest::ogm::Array::GetParent
<b>Type:</b>	Member function
<b>Syntax:</b>	Node* ara::rest::ogm::Array::GetParent()
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to its parent node
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	Returns a (strongly-typed) pointer to its parent node.

**Table 8.50: ara::rest::ogm::Array::GetParent**

[SWS\_REST\_02043] **ara::rest::ogm::Array::GetParent** [Table 8.50 describes the interface `ara::rest::ogm::Array::GetParent.`]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.9 GetParent

<b>Service name:</b>	<code>ara::rest::ogm::Array::GetParent</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>const Node* ara::rest::ogm::Array::GetParent() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to its parent node
<b>Exceptions:</b>	<code>noexcept</code>
<b>Header file:</b>	<code>ara/rest/ogm/array.h</code>
<b>Class:</b>	<code>ara::rest::ogm::Array</code>
<b>Description:</b>	Returns a (strongly-typed) pointer to its parent node.

**Table 8.51: ara::rest::ogm::Array::GetParent**

[SWS\_REST\_02044] **ara::rest::ogm::Array::GetParent** [Table 8.51 describes the interface `ara::rest::ogm::Array::GetParent.`]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.10 HasParent

<b>Service name:</b>	<code>ara::rest::ogm::Array::HasParent</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>bool ara::rest::ogm::Array::HasParent() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	true if this node has a structural parent
<b>Exceptions:</b>	<code>noexcept</code>
<b>Header file:</b>	<code>ara/rest/ogm/array.h</code>
<b>Class:</b>	<code>ara::rest::ogm::Array</code>
<b>Description:</b>	Denotes whether this node has a structural parent.

**Table 8.52: ara::rest::ogm::Array::HasParent**

[SWS\_REST\_02045] **ara::rest::ogm::Array::HasParent** [Table 8.52 describes the interface `ara::rest::ogm::Array::HasParent.`]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.11 GetSize

<b>Service name:</b>	<code>ara::rest::ogm::Array::GetSize</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>std::size_t ara::rest::ogm::Array::GetSize() const</code>

<b>Function param:</b>	None
<b>Return value:</b>	the number of array elements
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	Returns the number of elements.

**Table 8.53: ara::rest::ogm::Array::GetSize**

[SWS\_REST\_02046] **ara::rest::ogm::Array::GetSize** [Table 8.53 describes the interface `ara::rest::ogm::Array::GetSize`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.12 IsEmpty

<b>Service name:</b>	ara::rest::ogm::Array::IsEmpty
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>bool ara::rest::ogm::Array::IsEmpty() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	true if the array holds no elements
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	Returns whether the array holds no elements.

**Table 8.54: ara::rest::ogm::Array::IsEmpty**

[SWS\_REST\_02047] **ara::rest::ogm::Array::IsEmpty** [Table 8.54 describes the interface `ara::rest::ogm::Array::IsEmpty`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.13 GetValue

<b>Service name:</b>	ara::rest::ogm::Array::GetValue
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Value&amp; ara::rest::ogm::Array::GetValue(std::size_t index)</code>
<b>Function param:</b>	index   an integral index into the array
<b>Return value:</b>	a reference to a Value
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	Returns a Value at a specific index. If the index is out-of-bounds, the result is undefined.

**Table 8.55: ara::rest::ogm::Array::GetValue**

[SWS\_REST\_02048] **ara::rest::ogm::Array::GetValue** [Table 8.55 describes the interface `ara::rest::ogm::Array::GetValue`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.14 GetValue

<b>Service name:</b>	ara::rest::ogm::Array::GetValue	
<b>Type:</b>	Member function	
<b>Syntax:</b>	const Value& ara::rest::ogm::Array::GetValue(std::size_t index) const	
<b>Function param:</b>	index	an integral index into the array
<b>Return value:</b>	a reference to a Value	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/array.h	
<b>Class:</b>	ara::rest::ogm::Array	
<b>Description:</b>	Returns a Value at a specific index. If the index is out-of-bounds, the result is undefined.	

**Table 8.56: ara::rest::ogm::Array::GetValue**

[SWS\_REST\_02049] **ara::rest::ogm::Array::GetValue** [Table 8.56 describes the interface `ara::rest::ogm::Array::GetValue`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.15 GetValues

<b>Service name:</b>	ara::rest::ogm::Array::GetValues
<b>Type:</b>	Member function
<b>Syntax:</b>	ValueRange ara::rest::ogm::Array::GetValues()
<b>Function param:</b>	None
<b>Return value:</b>	an iterator range of values
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	Returns a range of values.

**Table 8.57: ara::rest::ogm::Array::GetValues**

[SWS\_REST\_02050] **ara::rest::ogm::Array::GetValues** [Table 8.57 describes the interface `ara::rest::ogm::Array::GetValues`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.16 GetValues

<b>Service name:</b>	ara::rest::ogm::Array::GetValues
<b>Type:</b>	Member function
<b>Syntax:</b>	ConstValueRange ara::rest::ogm::Array::GetValues() const
<b>Function param:</b>	None
<b>Return value:</b>	an iterator range of values
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	Returns a range of values.

**Table 8.58: ara::rest::ogm::Array::GetValues**

[SWS\_REST\_02051] **ara::rest::ogm::Array::GetValues** [Table 8.58 describes the interface `ara::rest::ogm::Array::GetValues.`] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.17 Append

<b>Service name:</b>	ara::rest::ogm::Array::Append
<b>Type:</b>	Member function
<b>Syntax:</b>	void ara::rest::ogm::Array::Append(Pointer< Value > &&v)
<b>Function param:</b>	v   a Pointer to a value
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	Appends a Value object to the array.

**Table 8.59: ara::rest::ogm::Array::Append**

[SWS\_REST\_02052] **ara::rest::ogm::Array::Append** [Table 8.59 describes the interface `ara::rest::ogm::Array::Append.`] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.18 Insert

<b>Service name:</b>	ara::rest::ogm::Array::Insert
<b>Type:</b>	Member function
<b>Syntax:</b>	void ara::rest::ogm::Array::Insert(Iterator iter, Pointer< Value > &&v)
<b>Function param:</b>	iter   an Array iterator
<b>Function param:</b>	v   a value to insert.
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/array.h

<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	Inserts a Value at a specific position into the Array. Inserts a value before the element pointed to by the iterator argument. To insert a Value ownership has to be passed to the Array

**Table 8.60: ara::rest::ogm::Array::Insert**

[SWS\_REST\_02053] **ara::rest::ogm::Array::Insert** [Table 8.60 describes the interface `ara::rest::ogm::Array::Insert`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.19 Remove

<b>Service name:</b>	ara::rest::ogm::Array::Remove
<b>Type:</b>	Member function
<b>Syntax:</b>	Iterator ara::rest::ogm::Array::Remove(Iterator iter)
<b>Function param:</b>	iter   an iterator pointing to an array element
<b>Return value:</b>	an iterator pointing to the element following the one just removed.
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	Removes an element from the array. Removes the element pointed to by the iterator argument

**Table 8.61: ara::rest::ogm::Array::Remove**

[SWS\_REST\_02054] **ara::rest::ogm::Array::Remove** [Table 8.61 describes the interface `ara::rest::ogm::Array::Remove`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.20 Release

<b>Service name:</b>	ara::rest::ogm::Array::Release
<b>Type:</b>	Member function
<b>Syntax:</b>	std::pair<Iterator, Pointer<Value> > ara::rest::ogm::Array::Release(Iterator iter)
<b>Function param:</b>	iter   an iterator pointing to the element to be removed
<b>Return value:</b>	a pair of the iterator pointing to the element following the one just deleted and a pointer to the element
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	Similar to Remove but does not destroy the removed element. Instead of destroying the removed element, ownership is passed back to the user

**Table 8.62: ara::rest::ogm::Array::Release**

[SWS\_REST\_02055] **ara::rest::ogm::Array::Release** [Table 8.62 describes the interface `ara::rest::ogm::Array::Release`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.21 Replace

<b>Service name:</b>	ara::rest::ogm::Array::Replace	
<b>Type:</b>	Member function	
<b>Syntax:</b>	Pointer<Value> ara::rest::ogm::Array::Replace(Iterator iter, Pointer< Value > &&v)	
<b>Function param:</b>	iter	an iterator pointing to the element to be removed
<b>Function param:</b>	v	a pointer to the value to replace the one pointed to by iter
<b>Return value:</b>	a pointer to the old array element	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/array.h	
<b>Class:</b>	ara::rest::ogm::Array	
<b>Description:</b>	Replaces an element by a new one without the destroying the old one. Replaces an array element without destroying it. Instead the replaced element is returned. Effectively, ownership is passed back to the user.	

**Table 8.63: ara::rest::ogm::Array::Replace**

[SWS\_REST\_02056] **ara::rest::ogm::Array::Replace** [Table 8.63 describes the interface `ara::rest::ogm::Array::Replace`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.22 Clear

<b>Service name:</b>	ara::rest::ogm::Array::Clear
<b>Type:</b>	Member function
<b>Syntax:</b>	void ara::rest::ogm::Array::Clear()
<b>Function param:</b>	None
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	Removes and destroys all elements of the array.

**Table 8.64: ara::rest::ogm::Array::Clear**

[SWS\_REST\_02057] **ara::rest::ogm::Array::Clear** [Table 8.64 describes the interface `ara::rest::ogm::Array::Clear`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))



### 8.8.23 Make

<b>Service name:</b>	ara::rest::ogm::Array::Make	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>template &lt;typename... Ts&gt; static Pointer&lt;SelfType&gt; ara::rest::ogm::Array::Make(Ts &amp;&amp;...ts)</pre>	
<b>Function param:</b>	ts	constructor arguments forwarded to the constructor of this type
<b>Return value:</b>	a pointer to a node of type SelfType	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/array.h	
<b>Class:</b>	ara::rest::ogm::Array	
<b>Description:</b>	Creates a node of type SelfType.	

**Table 8.65: ara::rest::ogm::Array::Make**

[SWS\_REST\_02058] **ara::rest::ogm::Array::Make** [Table 8.65 describes the interface `ara::rest::ogm::Array::Make`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.24 Make

<b>Service name:</b>	ara::rest::ogm::Array::Make	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>template &lt;typename... Ts&gt; static Pointer&lt;SelfType&gt; ara::rest::ogm::Array::Make(Allocator *alloc, Ts &amp;&amp;...ts)</pre>	
<b>Function param:</b>	alloc	an allocator to use to construct this node
<b>Function param:</b>	ts	constructor arguments forwarded to the constructor of this type
<b>Return value:</b>	a pointer to a node of type SelfType	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/array.h	
<b>Class:</b>	ara::rest::ogm::Array	
<b>Description:</b>	Creates a node of type SelfType.	

**Table 8.66: ara::rest::ogm::Array::Make**

[SWS\_REST\_02059] **ara::rest::ogm::Array::Make** [Table 8.66 describes the interface `ara::rest::ogm::Array::Make`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.8.25 Array

<b>Service name:</b>	ara::rest::ogm::Array::Array
<b>Type:</b>	Member function

<b>Syntax:</b>	template <typename... Ts> ara::rest::ogm::Array::Array(Pointer< Ts > &&...ts)
<b>Function param:</b>	ts   OGM objects to insert into the array
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	Constructs an Array.

**Table 8.67: ara::rest::ogm::Array::Array**

[SWS\_REST\_02060] ara::rest::ogm::Array::Array [Table 8.67 describes the interface ara::rest::ogm::Array::Array.] (RS\_CM\_00300, RS\_CM\_00305, RS\_CM\_00306, RS\_CM\_00307, RS\_CM\_00308)

## 8.8.26 Array

<b>Service name:</b>	ara::rest::ogm::Array::Array
<b>Type:</b>	Member function
<b>Syntax:</b>	template <typename... Ts> ara::rest::ogm::Array::Array(Allocator *alloc, Pointer< Ts > &&...ts)
<b>Function param:</b>	alloc   an allocator
<b>Function param:</b>	ts   OGM objects to insert into the array
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/array.h
<b>Class:</b>	ara::rest::ogm::Array
<b>Description:</b>	Constructs an Array.

**Table 8.68: ara::rest::ogm::Array::Array**

[SWS\_REST\_02061] ara::rest::ogm::Array::Array [Table 8.68 describes the interface ara::rest::ogm::Array::Array.] (RS\_CM\_00300, RS\_CM\_00305, RS\_CM\_00306, RS\_CM\_00307, RS\_CM\_00308)

## 8.9 ara::rest::ogm::Field

[SWS\_REST\_02062] [ara::rest::ogm::Field class shall be declared in the ara/rest/ogm/field.h header file:

```
1     class ara::rest::ogm::Field : public ara::rest::ogm::Node;
```

] (RS\_CM\_00300, RS\_CM\_00305, RS\_CM\_00306, RS\_CM\_00307)

### 8.9.1 SelfType

<b>Name:</b>	SelfType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Field::SelfType = Field
<b>Header file:</b>	ara/rest/ogm/field.h
<b>Class:</b>	ara::rest::ogm::Field
<b>Description:</b>	Its own type.

**Table 8.69: ara::rest::ogm::Field::SelfType**

[SWS\_REST\_02063] **SelfType** [Table 8.69 describes the type alias `ara::rest::ogm::Field::SelfType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.9.2 ParentType

<b>Name:</b>	ParentType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Field::ParentType = Node
<b>Header file:</b>	ara/rest/ogm/field.h
<b>Class:</b>	ara::rest::ogm::Field
<b>Description:</b>	Type of its parent in the OGM type hierarchy.

**Table 8.70: ara::rest::ogm::Field::ParentType**

[SWS\_REST\_02064] **ParentType** [Table 8.70 describes the type alias `ara::rest::ogm::Field::ParentType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.9.3 GetParent

<b>Service name:</b>	ara::rest::ogm::Field::GetParent
<b>Type:</b>	Member function
<b>Syntax:</b>	Node* ara::rest::ogm::Field::GetParent()
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to its parent node
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/field.h
<b>Class:</b>	ara::rest::ogm::Field
<b>Description:</b>	Returns a (strongly-typed) pointer to its parent node.

**Table 8.71: ara::rest::ogm::Field::GetParent**

[SWS\_REST\_02065] **ara::rest::ogm::Field::GetParent** [Table 8.71 describes the interface `ara::rest::ogm::Field::GetParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.9.4 GetParent

<b>Service name:</b>	ara::rest::ogm::Field::GetParent
<b>Type:</b>	Member function
<b>Syntax:</b>	const Node* ara::rest::ogm::Field::GetParent() const
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to its parent node
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/field.h
<b>Class:</b>	ara::rest::ogm::Field
<b>Description:</b>	Returns a (strongly-typed) pointer to its parent node.

**Table 8.72: ara::rest::ogm::Field::GetParent**

[SWS\_REST\_02066] **ara::rest::ogm::Field::GetParent** [Table 8.72 describes the interface `ara::rest::ogm::Field::GetParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.9.5 HasParent

<b>Service name:</b>	ara::rest::ogm::Field::HasParent
<b>Type:</b>	Member function
<b>Syntax:</b>	bool ara::rest::ogm::Field::HasParent() const
<b>Function param:</b>	None
<b>Return value:</b>	true if this node has a structural parent
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/field.h
<b>Class:</b>	ara::rest::ogm::Field
<b>Description:</b>	Denotes whether this node has a structural parent.

**Table 8.73: ara::rest::ogm::Field::HasParent**

[SWS\_REST\_02067] **ara::rest::ogm::Field::HasParent** [Table 8.73 describes the interface `ara::rest::ogm::Field::HasParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.9.6 GetName

<b>Service name:</b>	ara::rest::ogm::Field::GetName
<b>Type:</b>	Member function
<b>Syntax:</b>	const StringView& ara::rest::ogm::Field::GetName() const
<b>Function param:</b>	None
<b>Return value:</b>	a name
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/field.h
<b>Class:</b>	ara::rest::ogm::Field

<b>Description:</b>	Return the name of a Field. Fields names are immutable. To set a different name a new Field must be inserted.
---------------------	---

**Table 8.74: ara::rest::ogm::Field::GetName**

[SWS\_REST\_02068] **ara::rest::ogm::Field::GetName** [Table 8.74 describes the interface `ara::rest::ogm::Field::GetName`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.9.7 GetValue

<b>Service name:</b>	ara::rest::ogm::Field::GetValue
<b>Type:</b>	Member function
<b>Syntax:</b>	const Value& ara::rest::ogm::Field::GetValue() const
<b>Function param:</b>	None
<b>Return value:</b>	a reference to the current field value
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/field.h
<b>Class:</b>	ara::rest::ogm::Field
<b>Description:</b>	Returns the value represented by a Field.

**Table 8.75: ara::rest::ogm::Field::GetValue**

[SWS\_REST\_02069] **ara::rest::ogm::Field::GetValue** [Table 8.75 describes the interface `ara::rest::ogm::Field::GetValue`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.9.8 GetValue

<b>Service name:</b>	ara::rest::ogm::Field::GetValue
<b>Type:</b>	Member function
<b>Syntax:</b>	Value& ara::rest::ogm::Field::GetValue()
<b>Function param:</b>	None
<b>Return value:</b>	a reference to the current field value
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/field.h
<b>Class:</b>	ara::rest::ogm::Field
<b>Description:</b>	Returns the value represented by a Field.

**Table 8.76: ara::rest::ogm::Field::GetValue**

[SWS\_REST\_02070] **ara::rest::ogm::Field::GetValue** [Table 8.76 describes the interface `ara::rest::ogm::Field::GetValue`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.9.9 SetValue

<b>Service name:</b>	ara::rest::ogm::Field::SetValue	
<b>Type:</b>	Member function	
<b>Syntax:</b>	void ara::rest::ogm::Field::SetValue(Pointer< Value > &&v)	
<b>Function param:</b>	v	a new Value
<b>Return value:</b>	None	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/ogm/field.h	
<b>Class:</b>	ara::rest::ogm::Field	
<b>Description:</b>	Sets a new value. The previous value is destroyed	

**Table 8.77: ara::rest::ogm::Field::SetValue**

[SWS\_REST\_02071] **ara::rest::ogm::Field::SetValue** [Table 8.77 describes the interface `ara::rest::ogm::Field::SetValue`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.9.10 ReplaceValue

<b>Service name:</b>	ara::rest::ogm::Field::ReplaceValue	
<b>Type:</b>	Member function	
<b>Syntax:</b>	Pointer<Value> ara::rest::ogm::Field::ReplaceValue(Pointer< Value > &&v)	
<b>Function param:</b>	v	a new Value
<b>Return value:</b>	the old value	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/ogm/field.h	
<b>Class:</b>	ara::rest::ogm::Field	
<b>Description:</b>	Sets a new value and returns the old one.	

**Table 8.78: ara::rest::ogm::Field::ReplaceValue**

[SWS\_REST\_02072] **ara::rest::ogm::Field::ReplaceValue** [Table 8.78 describes the interface `ara::rest::ogm::Field::ReplaceValue`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.9.11 Make

<b>Service name:</b>	ara::rest::ogm::Field::Make	
<b>Type:</b>	Member function	
<b>Syntax:</b>	template <typename... Ts> static Pointer<SelfType> ara::rest::ogm::Field::Make(Ts &&...ts)	
<b>Function param:</b>	ts	constructor arguments forwarded to the constructor of this type

<b>Return value:</b>	a pointer to a node of type SelfType
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/field.h
<b>Class:</b>	ara::rest::ogm::Field
<b>Description:</b>	Creates a node of type SelfType.

**Table 8.79: ara::rest::ogm::Field::Make**

[SWS\_REST\_02073] **ara::rest::ogm::Field::Make** [Table 8.79 describes the interface `ara::rest::ogm::Field::Make.`]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.9.12 Make

<b>Service name:</b>	ara::rest::ogm::Field::Make	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>template &lt;typename... Ts&gt; static Pointer&lt;SelfType&gt; ara::rest::ogm::Field::Make(Allocator *alloc, Ts &amp;&amp;...ts)</pre>	
<b>Function param:</b>	alloc	an allocator to use to construct this node
<b>Function param:</b>	ts	constructor arguments forwarded to the constructor of this type
<b>Return value:</b>	a pointer to a node of type SelfType	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/field.h	
<b>Class:</b>	ara::rest::ogm::Field	
<b>Description:</b>	Creates a node of type SelfType.	

**Table 8.80: ara::rest::ogm::Field::Make**

[SWS\_REST\_02074] **ara::rest::ogm::Field::Make** [Table 8.80 describes the interface `ara::rest::ogm::Field::Make.`]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.9.13 Field

<b>Service name:</b>	ara::rest::ogm::Field::Field	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>ara::rest::ogm::Field::Field(const String &amp;name, Pointer&lt; Value &gt; &amp;&amp;value)</pre>	
<b>Function param:</b>	name	name of this Field
<b>Function param:</b>	value	value object attached to this Field
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/field.h	
<b>Class:</b>	ara::rest::ogm::Field	

<b>Description:</b>	Constructs a Field.
---------------------	---------------------

**Table 8.81: ara::rest::ogm::Field::Field**

[SWS\_REST\_02075] **ara::rest::ogm::Field::Field** [Table 8.81 describes the interface `ara::rest::ogm::Field::Field`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.9.14 Field

<b>Service name:</b>	ara::rest::ogm::Field::Field	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ara::rest::ogm::Field::Field(Allocator *alloc, const String &key, Pointer< Value > &&val)	
<b>Function param:</b>	alloc	an allocator
<b>Function param:</b>	key	a field name
<b>Function param:</b>	val	a field value
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/field.h	
<b>Class:</b>	ara::rest::ogm::Field	
<b>Description:</b>	Constructs field, providing an allocator. The allocator argument may be used for internal allocation purposes.	

**Table 8.82: ara::rest::ogm::Field::Field**

[SWS\_REST\_02076] **ara::rest::ogm::Field::Field** [Table 8.82 describes the interface `ara::rest::ogm::Field::Field`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

## 8.10 ara::rest::ogm::Int

[SWS\_REST\_02077] [ara::rest::ogm::Int class shall be declared in the `ara/rest/ogm/int.h` header file:

```
1     class ara::rest::ogm::Int : public ara::rest::ogm::Value;
```

] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.10.1 SelfType

<b>Name:</b>	SelfType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Int::SelfType = Int
<b>Header file:</b>	ara/rest/ogm/int.h



<b>Class:</b>	ara::rest::ogm::Int
<b>Description:</b>	Its own type.

**Table 8.83: ara::rest::ogm::Int::SelfType**

[SWS\_REST\_02078] **SelfType** [Table 8.83 describes the type alias `ara::rest::ogm::Int::SelfType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.10.2 ParentType

<b>Name:</b>	ParentType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Int::ParentType = Value
<b>Header file:</b>	ara/rest/ogm/int.h
<b>Class:</b>	ara::rest::ogm::Int
<b>Description:</b>	Type of its parent in the OGM type hierarchy.

**Table 8.84: ara::rest::ogm::Int::ParentType**

[SWS\_REST\_02079] **ParentType** [Table 8.84 describes the type alias `ara::rest::ogm::Int::ParentType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.10.3 ValueType

<b>Name:</b>	ValueType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Int::ValueType = std::int64_t
<b>Header file:</b>	ara/rest/ogm/int.h
<b>Class:</b>	ara::rest::ogm::Int
<b>Description:</b>	Type of its corresponding C++ data type.

**Table 8.85: ara::rest::ogm::Int::ValueType**

[SWS\_REST\_02080] **ValueType** [Table 8.85 describes the type alias `ara::rest::ogm::Int::ValueType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.10.4 GetParent

<b>Service name:</b>	ara::rest::ogm::Int::GetParent
<b>Type:</b>	Member function
<b>Syntax:</b>	Node* ara::rest::ogm::Int::GetParent ()

<b>Function param:</b>	None
<b>Return value:</b>	a pointer to its parent node
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/int.h
<b>Class:</b>	ara::rest::ogm::Int
<b>Description:</b>	Returns a (strongly-typed) pointer to its parent node.

**Table 8.86: ara::rest::ogm::Int::GetParent**

[SWS\_REST\_02081] `ara::rest::ogm::Int::GetParent` [Table 8.86 describes the interface `ara::rest::ogm::Int::GetParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.10.5 GetParent

<b>Service name:</b>	ara::rest::ogm::Int::GetParent
<b>Type:</b>	Member function
<b>Syntax:</b>	const Node* ara::rest::ogm::Int::GetParent() const
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to its parent node
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/int.h
<b>Class:</b>	ara::rest::ogm::Int
<b>Description:</b>	Returns a (strongly-typed) pointer to its parent node.

**Table 8.87: ara::rest::ogm::Int::GetParent**

[SWS\_REST\_02082] `ara::rest::ogm::Int::GetParent` [Table 8.87 describes the interface `ara::rest::ogm::Int::GetParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.10.6 HasParent

<b>Service name:</b>	ara::rest::ogm::Int::HasParent
<b>Type:</b>	Member function
<b>Syntax:</b>	bool ara::rest::ogm::Int::HasParent() const
<b>Function param:</b>	None
<b>Return value:</b>	true if this node has a structural parent
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/int.h
<b>Class:</b>	ara::rest::ogm::Int
<b>Description:</b>	Denotes whether this node has a structural parent.

**Table 8.88: ara::rest::ogm::Int::HasParent**

[SWS\_REST\_02083] **ara::rest::ogm::Int::HasParent** [Table 8.88 describes the interface `ara::rest::ogm::Int::HasParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.10.7 GetValue

<b>Service name:</b>	ara::rest::ogm::Int::GetValue
<b>Type:</b>	Member function
<b>Syntax:</b>	ValueType ara::rest::ogm::Int::GetValue() const
<b>Function param:</b>	None
<b>Return value:</b>	a value of type ValueType
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/int.h
<b>Class:</b>	ara::rest::ogm::Int
<b>Description:</b>	Returns its value as a C++ data type.

**Table 8.89: ara::rest::ogm::Int::GetValue**

[SWS\_REST\_02084] **ara::rest::ogm::Int::GetValue** [Table 8.89 describes the interface `ara::rest::ogm::Int::GetValue`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.10.8 SetValue

<b>Service name:</b>	ara::rest::ogm::Int::SetValue
<b>Type:</b>	Member function
<b>Syntax:</b>	void ara::rest::ogm::Int::SetValue(ValueType v)
<b>Function param:</b>	v   a value
<b>Return value:</b>	None
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/int.h
<b>Class:</b>	ara::rest::ogm::Int
<b>Description:</b>	Sets the current value from a C++ data type.

**Table 8.90: ara::rest::ogm::Int::SetValue**

[SWS\_REST\_02085] **ara::rest::ogm::Int::SetValue** [Table 8.90 describes the interface `ara::rest::ogm::Int::SetValue`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.10.9 Make

<b>Service name:</b>	ara::rest::ogm::Int::Make
<b>Type:</b>	Member function

<b>Syntax:</b>	<pre>template &lt;typename... Ts&gt; static Pointer&lt;SelfType&gt; ara::rest::ogm::Int::Make(Ts &amp;&amp;...ts)</pre>	
<b>Function param:</b>	ts	constructor arguments forwarded to the constructor of this type
<b>Return value:</b>	a pointer to a node of type SelfType	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/int.h	
<b>Class:</b>	ara::rest::ogm::Int	
<b>Description:</b>	Creates a node of type SelfType.	

**Table 8.91: ara::rest::ogm::Int::Make**

[SWS\_REST\_02086] **ara::rest::ogm::Int::Make** [Table 8.91 describes the interface `ara::rest::ogm::Int::Make`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.10.10 Make

<b>Service name:</b>	ara::rest::ogm::Int::Make	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>template &lt;typename... Ts&gt; static Pointer&lt;SelfType&gt; ara::rest::ogm::Int::Make(Allocator *alloc, Ts &amp;&amp;...ts)</pre>	
<b>Function param:</b>	alloc	an allocator to use to construct this node
<b>Function param:</b>	ts	constructor arguments forwarded to the constructor of this type
<b>Return value:</b>	a pointer to a node of type SelfType	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/int.h	
<b>Class:</b>	ara::rest::ogm::Int	
<b>Description:</b>	Creates a node of type SelfType.	

**Table 8.92: ara::rest::ogm::Int::Make**

[SWS\_REST\_02087] **ara::rest::ogm::Int::Make** [Table 8.92 describes the interface `ara::rest::ogm::Int::Make`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.10.11 Int

<b>Service name:</b>	ara::rest::ogm::Int::Int	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>ara::rest::ogm::Int::Int(ValueType value=ValueType{})</pre>	
<b>Function param:</b>	value	an initial value
<b>Return value:</b>	None	

<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/int.h
<b>Class:</b>	ara::rest::ogm::Int
<b>Description:</b>	Connstructs an Int.

**Table 8.93: ara::rest::ogm::Int::Int**

[SWS\_REST\_02088] **ara::rest::ogm::Int::Int** [Table 8.93 describes the interface `ara::rest::ogm::Int::Int`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

## 8.11 ara::rest::ogm::Node

[SWS\_REST\_02089] [ara::rest::ogm::Node class shall be declared in the `ara/rest/ogm/node.h` header file:

```
1     class ara::rest::ogm::Node;
```

] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.11.1 SelfType

<b>Name:</b>	SelfType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Node::SelfType = Node
<b>Header file:</b>	ara/rest/ogm/node.h
<b>Class:</b>	ara::rest::ogm::Node
<b>Description:</b>	Type of this OGM node.

**Table 8.94: ara::rest::ogm::Node::SelfType**

[SWS\_REST\_02090] **SelfType** [Table 8.94 describes the type alias `ara::rest::ogm::Node::SelfType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.11.2 ParentType

<b>Name:</b>	ParentType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Node::ParentType = void
<b>Header file:</b>	ara/rest/ogm/node.h
<b>Class:</b>	ara::rest::ogm::Node
<b>Description:</b>	Type of its parent in the OGM type hierarchy.

**Table 8.95: ara::rest::ogm::Node::ParentType**

[SWS\_REST\_02091] **ParentType** [Table 8.95 describes the type alias `ara::rest::ogm::Node::ParentType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.11.3 GetParent

<b>Service name:</b>	<code>ara::rest::ogm::Node::GetParent</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>ParentType* ara::rest::ogm::Node::GetParent()</code>
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to its parent node
<b>Exceptions:</b>	<code>noexcept</code>
<b>Header file:</b>	<code>ara/rest/ogm/node.h</code>
<b>Class:</b>	<code>ara::rest::ogm::Node</code>
<b>Description:</b>	Returns a (strongly-typed) pointer to its parent node.

**Table 8.96: `ara::rest::ogm::Node::GetParent`**

[SWS\_REST\_02092] **`ara::rest::ogm::Node::GetParent`** [Table 8.96 describes the interface `ara::rest::ogm::Node::GetParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.11.4 GetParent

<b>Service name:</b>	<code>ara::rest::ogm::Node::GetParent</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>const ParentType* ara::rest::ogm::Node::GetParent() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to its parent node
<b>Exceptions:</b>	<code>noexcept</code>
<b>Header file:</b>	<code>ara/rest/ogm/node.h</code>
<b>Class:</b>	<code>ara::rest::ogm::Node</code>
<b>Description:</b>	Returns a (strongly-typed) pointer to its parent node.

**Table 8.97: `ara::rest::ogm::Node::GetParent`**

[SWS\_REST\_02093] **`ara::rest::ogm::Node::GetParent`** [Table 8.97 describes the interface `ara::rest::ogm::Node::GetParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.11.5 HasParent

<b>Service name:</b>	<code>ara::rest::ogm::Node::HasParent</code>
<b>Type:</b>	Member function

<b>Syntax:</b>	<code>bool ara::rest::ogm::Node::HasParent() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	true if this node has a parent
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/node.h
<b>Class:</b>	ara::rest::ogm::Node
<b>Description:</b>	Denotes whether this node has a structural parent.

**Table 8.98: ara::rest::ogm::Node::HasParent**

[SWS\_REST\_02094] **ara::rest::ogm::Node::HasParent** [Table 8.98 describes the interface `ara::rest::ogm::Node::HasParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.11.6 ~Node

<b>Service name:</b>	ara::rest::ogm::Node::~~Node
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>virtual ara::rest::ogm::Node::~~Node()</code>
<b>Function param:</b>	None
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/node.h
<b>Class:</b>	ara::rest::ogm::Node
<b>Description:</b>	Destructor.

**Table 8.99: ara::rest::ogm::Node::~~Node**

[SWS\_REST\_02095] **ara::rest::ogm::Node::~~Node** [Table 8.99 describes the interface `ara::rest::ogm::Node::~~Node`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.11.7 Node

<b>Service name:</b>	ara::rest::ogm::Node::Node
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>ara::rest::ogm::Node::Node(const Node &amp;)=delete</code>
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/node.h
<b>Class:</b>	ara::rest::ogm::Node
<b>Description:</b>	Non-copyable; copy with <code>ogm::Copy()</code>

**Table 8.100: ara::rest::ogm::Node::Node**

[SWS\_REST\_02096] `ara::rest::ogm::Node::Node` [Table 8.100 describes the interface `ara::rest::ogm::Node::Node`]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.11.8 operator=

<b>Service name:</b>	<code>ara::rest::ogm::Node::operator=</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Node&amp; ara::rest::ogm::Node::operator=(const Node &amp;)=delete</code>
<b>Return value:</b>	a value of type <code>Node &amp;</code>
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/ogm/node.h</code>
<b>Class:</b>	<code>ara::rest::ogm::Node</code>
<b>Description:</b>	Non-copy-assignable; copy with <code>ogm::Copy()</code>

**Table 8.101: `ara::rest::ogm::Node::operator=`**

[SWS\_REST\_02097] `ara::rest::ogm::Node::operator=` [Table 8.101 describes the interface `ara::rest::ogm::Node::operator=`]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.11.9 GetAllocator

<b>Service name:</b>	<code>ara::rest::ogm::Node::GetAllocator</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Allocator* ara::rest::ogm::Node::GetAllocator()</code>
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to an allocator
<b>Exceptions:</b>	<code>noexcept</code>
<b>Header file:</b>	<code>ara/rest/ogm/node.h</code>
<b>Class:</b>	<code>ara::rest::ogm::Node</code>
<b>Description:</b>	Returns a pointer to the allocator that manages this subtree.

**Table 8.102: `ara::rest::ogm::Node::GetAllocator`**

[SWS\_REST\_02098] `ara::rest::ogm::Node::GetAllocator` [Table 8.102 describes the interface `ara::rest::ogm::Node::GetAllocator`]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.11.10 GetAllocator

<b>Service name:</b>	<code>ara::rest::ogm::Node::GetAllocator</code>
<b>Type:</b>	Member function



<b>Syntax:</b>	<code>const Allocator* ara::rest::ogm::Node::GetAllocator() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to an allocator
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/node.h
<b>Class:</b>	ara::rest::ogm::Node
<b>Description:</b>	Returns a pointer to the allocator that manages this subtree.

**Table 8.103: ara::rest::ogm::Node::GetAllocator**

[SWS\_REST\_02099] **ara::rest::ogm::Node::GetAllocator** [Table 8.103 describes the interface `ara::rest::ogm::Node::GetAllocator`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.11.11 Node

<b>Service name:</b>	ara::rest::ogm::Node::Node
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>ara::rest::ogm::Node::Node()</code>
<b>Function param:</b>	None
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/node.h
<b>Class:</b>	ara::rest::ogm::Node
<b>Description:</b>	Constructs a node. Inaccessible to the user

**Table 8.104: ara::rest::ogm::Node::Node**

[SWS\_REST\_02100] **ara::rest::ogm::Node::Node** [Table 8.104 describes the interface `ara::rest::ogm::Node::Node`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

## 8.12 ara::rest::ogm::Object

[SWS\_REST\_02101] [ara::rest::ogm::Object class shall be declared in the `ara/rest/ogm/object.h` header file:

```
1     class ara::rest::ogm::Object : public ara::rest::ogm::Value;
    ] (RS\_CM\_00300, RS\_CM\_00305, RS\_CM\_00306, RS\_CM\_00307)
```

### 8.12.1 SelfType

<b>Name:</b>	SelfType
--------------	----------

<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Object::SelfType = Object
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Its own type.

**Table 8.105: ara::rest::ogm::Object::SelfType**

[SWS\_REST\_02102] **SelfType** [Table 8.105 describes the type alias `ara::rest::ogm::Object::SelfType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.12.2 ParentType

<b>Name:</b>	ParentType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Object::ParentType = Value
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Type of its parent in the OGM type hierarchy.

**Table 8.106: ara::rest::ogm::Object::ParentType**

[SWS\_REST\_02103] **ParentType** [Table 8.106 describes the type alias `ara::rest::ogm::Object::ParentType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.12.3 Iterator

<b>Name:</b>	Iterator
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Object::Iterator = unspecified_iterator_type
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Value iterator.

**Table 8.107: ara::rest::ogm::Object::Iterator**

[SWS\_REST\_02104] **Iterator** [Table 8.107 describes the type alias `ara::rest::ogm::Object::Iterator`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.4 ConstIterator

<b>Name:</b>	ConstIterator
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Object::ConstIterator = unspecified_iterator_type
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Value iterator.

**Table 8.108: ara::rest::ogm::Object::ConstIterator**

[SWS\_REST\_02105] **ConstIterator** [Table 8.108 describes the type alias `ara::rest::ogm::Object::ConstIterator`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.5 FieldRange

<b>Name:</b>	FieldRange
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Object::FieldRange = IteratorRange<Iterator>
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Iterator range.

**Table 8.109: ara::rest::ogm::Object::FieldRange**

[SWS\_REST\_02106] **FieldRange** [Table 8.109 describes the type alias `ara::rest::ogm::Object::FieldRange`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.6 ConstFieldRange

<b>Name:</b>	ConstFieldRange
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Object::ConstFieldRange = IteratorRange<ConstIterator>
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Iterator range.

**Table 8.110: ara::rest::ogm::Object::ConstFieldRange**

[SWS\_REST\_02107] **ConstFieldRange** [Table 8.110 describes the type alias `ara::rest::ogm::Object::ConstFieldRange`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.7 MoveFieldRange

<b>Name:</b>	MoveFieldRange
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Object::MoveFieldRange = IteratorRange<MoveIterator>
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Movelterator range.

**Table 8.111: ara::rest::ogm::Object::MoveFieldRange**

[SWS\_REST\_02404] **MoveFieldRange** [Table 8.111 describes the type alias `ara::rest::ogm::Object::MoveFieldRange`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.8 GetParent

<b>Service name:</b>	ara::rest::ogm::Object::GetParent
<b>Type:</b>	Member function
<b>Syntax:</b>	Node* ara::rest::ogm::Object::GetParent()
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to its parent node
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Returns a (strongly-typed) pointer to its parent node.

**Table 8.112: ara::rest::ogm::Object::GetParent**

[SWS\_REST\_02108] **ara::rest::ogm::Object::GetParent** [Table 8.112 describes the interface `ara::rest::ogm::Object::GetParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.9 GetParent

<b>Service name:</b>	ara::rest::ogm::Object::GetParent
<b>Type:</b>	Member function
<b>Syntax:</b>	const Node* ara::rest::ogm::Object::GetParent() const
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to its parent node
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Returns a (strongly-typed) pointer to its parent node.

**Table 8.113: ara::rest::ogm::Object::GetParent**

[SWS\_REST\_02109] **ara::rest::ogm::Object::GetParent** [Table 8.113 describes the interface `ara::rest::ogm::Object::GetParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.10 HasParent

<b>Service name:</b>	ara::rest::ogm::Object::HasParent
<b>Type:</b>	Member function
<b>Syntax:</b>	bool ara::rest::ogm::Object::HasParent() const
<b>Function param:</b>	None
<b>Return value:</b>	true if this node has a structural parent
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Denotes whether this node has a structural parent.

**Table 8.114: ara::rest::ogm::Object::HasParent**

[SWS\_REST\_02110] **ara::rest::ogm::Object::HasParent** [Table 8.114 describes the interface `ara::rest::ogm::Object::HasParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.11 GetSize

<b>Service name:</b>	ara::rest::ogm::Object::GetSize
<b>Type:</b>	Member function
<b>Syntax:</b>	std::size_t ara::rest::ogm::Object::GetSize() const
<b>Function param:</b>	None
<b>Return value:</b>	the number of array elements
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Returns the number of elements.

**Table 8.115: ara::rest::ogm::Object::GetSize**

[SWS\_REST\_02111] **ara::rest::ogm::Object::GetSize** [Table 8.115 describes the interface `ara::rest::ogm::Object::GetSize`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.12 IsEmpty

<b>Service name:</b>	ara::rest::ogm::Object::IsEmpty
<b>Type:</b>	Member function
<b>Syntax:</b>	bool ara::rest::ogm::Object::IsEmpty() const

<b>Function param:</b>	None
<b>Return value:</b>	true if the array holds no elements
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Returns whether the object holds no elements.

**Table 8.116: ara::rest::ogm::Object::IsEmpty**

[SWS\_REST\_02112] **ara::rest::ogm::Object::IsEmpty** [Table 8.116 describes the interface `ara::rest::ogm::Object::IsEmpty`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.13 GetFields

<b>Service name:</b>	ara::rest::ogm::Object::GetFields
<b>Type:</b>	Member function
<b>Syntax:</b>	FieldRange ara::rest::ogm::Object::GetFields()
<b>Function param:</b>	None
<b>Return value:</b>	an iterator range of fields
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Returns a range of fields.

**Table 8.117: ara::rest::ogm::Object::GetFields**

[SWS\_REST\_02113] **ara::rest::ogm::Object::GetFields** [Table 8.117 describes the interface `ara::rest::ogm::Object::GetFields`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.14 GetFields

<b>Service name:</b>	ara::rest::ogm::Object::GetFields
<b>Type:</b>	Member function
<b>Syntax:</b>	ConstFieldRange ara::rest::ogm::Object::GetFields() const
<b>Function param:</b>	None
<b>Return value:</b>	an iterator range of fields
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Returns a range of fields.

**Table 8.118: ara::rest::ogm::Object::GetFields**

[SWS\_REST\_02114] **ara::rest::ogm::Object::GetFields** [Table 8.118 describes the interface `ara::rest::ogm::Object::GetFields`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.15 HasField

<b>Service name:</b>	ara::rest::ogm::Object::HasField	
<b>Type:</b>	Member function	
<b>Syntax:</b>	bool ara::rest::ogm::Object::HasField(StringView name) const	
<b>Function param:</b>	name	of the field to search for
<b>Return value:</b>	true if a field of the given name exists	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/ogm/object.h	
<b>Class:</b>	ara::rest::ogm::Object	
<b>Description:</b>	Checks whether a field of a given name exists.	

**Table 8.119: ara::rest::ogm::Object::HasField**

[SWS\_REST\_02115] **ara::rest::ogm::Object::HasField** [Table 8.119 describes the interface `ara::rest::ogm::Object::HasField`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.16 Find

<b>Service name:</b>	ara::rest::ogm::Object::Find	
<b>Type:</b>	Member function	
<b>Syntax:</b>	Iterator ara::rest::ogm::Object::Find(StringView name)	
<b>Function param:</b>	name	field name to look up
<b>Return value:</b>	an iterator pointing to the position of the element.	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/ogm/object.h	
<b>Class:</b>	ara::rest::ogm::Object	
<b>Description:</b>	Searches for a field of the given name. If the given field name is not found, the return value will be equal to <code>GetFields().end()</code> .	

**Table 8.120: ara::rest::ogm::Object::Find**

[SWS\_REST\_02116] **ara::rest::ogm::Object::Find** [Table 8.120 describes the interface `ara::rest::ogm::Object::Find`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.17 Find

<b>Service name:</b>	ara::rest::ogm::Object::Find	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ConstIterator ara::rest::ogm::Object::Find(StringView name) const	
<b>Function param:</b>	name	field name to look up
<b>Return value:</b>	an iterator pointing to the position of the element.	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/ogm/object.h	
<b>Class:</b>	ara::rest::ogm::Object	
<b>Description:</b>	Searches for a field of the given name. If the given field name is not found, the return value will be equal to GetFields().end().	

**Table 8.121: ara::rest::ogm::Object::Find**

[SWS\_REST\_02117] **ara::rest::ogm::Object::Find** [Table 8.121 describes the interface `ara::rest::ogm::Object::Find`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.18 Insert

<b>Service name:</b>	ara::rest::ogm::Object::Insert	
<b>Type:</b>	Member function	
<b>Syntax:</b>	bool ara::rest::ogm::Object::Insert(Pointer< Field > &&f)	
<b>Function param:</b>	f	field to insert
<b>Return value:</b>	true if insertion was performed.	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/object.h	
<b>Class:</b>	ara::rest::ogm::Object	
<b>Description:</b>	Inserts a field into the object. If a field of the same name already exists, no insertion is performed. In this case the passed pointer to Field is not invalidated.	

**Table 8.122: ara::rest::ogm::Object::Insert**

[SWS\_REST\_02118] **ara::rest::ogm::Object::Insert** [Table 8.122 describes the interface `ara::rest::ogm::Object::Insert`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.19 Remove

<b>Service name:</b>	ara::rest::ogm::Object::Remove	
<b>Type:</b>	Member function	
<b>Syntax:</b>	Iterator ara::rest::ogm::Object::Remove(Iterator iter)	
<b>Function param:</b>	iter	an iterator pointing to an element.
<b>Return value:</b>	an iterator pointing to the element following the one just removed.	



<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Removes value from the set. Removes an element from the set. Removal invalidates all iterators referencing the respective element.

**Table 8.123: ara::rest::ogm::Object::Remove**

[SWS\_REST\_02119] **ara::rest::ogm::Object::Remove** [Table 8.123 describes the interface `ara::rest::ogm::Object::Remove`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.20 Release

<b>Service name:</b>	ara::rest::ogm::Object::Release	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>std::pair&lt;Iterator, Pointer&lt;Field&gt; &gt;</code> <code>ara::rest::ogm::Object::Release(Iterator iter)</code>	
<b>Function param:</b>	iter	an iterator pointing to the element to be removed
<b>Return value:</b>	a pair of the iterator pointing to the element following the one just deleted and a pointer to the element	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/object.h	
<b>Class:</b>	ara::rest::ogm::Object	
<b>Description:</b>	Similar to Remove but does not destroy the removed element. Instead of destroying the removed element, ownership is passed back to the user	

**Table 8.124: ara::rest::ogm::Object::Release**

[SWS\_REST\_02120] **ara::rest::ogm::Object::Release** [Table 8.124 describes the interface `ara::rest::ogm::Object::Release`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.21 Replace

<b>Service name:</b>	ara::rest::ogm::Object::Replace	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>Pointer&lt;Field&gt;</code> <code>ara::rest::ogm::Object::Replace(Iterator iter, Pointer&lt;Field&gt; &amp;&amp;field)</code>	
<b>Function param:</b>	iter	an iterator pointing to the element to be replaced
<b>Function param:</b>	field	Field to replace the current value
<b>Return value:</b>	a Pointer to the old element	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/object.h	
<b>Class:</b>	ara::rest::ogm::Object	

<b>Description:</b>	Replaces an element by a new one without the destroying the old one. Replaces a field without destroying it. Instead the replaced element is returned. Effectively, ownership of the old element is passed back to the user.
---------------------	--

**Table 8.125: ara::rest::ogm::Object::Replace**

[SWS\_REST\_02121] **ara::rest::ogm::Object::Replace** [Table 8.125 describes the interface `ara::rest::ogm::Object::Replace`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.22 Clear

<b>Service name:</b>	ara::rest::ogm::Object::Clear
<b>Type:</b>	Member function
<b>Syntax:</b>	void ara::rest::ogm::Object::Clear()
<b>Function param:</b>	None
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Removes all elements.

**Table 8.126: ara::rest::ogm::Object::Clear**

[SWS\_REST\_02122] **ara::rest::ogm::Object::Clear** [Table 8.126 describes the interface `ara::rest::ogm::Object::Clear`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.23 Make

<b>Service name:</b>	ara::rest::ogm::Object::Make
<b>Type:</b>	Member function
<b>Syntax:</b>	template <typename... Ts> static Pointer<SelfType> ara::rest::ogm::Object::Make(Ts &&...ts)
<b>Function param:</b>	ts constructor arguments forwarded to the constructor of this type
<b>Return value:</b>	a pointer to a node of type SelfType
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/object.h
<b>Class:</b>	ara::rest::ogm::Object
<b>Description:</b>	Creates a node of type SelfType.

**Table 8.127: ara::rest::ogm::Object::Make**

[SWS\_REST\_02123] `ara::rest::ogm::Object::Make` [Table 8.127 describes the interface `ara::rest::ogm::Object::Make`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.24 Make

<b>Service name:</b>	<code>ara::rest::ogm::Object::Make</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>template &lt;typename... Ts&gt; static Pointer&lt;SelfType&gt; ara::rest::ogm::Object::Make(Allocator *alloc, Ts &amp;&amp;...ts)</pre>	
<b>Function param:</b>	<code>alloc</code>	an allocator to use to construct this node
<b>Function param:</b>	<code>ts</code>	constructor arguments forwarded to the constructor of this type
<b>Return value:</b>	a pointer to a node of type <code>SelfType</code>	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	<code>ara/rest/ogm/object.h</code>	
<b>Class:</b>	<code>ara::rest::ogm::Object</code>	
<b>Description:</b>	Creates a node of type <code>SelfType</code> .	

**Table 8.128: `ara::rest::ogm::Object::Make`**

[SWS\_REST\_02124] `ara::rest::ogm::Object::Make` [Table 8.128 describes the interface `ara::rest::ogm::Object::Make`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.25 Object

<b>Service name:</b>	<code>ara::rest::ogm::Object::Object</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>template &lt;typename... Ts&gt; ara::rest::ogm::Object::Object(Pointer&lt; Ts &gt; &amp;&amp;...fields)</pre>	
<b>Function param:</b>	<code>fields</code>	Fields to be attached to this object
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	<code>ara/rest/ogm/object.h</code>	
<b>Class:</b>	<code>ara::rest::ogm::Object</code>	
<b>Description:</b>	Constructs an <code>Object</code> .	

**Table 8.129: `ara::rest::ogm::Object::Object`**

[SWS\_REST\_02125] `ara::rest::ogm::Object::Object` [Table 8.129 describes the interface `ara::rest::ogm::Object::Object`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.12.26 Object

<b>Service name:</b>	ara::rest::ogm::Object::Object	
<b>Type:</b>	Member function	
<b>Syntax:</b>	template <typename... Ts> ara::rest::ogm::Object::Object (Allocator *alloc, Pointer< Ts > &&...fields)	
<b>Function param:</b>	alloc	an allocator
<b>Function param:</b>	fields	Fields to be attached to this object
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/object.h	
<b>Class:</b>	ara::rest::ogm::Object	
<b>Description:</b>	Constructs an Object.	

**Table 8.130: ara::rest::ogm::Object::Object**

[SWS\_REST\_02126] `ara::rest::ogm::Object::Object` [Table 8.130 describes the interface `ara::rest::ogm::Object::Object`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

## 8.13 ara::rest::ogm::Real

[SWS\_REST\_02127] [`ara::rest::ogm::Real` class shall be declared in the `ara/rest/ogm/real.h` header file:

```
1     class ara::rest::ogm::Real : public ara::rest::ogm::Value;
```

]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.13.1 SelfType

<b>Name:</b>	SelfType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Real::SelfType = Real
<b>Header file:</b>	ara/rest/ogm/real.h
<b>Class:</b>	ara::rest::ogm::Real
<b>Description:</b>	Its own type.

**Table 8.131: ara::rest::ogm::Real::SelfType**

[SWS\_REST\_02128] `SelfType` [Table 8.131 describes the type alias `ara::rest::ogm::Real::SelfType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.13.2 ParentType

<b>Name:</b>	ParentType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Real::ParentType = Value
<b>Header file:</b>	ara/rest/ogm/real.h
<b>Class:</b>	ara::rest::ogm::Real
<b>Description:</b>	Type of its parent in the OGM type hierarchy.

**Table 8.132: ara::rest::ogm::Real::ParentType**

[SWS\_REST\_02129] **ParentType** [Table 8.132 describes the type alias `ara::rest::ogm::Real::ParentType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.13.3 ValueType

<b>Name:</b>	ValueType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Real::ValueType = long double
<b>Header file:</b>	ara/rest/ogm/real.h
<b>Class:</b>	ara::rest::ogm::Real
<b>Description:</b>	Type of its corresponding C++ data type.

**Table 8.133: ara::rest::ogm::Real::ValueType**

[SWS\_REST\_02130] **ValueType** [Table 8.133 describes the type alias `ara::rest::ogm::Real::ValueType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.13.4 GetParent

<b>Service name:</b>	ara::rest::ogm::Real::GetParent
<b>Type:</b>	Member function
<b>Syntax:</b>	Node* ara::rest::ogm::Real::GetParent()
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to its parent node
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/real.h
<b>Class:</b>	ara::rest::ogm::Real
<b>Description:</b>	Returns a (strongly-typed) pointer to its parent node.

**Table 8.134: ara::rest::ogm::Real::GetParent**

[SWS\_REST\_02131] **ara::rest::ogm::Real::GetParent** [Table 8.134 describes the interface `ara::rest::ogm::Real::GetParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.13.5 GetParent

<b>Service name:</b>	ara::rest::ogm::Real::GetParent
<b>Type:</b>	Member function
<b>Syntax:</b>	const Node* ara::rest::ogm::Real::GetParent() const
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to its parent node
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/real.h
<b>Class:</b>	ara::rest::ogm::Real
<b>Description:</b>	Returns a (strongly-typed) pointer to its parent node.

**Table 8.135: ara::rest::ogm::Real::GetParent**

[SWS\_REST\_02132] **ara::rest::ogm::Real::GetParent** [Table 8.135 describes the interface `ara::rest::ogm::Real::GetParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.13.6 HasParent

<b>Service name:</b>	ara::rest::ogm::Real::HasParent
<b>Type:</b>	Member function
<b>Syntax:</b>	bool ara::rest::ogm::Real::HasParent() const
<b>Function param:</b>	None
<b>Return value:</b>	true if this node has a structural parent
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/real.h
<b>Class:</b>	ara::rest::ogm::Real
<b>Description:</b>	Denotes whether this node has a structural parent.

**Table 8.136: ara::rest::ogm::Real::HasParent**

[SWS\_REST\_02133] **ara::rest::ogm::Real::HasParent** [Table 8.136 describes the interface `ara::rest::ogm::Real::HasParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.13.7 GetValue

<b>Service name:</b>	ara::rest::ogm::Real::GetValue
<b>Type:</b>	Member function
<b>Syntax:</b>	ValueType ara::rest::ogm::Real::GetValue() const
<b>Function param:</b>	None
<b>Return value:</b>	a value of type ValueType
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/real.h
<b>Class:</b>	ara::rest::ogm::Real
<b>Description:</b>	Returns its value as a C++ data type.

**Table 8.137: ara::rest::ogm::Real::GetValue**

[SWS\_REST\_02134] **ara::rest::ogm::Real::GetValue** [Table 8.137 describes the interface `ara::rest::ogm::Real::GetValue`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.13.8 SetValue

<b>Service name:</b>	ara::rest::ogm::Real::SetValue	
<b>Type:</b>	Member function	
<b>Syntax:</b>	void ara::rest::ogm::Real::SetValue(ValueType v)	
<b>Function param:</b>	v	a value
<b>Return value:</b>	None	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/ogm/real.h	
<b>Class:</b>	ara::rest::ogm::Real	
<b>Description:</b>	Sets the current value from a C++ data type.	

**Table 8.138: ara::rest::ogm::Real::SetValue**

[SWS\_REST\_02135] **ara::rest::ogm::Real::SetValue** [Table 8.138 describes the interface `ara::rest::ogm::Real::SetValue`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.13.9 Make

<b>Service name:</b>	ara::rest::ogm::Real::Make	
<b>Type:</b>	Member function	
<b>Syntax:</b>	template <typename... Ts> static Pointer<SelfType> ara::rest::ogm::Real::Make(Ts &&...ts)	
<b>Function param:</b>	ts	constructor arguments forwarded to the constructor of this type
<b>Return value:</b>	a pointer to a node of type SelfType	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/real.h	
<b>Class:</b>	ara::rest::ogm::Real	
<b>Description:</b>	Creates a node of type SelfType.	

**Table 8.139: ara::rest::ogm::Real::Make**

[SWS\_REST\_02136] **ara::rest::ogm::Real::Make** [Table 8.139 describes the interface `ara::rest::ogm::Real::Make`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.13.10 Make

<b>Service name:</b>	ara::rest::ogm::Real::Make	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>template &lt;typename... Ts&gt; static Pointer&lt;SelfType&gt; ara::rest::ogm::Real::Make(Allocator *alloc, Ts &amp;&amp;...ts)</pre>	
<b>Function param:</b>	alloc	an allocator to use to construct this node
<b>Function param:</b>	ts	constructor arguments forwarded to the constructor of this type
<b>Return value:</b>	a pointer to a node of type SelfType	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/real.h	
<b>Class:</b>	ara::rest::ogm::Real	
<b>Description:</b>	Creates a node of type SelfType.	

**Table 8.140: ara::rest::ogm::Real::Make**

[SWS\_REST\_02137] **ara::rest::ogm::Real::Make** [Table 8.140 describes the interface `ara::rest::ogm::Real::Make`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.13.11 Real

<b>Service name:</b>	ara::rest::ogm::Real::Real	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>ara::rest::ogm::Real::Real(ValueType value=ValueType{})</pre>	
<b>Function param:</b>	value	an initial value
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/real.h	
<b>Class:</b>	ara::rest::ogm::Real	
<b>Description:</b>	Connstructs an Real.	

**Table 8.141: ara::rest::ogm::Real::Real**

[SWS\_REST\_02138] **ara::rest::ogm::Real::Real** [Table 8.141 describes the interface `ara::rest::ogm::Real::Real`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

## 8.14 ara::rest::ogm::String

[SWS\_REST\_02139] [ara::rest::ogm::String class shall be declared in the ara/rest/ogm/string.h header file:

```
1     class ara::rest::ogm::String : public ara::rest::ogm::Value;
```



]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.14.1 SelfType

<b>Name:</b>	SelfType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::String::SelfType = String
<b>Header file:</b>	ara/rest/ogm/string.h
<b>Class:</b>	ara::rest::ogm::String
<b>Description:</b>	Its own type.

**Table 8.142: ara::rest::ogm::String::SelfType**

[SWS\_REST\_02140] **SelfType** [Table 8.142 describes the type alias `ara::rest::ogm::String::SelfType`.]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.14.2 ParentType

<b>Name:</b>	ParentType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::String::ParentType = Value
<b>Header file:</b>	ara/rest/ogm/string.h
<b>Class:</b>	ara::rest::ogm::String
<b>Description:</b>	Type of its parent in the OGM type hierarchy.

**Table 8.143: ara::rest::ogm::String::ParentType**

[SWS\_REST\_02141] **ParentType** [Table 8.143 describes the type alias `ara::rest::ogm::String::ParentType`.]([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.14.3 ValueType

<b>Name:</b>	ValueType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::String::ValueType = ara::core::StringView
<b>Header file:</b>	ara/rest/ogm/string.h
<b>Class:</b>	ara::rest::ogm::String
<b>Description:</b>	Type of its corresponding C++ data type.

**Table 8.144: ara::rest::ogm::String::ValueType**

[SWS\_REST\_02142] **ValueType** [Table 8.144 describes the type alias `ara::rest::ogm::String::ValueType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

#### 8.14.4 GetParent

<b>Service name:</b>	<code>ara::rest::ogm::String::GetParent</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Node* ara::rest::ogm::String::GetParent()</code>
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to its parent node
<b>Exceptions:</b>	<code>noexcept</code>
<b>Header file:</b>	<code>ara/rest/ogm/string.h</code>
<b>Class:</b>	<code>ara::rest::ogm::String</code>
<b>Description:</b>	Returns a (strongly-typed) pointer to its parent node.

**Table 8.145: `ara::rest::ogm::String::GetParent`**

[SWS\_REST\_02143] **`ara::rest::ogm::String::GetParent`** [Table 8.145 describes the interface `ara::rest::ogm::String::GetParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

#### 8.14.5 GetParent

<b>Service name:</b>	<code>ara::rest::ogm::String::GetParent</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>const Node* ara::rest::ogm::String::GetParent() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to its parent node
<b>Exceptions:</b>	<code>noexcept</code>
<b>Header file:</b>	<code>ara/rest/ogm/string.h</code>
<b>Class:</b>	<code>ara::rest::ogm::String</code>
<b>Description:</b>	Returns a (strongly-typed) pointer to its parent node.

**Table 8.146: `ara::rest::ogm::String::GetParent`**

[SWS\_REST\_02144] **`ara::rest::ogm::String::GetParent`** [Table 8.146 describes the interface `ara::rest::ogm::String::GetParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

#### 8.14.6 HasParent

<b>Service name:</b>	<code>ara::rest::ogm::String::HasParent</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>bool ara::rest::ogm::String::HasParent() const</code>

<b>Function param:</b>	None
<b>Return value:</b>	true if this node has a structural parent
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/string.h
<b>Class:</b>	ara::rest::ogm::String
<b>Description:</b>	Denotes whether this node has a structural parent.

**Table 8.147: ara::rest::ogm::String::HasParent**

[SWS\_REST\_02145] **ara::rest::ogm::String::HasParent** [Table 8.147 describes the interface `ara::rest::ogm::String::HasParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.14.7 GetValue

<b>Service name:</b>	ara::rest::ogm::String::GetValue
<b>Type:</b>	Member function
<b>Syntax:</b>	ValueType ara::rest::ogm::String::GetValue() const
<b>Function param:</b>	None
<b>Return value:</b>	a value of type ValueType
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/string.h
<b>Class:</b>	ara::rest::ogm::String
<b>Description:</b>	Returns its value as a C++ data type.

**Table 8.148: ara::rest::ogm::String::GetValue**

[SWS\_REST\_02146] **ara::rest::ogm::String::GetValue** [Table 8.148 describes the interface `ara::rest::ogm::String::GetValue`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.14.8 SetValue

<b>Service name:</b>	ara::rest::ogm::String::SetValue
<b>Type:</b>	Member function
<b>Syntax:</b>	void ara::rest::ogm::String::SetValue(const ValueType &v)
<b>Function param:</b>	v   a value
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/string.h
<b>Class:</b>	ara::rest::ogm::String
<b>Description:</b>	Sets the current value from a C++ data type.

**Table 8.149: ara::rest::ogm::String::SetValue**

[SWS\_REST\_02147] `ara::rest::ogm::String::SetValue` [Table 8.149 describes the interface `ara::rest::ogm::String::SetValue`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.14.9 Make

<b>Service name:</b>	<code>ara::rest::ogm::String::Make</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>template &lt;typename... Ts&gt; static Pointer&lt;SelfType&gt; ara::rest::ogm::String::Make(Ts &amp;&amp;...ts)</pre>	
<b>Function param:</b>	<code>ts</code>	constructor arguments forwarded to the constructor of this type
<b>Return value:</b>	a pointer to a node of type <code>SelfType</code>	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	<code>ara/rest/ogm/string.h</code>	
<b>Class:</b>	<code>ara::rest::ogm::String</code>	
<b>Description:</b>	Creates a node of type <code>SelfType</code> .	

**Table 8.150: `ara::rest::ogm::String::Make`**

[SWS\_REST\_02148] `ara::rest::ogm::String::Make` [Table 8.150 describes the interface `ara::rest::ogm::String::Make`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.14.10 Make

<b>Service name:</b>	<code>ara::rest::ogm::String::Make</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>template &lt;typename... Ts&gt; static Pointer&lt;SelfType&gt; ara::rest::ogm::String::Make(Allocator *alloc, Ts &amp;&amp;...ts)</pre>	
<b>Function param:</b>	<code>alloc</code>	an allocator to use to construct this node
<b>Function param:</b>	<code>ts</code>	constructor arguments forwarded to the constructor of this type
<b>Return value:</b>	a pointer to a node of type <code>SelfType</code>	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	<code>ara/rest/ogm/string.h</code>	
<b>Class:</b>	<code>ara::rest::ogm::String</code>	
<b>Description:</b>	Creates a node of type <code>SelfType</code> .	

**Table 8.151: `ara::rest::ogm::String::Make`**

[SWS\_REST\_02149] `ara::rest::ogm::String::Make` [Table 8.151 describes the interface `ara::rest::ogm::String::Make`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.14.11 String

<b>Service name:</b>	ara::rest::ogm::String::String	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ara::rest::ogm::String::String(ValueType value=ValueType{})	
<b>Function param:</b>	value	an intial value
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/string.h	
<b>Class:</b>	ara::rest::ogm::String	
<b>Description:</b>	Connstrcuts an String.	

**Table 8.152: ara::rest::ogm::String::String**

[SWS\_REST\_02150] **ara::rest::ogm::String::String** [Table 8.152 describes the interface [ara::rest::ogm::String::String](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.14.12 String

<b>Service name:</b>	ara::rest::ogm::String::String	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ara::rest::ogm::String::String(Allocator *alloc, ValueType value=ValueType{})	
<b>Function param:</b>	alloc	an allocator
<b>Function param:</b>	value	an intial value
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/string.h	
<b>Class:</b>	ara::rest::ogm::String	
<b>Description:</b>	Connstrcuts an String.	

**Table 8.153: ara::rest::ogm::String::String**

[SWS\_REST\_02151] **ara::rest::ogm::String::String** [Table 8.153 describes the interface [ara::rest::ogm::String::String](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

## 8.15 ara::rest::ogm::Value

[SWS\_REST\_02152] [ara::rest::ogm::Value class shall be declared in the ara/rest/ogm/value.h header file:

```
1     class ara::rest::ogm::Value : public ara::rest::ogm::Node;
```

] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.15.1 SelfType

<b>Name:</b>	SelfType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Value::SelfType = Value
<b>Header file:</b>	ara/rest/ogm/value.h
<b>Class:</b>	ara::rest::ogm::Value
<b>Description:</b>	Its own type.

**Table 8.154: ara::rest::ogm::Value::SelfType**

[SWS\_REST\_02153] **SelfType** [Table 8.154 describes the type alias `ara::rest::ogm::Value::SelfType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.15.2 ParentType

<b>Name:</b>	ParentType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ogm::Value::ParentType = Node
<b>Header file:</b>	ara/rest/ogm/value.h
<b>Class:</b>	ara::rest::ogm::Value
<b>Description:</b>	Type of its parent in the OGM type hierarchy.

**Table 8.155: ara::rest::ogm::Value::ParentType**

[SWS\_REST\_02154] **ParentType** [Table 8.155 describes the type alias `ara::rest::ogm::Value::ParentType`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#))

### 8.15.3 GetParent

<b>Service name:</b>	ara::rest::ogm::Value::GetParent
<b>Type:</b>	Member function
<b>Syntax:</b>	Node* ara::rest::ogm::Value::GetParent()
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to its parent node
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/value.h
<b>Class:</b>	ara::rest::ogm::Value
<b>Description:</b>	Returns a (strongly-typed) pointer to its parent node.

**Table 8.156: ara::rest::ogm::Value::GetParent**

[SWS\_REST\_02155] **ara::rest::ogm::Value::GetParent** [Table 8.156 describes the interface `ara::rest::ogm::Value::GetParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.15.4 GetParent

<b>Service name:</b>	ara::rest::ogm::Value::GetParent
<b>Type:</b>	Member function
<b>Syntax:</b>	const Node* ara::rest::ogm::Value::GetParent() const
<b>Function param:</b>	None
<b>Return value:</b>	a pointer to its parent node
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/value.h
<b>Class:</b>	ara::rest::ogm::Value
<b>Description:</b>	Returns a (strongly-typed) pointer to its parent node.

**Table 8.157: ara::rest::ogm::Value::GetParent**

[SWS\_REST\_02156] **ara::rest::ogm::Value::GetParent** [Table 8.157 describes the interface `ara::rest::ogm::Value::GetParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.15.5 HasParent

<b>Service name:</b>	ara::rest::ogm::Value::HasParent
<b>Type:</b>	Member function
<b>Syntax:</b>	bool ara::rest::ogm::Value::HasParent() const
<b>Function param:</b>	None
<b>Return value:</b>	true if this node has a structural parent
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/ogm/value.h
<b>Class:</b>	ara::rest::ogm::Value
<b>Description:</b>	Denotes whether this node has a structural parent.

**Table 8.158: ara::rest::ogm::Value::HasParent**

[SWS\_REST\_02157] **ara::rest::ogm::Value::HasParent** [Table 8.158 describes the interface `ara::rest::ogm::Value::HasParent`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.15.6 Value

<b>Service name:</b>	ara::rest::ogm::Value::Value
<b>Type:</b>	Member function
<b>Syntax:</b>	ara::rest::ogm::Value::Value()
<b>Function param:</b>	None
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/ogm/value.h
<b>Class:</b>	ara::rest::ogm::Value
<b>Description:</b>	Constructs a node. Inaccessible to the user

**Table 8.159: ara::rest::ogm::Value::Value**

[SWS\_REST\_02158] **ara::rest::ogm::Value::Value** [Table 8.159 describes the interface `ara::rest::ogm::Value::Value`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

## 8.16 ara::rest::Pattern

[SWS\_REST\_02159] [ara::rest::Pattern class shall be declared in the `ara/rest/routing.h` header file:

```
1     class ara::rest::Pattern;
```

] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00309](#))

### 8.16.1 Pattern

<b>Service name:</b>	ara::rest::Pattern::Pattern	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ara::rest::Pattern::Pattern(StringView pat)	
<b>Function param:</b>	pat	a pattern string
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/routing.h	
<b>Class:</b>	ara::rest::Pattern	
<b>Description:</b>	Constructs a Pattern.	

**Table 8.160: ara::rest::Pattern::Pattern**

[SWS\_REST\_02160] **ara::rest::Pattern::Pattern** [Table 8.160 describes the interface `ara::rest::Pattern::Pattern`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00309](#))

### 8.16.2 operator==

<b>Service name:</b>	ara::rest::Pattern::operator==	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	friend bool operator==(const Pattern &a, const Pattern &b)	
<b>Function param:</b>	a	a Pattern
<b>Function param:</b>	b	a Patern
<b>Return value:</b>	true if arguments compare equal	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/routing.h	
<b>Namespace:</b>	ara::rest::Pattern	



<b>Description:</b>	Compares patterns for equality.
---------------------	---------------------------------

**Table 8.161: ara::rest::Pattern::operator==**

[SWS\_REST\_02161] `ara::rest::Pattern::operator==` [Table 8.161 describes the interface `ara::rest::Pattern::operator==`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00309](#))

### 8.16.3 operator!=

<b>Service name:</b>	<code>ara::rest::Pattern::operator!=</code>	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<code>friend bool operator!=(const Pattern &amp;a, const Pattern &amp;b)</code>	
<b>Function param:</b>	<code>a</code>	<code>a Pattern</code>
<b>Function param:</b>	<code>b</code>	<code>a Patern</code>
<b>Return value:</b>	true if arguments compare unequal	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	<code>ara/rest/routing.h</code>	
<b>Namespace:</b>	<code>ara::rest::Pattern</code>	
<b>Description:</b>	Compares patterns for inequality.	

**Table 8.162: ara::rest::Pattern::operator!=**

[SWS\_REST\_02162] `ara::rest::Pattern::operator!=` [Table 8.162 describes the interface `ara::rest::Pattern::operator!=`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00309](#))

### 8.16.4 operator<

<b>Service name:</b>	<code>ara::rest::Pattern::operator&lt;</code>	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<code>friend bool operator&lt;(const Pattern &amp;a, const Pattern &amp;b)</code>	
<b>Function param:</b>	<code>a</code>	<code>a Pattern</code>
<b>Function param:</b>	<code>b</code>	<code>a Patern</code>
<b>Return value:</b>	true if 'a' is less-than 'b' accoring to Pattern order criteria.	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	<code>ara/rest/routing.h</code>	
<b>Namespace:</b>	<code>ara::rest::Pattern</code>	
<b>Description:</b>	Compares patterns for order.	

**Table 8.163: ara::rest::Pattern::operator<**

[SWS\_REST\_02163] `ara::rest::Pattern::operator<` [Table 8.163 describes the interface `ara::rest::Pattern::operator<`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00309](#))

## 8.17 ara::rest::ReplyHeader

[SWS\_REST\_02164] [ara::rest::ReplyHeader class shall be declared in the ara/rest/header.h header file:

```
1      class ara::rest::ReplyHeader;
```

](RS\_CM\_00300)

### 8.17.1 GetStatus

<b>Service name:</b>	ara::rest::ReplyHeader::GetStatus
<b>Type:</b>	Member function
<b>Syntax:</b>	int ara::rest::ReplyHeader::GetStatus() const
<b>Function param:</b>	None
<b>Return value:</b>	a status code
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::ReplyHeader
<b>Description:</b>	Returns the current message status code. Status codes are binding-specific

**Table 8.164: ara::rest::ReplyHeader::GetStatus**

[SWS\_REST\_02165] ara::rest::ReplyHeader::GetStatus [Table 8.164 describes the interface ara::rest::ReplyHeader::GetStatus.](RS\_CM\_00300)

### 8.17.2 SetStatus

<b>Service name:</b>	ara::rest::ReplyHeader::SetStatus
<b>Type:</b>	Member function
<b>Syntax:</b>	void ara::rest::ReplyHeader::SetStatus(int code) const
<b>Function param:</b>	code   an integral status code
<b>Return value:</b>	None
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::ReplyHeader
<b>Description:</b>	Sets a message status code. Status codes are binding-specific

**Table 8.165: ara::rest::ReplyHeader::SetStatus**

[SWS\_REST\_02166] ara::rest::ReplyHeader::SetStatus [Table 8.165 describes the interface ara::rest::ReplyHeader::SetStatus.](RS\_CM\_00300)

### 8.17.3 GetUri

<b>Service name:</b>	ara::rest::ReplyHeader::GetUri
<b>Type:</b>	Member function
<b>Syntax:</b>	const Uri& ara::rest::ReplyHeader::GetUri() const
<b>Function param:</b>	None
<b>Return value:</b>	a Uri
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::ReplyHeader
<b>Description:</b>	Returns a Uri. It is binding-specific how Uri map to the transport protocol format.

**Table 8.166: ara::rest::ReplyHeader::GetUri**

[SWS\_REST\_02167] [ara::rest::ReplyHeader::GetUri](#) [Table 8.166 describes the interface [ara::rest::ReplyHeader::GetUri](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

#### 8.17.4 SetUri

<b>Service name:</b>	ara::rest::ReplyHeader::SetUri
<b>Type:</b>	Member function
<b>Syntax:</b>	void ara::rest::ReplyHeader::SetUri(const Uri &uri)
<b>Function param:</b>	uri   a Uri
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::ReplyHeader
<b>Description:</b>	Allows to set a Uri. It is binding-specific how Uri map to the transport protocol format.

**Table 8.167: ara::rest::ReplyHeader::SetUri**

[SWS\_REST\_02168] [ara::rest::ReplyHeader::SetUri](#) [Table 8.167 describes the interface [ara::rest::ReplyHeader::SetUri](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

#### 8.17.5 HasField

<b>Service name:</b>	ara::rest::ReplyHeader::HasField
<b>Type:</b>	Member function
<b>Syntax:</b>	bool ara::rest::ReplyHeader::HasField(const StringView &key) const
<b>Function param:</b>	key   key of the field.
<b>Return value:</b>	true if the field exists
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::ReplyHeader
<b>Description:</b>	Denotes whether a field exists

**Table 8.168: ara::rest::ReplyHeader::HasField**

[SWS\_REST\_02489] **ara::rest::ReplyHeader::HasField** [Table 8.168 describes the interface [ara::rest::ReplyHeader::HasField.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.17.6 InsertField

<b>Service name:</b>	ara::rest::ReplyHeader::InsertField	
<b>Type:</b>	Member function	
<b>Syntax:</b>	bool ara::rest::ReplyHeader::InsertField(const StringView &key, const StringView &value)	
<b>Function param:</b>	key	key of the field.
<b>Function param:</b>	value	value of the field.
<b>Return value:</b>	true if a new field has been inserted	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/header.h	
<b>Class:</b>	ara::rest::ReplyHeader	
<b>Description:</b>	Inserts a field if it does not exist	

**Table 8.169: ara::rest::ReplyHeader::InsertField**

[SWS\_REST\_02490] **ara::rest::ReplyHeader::InsertField** [Table 8.169 describes the interface [ara::rest::ReplyHeader::InsertField.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.17.7 EraseField

<b>Service name:</b>	ara::rest::ReplyHeader::EraseField	
<b>Type:</b>	Member function	
<b>Syntax:</b>	bool ara::rest::ReplyHeader::EraseField(const StringView &key) noexcept	
<b>Function param:</b>	key	key of the field to be erased.
<b>Return value:</b>	true if field has been erased otherwise false	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/header.h	
<b>Class:</b>	ara::rest::ReplyHeader	
<b>Description:</b>	Erases a field from the header	

**Table 8.170: ara::rest::ReplyHeader::EraseField**

[SWS\_REST\_02492] **ara::rest::ReplyHeader::EraseField** [Table 8.170 describes the interface [ara::rest::ReplyHeader::EraseField.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.17.8 GetField

<b>Service name:</b>	ara::rest::ReplyHeader::GetField	
<b>Type:</b>	Member function	
<b>Syntax:</b>	StringView ara::rest::ReplyHeader::GetField(const StringView &key)	
<b>Function param:</b>	key	key of the field to be accessed.
<b>Return value:</b>	StringView to the value of the field	
<b>Exceptions:</b>	std::invalid_argument if key does not exist	
<b>Header file:</b>	ara/rest/header.h	
<b>Class:</b>	ara::rest::ReplyHeader	
<b>Description:</b>	Accesses a field value	

**Table 8.171: ara::rest::ReplyHeader::GetField**

[SWS\_REST\_02493] **ara::rest::ReplyHeader::GetField** [Table 8.171 describes the interface [ara::rest::ReplyHeader::GetField.](#)]([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.17.9 SetField

<b>Service name:</b>	ara::rest::ReplyHeader::SetField	
<b>Type:</b>	Member function	
<b>Syntax:</b>	void ara::rest::ReplyHeader::SetField(const StringView &key, const StringView &value) noexcept	
<b>Function param:</b>	key	key of the field to be set.
<b>Function param:</b>	value	value of the field to be set.
<b>Return value:</b>	void	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/header.h	
<b>Class:</b>	ara::rest::ReplyHeader	
<b>Description:</b>	Sets a fields value. If field does not exist, it is inserted.	

**Table 8.172: ara::rest::ReplyHeader::SetField**

[SWS\_REST\_02494] **ara::rest::ReplyHeader::SetField** [Table 8.172 describes the interface [ara::rest::ReplyHeader::SetField.](#)]([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.17.10 NumFields

<b>Service name:</b>	ara::rest::ReplyHeader::NumFields	
<b>Type:</b>	Member function	
<b>Syntax:</b>	std::size_t ara::rest::ReplyHeader::NumFields() const noexcept	
<b>Return value:</b>	the number of fields.	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/header.h	
<b>Class:</b>	ara::rest::ReplyHeader	

<b>Description:</b>	Returns the number of fields in this header.
---------------------	--

**Table 8.173: ara::rest::ReplyHeader::NumFields**

[SWS\_REST\_02496] **ara::rest::ReplyHeader::NumFields** [Table 8.173 describes the interface `ara::rest::ReplyHeader::NumFields`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.17.11 ClearFields

<b>Service name:</b>	ara::rest::ReplyHeader::ClearFields
<b>Type:</b>	Member function
<b>Syntax:</b>	void ara::rest::ReplyHeader::ClearFields() noexcept
<b>Return value:</b>	void
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::ReplyHeader
<b>Description:</b>	Erases all fields in this header.

**Table 8.174: ara::rest::ReplyHeader::ClearFields**

[SWS\_REST\_02497] **ara::rest::ReplyHeader::ClearFields** [Table 8.174 describes the interface `ara::rest::ReplyHeader::ClearFields`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.17.12 FieldIteratorRange

<b>Name:</b>	FieldIteratorRange
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::ReplyHeader::FieldIteratorRange = ara::rest::IteratorRange<unspecified_iterator_type>
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::ReplyHeader
<b>Description:</b>	! Iterator range of header fields

**Table 8.175: ara::rest::ReplyHeader::FieldIteratorRange**

[SWS\_REST\_02515] **FieldIteratorRange** [Table 8.175 describes the type alias `ara::rest::ReplyHeader::FieldIteratorRange`.] ([RS\\_CM\\_00300](#))

### 8.17.13 ConstFieldIteratorRange

<b>Name:</b>	ConstFieldIteratorRange
<b>Type:</b>	Member type alias

<b>Syntax:</b>	<code>using ara::rest::ReplyHeader::ConstFieldIteratorRange = ara::rest::IteratorRange&lt;unspecified_iterator_type&gt;</code>
<b>Header file:</b>	<code>ara/rest/header.h</code>
<b>Class:</b>	<code>ara::rest::ReplyHeader</code>
<b>Description:</b>	! Const iterator range of header fields

**Table 8.176: `ara::rest::ReplyHeader::ConstFieldIteratorRange`**

[SWS\_REST\_02516] **`ConstFieldIteratorRange`** [Table 8.176 describes the type alias `ara::rest::ReplyHeader::ConstFieldIteratorRange`.] ([RS\\_CM\\_00300](#))

### 8.17.14 FindField

<b>Service name:</b>	<code>ara::rest::ReplyHeader::FindField</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>FieldIteratorRange::Iterator ara::rest::ReplyHeader::FindField(StringView key) noexcept</code>	
<b>Function param:</b>	<code>key</code>	key of the field to be found.
<b>Return value:</b>	an iterator to field	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	<code>ara/rest/header.h</code>	
<b>Class:</b>	<code>ara::rest::ReplyHeader</code>	
<b>Description:</b>	Returns an iterator to a field.	

**Table 8.177: `ara::rest::ReplyHeader::FindField`**

[SWS\_REST\_02517] **`ara::rest::ReplyHeader::FindField`** [Table 8.177 describes the interface `ara::rest::ReplyHeader::FindField`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

<b>Service name:</b>	<code>ara::rest::ReplyHeader::FindField</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>ConstFieldIteratorRange::Iterator ara::rest::ReplyHeader::FindField(StringView key) const noexcept</code>	
<b>Function param:</b>	<code>key</code>	key of the field to be found.
<b>Return value:</b>	an iterator to field	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	<code>ara/rest/header.h</code>	
<b>Class:</b>	<code>ara::rest::ReplyHeader</code>	
<b>Description:</b>	Returns an iterator to a field.	

**Table 8.178: `ara::rest::ReplyHeader::FindField const`**

[SWS\_REST\_02518] **`ara::rest::ReplyHeader::FindField`** [Table 8.178 describes the interface `ara::rest::ReplyHeader::FindField`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.17.15 GetFields

<b>Service name:</b>	ara::rest::ReplyHeader::GetFields
<b>Type:</b>	Member function
<b>Syntax:</b>	FieldIteratorRange::Iterator ara::rest::ReplyHeader::GetFields() noexcept
<b>Return value:</b>	an IteratorRange of header fields
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::ReplyHeader
<b>Description:</b>	Returns a range of header fields.

**Table 8.179: ara::rest::ReplyHeader::GetFields**

[SWS\_REST\_02519] **ara::rest::ReplyHeader::GetFields** [Table 8.179 describes the interface `ara::rest::ReplyHeader::GetFields`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

<b>Service name:</b>	ara::rest::ReplyHeader::GetFields
<b>Type:</b>	Member function
<b>Syntax:</b>	ConstFieldIteratorRange::Iterator ara::rest::ReplyHeader::GetFields() const noexcept
<b>Return value:</b>	an IteratorRange of header fields
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::ReplyHeader
<b>Description:</b>	Returns a range of header fields.

**Table 8.180: ara::rest::ReplyHeader::GetFields const**

[SWS\_REST\_02520] **ara::rest::ReplyHeader::GetFields** [Table 8.180 describes the interface `ara::rest::ReplyHeader::GetFields`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

## 8.18 ara::rest::Reply

[SWS\_REST\_02169] [ara::rest::Reply class shall be declared in the `ara/rest/client.h` header file:

```
1     class ara::rest::Reply;
```

] ([RS\\_CM\\_00300](#))

### 8.18.1 Reply

<b>Service name:</b>	ara::rest::Reply::Reply
<b>Type:</b>	Member function



<b>Syntax:</b>	<code>ara::rest::Reply::Reply(const Reply &amp;)=delete</code>
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/client.h</code>
<b>Class:</b>	<code>ara::rest::Reply</code>
<b>Description:</b>	Non-copyable.

**Table 8.181: `ara::rest::Reply::Reply`**

[SWS\_REST\_02170] `ara::rest::Reply::Reply` [Table 8.181 describes the interface `ara::rest::Reply::Reply`.] ([RS\\_CM\\_00300](#))

### 8.18.2 operator=

<b>Service name:</b>	<code>ara::rest::Reply::operator=</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Reply&amp; ara::rest::Reply::operator=(const Reply &amp;)=delete</code>
<b>Return value:</b>	a value of type <code>Reply &amp;</code>
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/client.h</code>
<b>Class:</b>	<code>ara::rest::Reply</code>
<b>Description:</b>	Non-copy-assignable.

**Table 8.182: `ara::rest::Reply::operator=`**

[SWS\_REST\_02171] `ara::rest::Reply::operator=` [Table 8.182 describes the interface `ara::rest::Reply::operator=`.] ([RS\\_CM\\_00300](#))

### 8.18.3 GetHeader

<b>Service name:</b>	<code>ara::rest::Reply::GetHeader</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>ReplyHeader const&amp; ara::rest::Reply::GetHeader() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a reference to a <code>ReplyHeader</code>
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/client.h</code>
<b>Class:</b>	<code>ara::rest::Reply</code>
<b>Description:</b>	Obtains the message header. Requests the message header from the endpoint. Accessing the message header is always synchronous.

**Table 8.183: `ara::rest::Reply::GetHeader`**

[SWS\_REST\_02172] `ara::rest::Reply::GetHeader` [Table 8.183 describes the interface `ara::rest::Reply::GetHeader`.] ([RS\\_CM\\_00300](#))

### 8.18.4 GetObject

<b>Service name:</b>	ara::rest::Reply::GetObject
<b>Type:</b>	Member function
<b>Syntax:</b>	Task<ogm::Object const&> ara::rest::Reply::GetObject () const
<b>Function param:</b>	None
<b>Return value:</b>	returns a task waiting for the message payload to be received.
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/client.h
<b>Class:</b>	ara::rest::Reply
<b>Description:</b>	Obtains the reply message payload.

**Table 8.184: ara::rest::Reply::GetObject**

[SWS\_REST\_02173] **ara::rest::Reply::GetObject** [Table 8.184 describes the interface [ara::rest::Reply::GetObject.](#)] ([RS\\_CM\\_00300](#))

### 8.18.5 ReleaseObject

<b>Service name:</b>	ara::rest::Reply::ReleaseObject
<b>Type:</b>	Member function
<b>Syntax:</b>	Task<Pointer<ogm::Object> > ara::rest::Reply::ReleaseObject ()
<b>Function param:</b>	None
<b>Return value:</b>	returns a task waiting for the message payload to be received.
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/client.h
<b>Class:</b>	ara::rest::Reply
<b>Description:</b>	Obtains the reply message payload.

**Table 8.185: ara::rest::Reply::ReleaseObject**

[SWS\_REST\_02174] **ara::rest::Reply::ReleaseObject** [Table 8.185 describes the interface [ara::rest::Reply::ReleaseObject.](#)] ([RS\\_CM\\_00300](#))

### 8.18.6 ReleaseBinary

<b>Service name:</b>	ara::rest::Reply::ReleaseBinary
<b>Type:</b>	Member function
<b>Syntax:</b>	Task<Pointer<ara::core::String> ara::rest::Reply::ReleaseBinary ()
<b>Function param:</b>	None
<b>Return value:</b>	returns a task waiting for the binary message payload to be received.
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/client.h
<b>Class:</b>	ara::rest::Reply
<b>Description:</b>	Obtains the reply binary payload.

**Table 8.186: ara::rest::Reply::ReleaseBinary**

[SWS\_REST\_02973] **ara::rest::Reply::ReleaseBinary** [Table 8.186 describes the interface `ara::rest::Reply::ReleaseBinary`.] ([RS\\_CM\\_00300](#))

## 8.19 ara::rest::RequestHeader

[SWS\_REST\_02175] [ `ara::rest::RequestHeader` class shall be declared in the `ara/rest/header.h` header file:

```
1     class ara::rest::RequestHeader;
```

] ([RS\\_CM\\_00300](#))

### 8.19.1 GetMethod

<b>Service name:</b>	ara::rest::RequestHeader::GetMethod
<b>Type:</b>	Member function
<b>Syntax:</b>	RequestMethod ara::rest::RequestHeader::GetMethod() const
<b>Function param:</b>	None
<b>Return value:</b>	a request method
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::RequestHeader
<b>Description:</b>	Returns the request method.

**Table 8.187: ara::rest::RequestHeader::GetMethod**

[SWS\_REST\_02176] **ara::rest::RequestHeader::GetMethod** [Table 8.187 describes the interface `ara::rest::RequestHeader::GetMethod`.] ([RS\\_CM\\_00300](#))

### 8.19.2 SetMethod

<b>Service name:</b>	ara::rest::RequestHeader::SetMethod	
<b>Type:</b>	Member function	
<b>Syntax:</b>	void ara::rest::RequestHeader::SetMethod(RequestMethod met)	
<b>Function param:</b>	met	a RequestMethod
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/header.h	
<b>Class:</b>	ara::rest::RequestHeader	

<b>Description:</b>	Allows to set the request method.
---------------------	-----------------------------------

**Table 8.188: ara::rest::RequestHeader::SetMethod**

[SWS\_REST\_02177] **ara::rest::RequestHeader::SetMethod** [Table 8.188 describes the interface `ara::rest::RequestHeader::SetMethod.`] ([RS\\_CM\\_00300](#))

### 8.19.3 GetUri

<b>Service name:</b>	ara::rest::RequestHeader::GetUri
<b>Type:</b>	Member function
<b>Syntax:</b>	const Uri& ara::rest::RequestHeader::GetUri() const
<b>Function param:</b>	None
<b>Return value:</b>	a Uri
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::RequestHeader
<b>Description:</b>	Returns a Uri.

**Table 8.189: ara::rest::RequestHeader::GetUri**

[SWS\_REST\_02178] **ara::rest::RequestHeader::GetUri** [Table 8.189 describes the interface `ara::rest::RequestHeader::GetUri.`] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.19.4 SetUri

<b>Service name:</b>	ara::rest::RequestHeader::SetUri
<b>Type:</b>	Member function
<b>Syntax:</b>	void ara::rest::RequestHeader::SetUri(const Uri &uri)
<b>Function param:</b>	uri   a Uri
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::RequestHeader
<b>Description:</b>	Allows to set a Uri.

**Table 8.190: ara::rest::RequestHeader::SetUri**

[SWS\_REST\_02179] **ara::rest::RequestHeader::SetUri** [Table 8.190 describes the interface `ara::rest::RequestHeader::SetUri.`] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.19.5 HasField

<b>Service name:</b>	ara::rest::RequestHeader::HasField	
<b>Type:</b>	Member function	
<b>Syntax:</b>	bool ara::rest::RequestHeader::HasField(const StringView &key) const	
<b>Function param:</b>	key	key of the field.
<b>Return value:</b>	true if the field exists	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/header.h	
<b>Class:</b>	ara::rest::RequestHeader	
<b>Description:</b>	Denotes whether a field exists	

**Table 8.191: ara::rest::RequestHeader::HasField**

[SWS\_REST\_02498] **ara::rest::RequestHeader::HasField** [Table 8.191 describes the interface [ara::rest::RequestHeader::HasField.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.19.6 InsertField

<b>Service name:</b>	ara::rest::RequestHeader::InsertField	
<b>Type:</b>	Member function	
<b>Syntax:</b>	bool ara::rest::RequestHeader::InsertField(const StringView &key, const Stringview &value)	
<b>Function param:</b>	key	key of the field.
<b>Function param:</b>	value	value of the field.
<b>Return value:</b>	true if a new field has been inserted	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/header.h	
<b>Class:</b>	ara::rest::RequestHeader	
<b>Description:</b>	Inserts a field if it does not exist	

**Table 8.192: ara::rest::RequestHeader::InsertField**

[SWS\_REST\_02499] **ara::rest::RequestHeader::InsertField** [Table 8.192 describes the interface [ara::rest::RequestHeader::InsertField.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.19.7 EraseField

<b>Service name:</b>	ara::rest::RequestHeader::EraseField	
<b>Type:</b>	Member function	
<b>Syntax:</b>	bool ara::rest::RequestHeader::EraseField(const StringView &key) noexcept	
<b>Function param:</b>	key	key of the field to be erased.
<b>Return value:</b>	true if field has been erased otherwise false	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/header.h	

<b>Class:</b>	ara::rest::RequestHeader
<b>Description:</b>	Erases a field from the header

**Table 8.193: ara::rest::RequestHeader::EraseField**

[SWS\_REST\_02501] **ara::rest::RequestHeader::EraseField** [Table 8.193 describes the interface [ara::rest::RequestHeader::EraseField.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.19.8 GetField

<b>Service name:</b>	ara::rest::RequestHeader::GetField	
<b>Type:</b>	Member function	
<b>Syntax:</b>	StringView ara::rest::RequestHeader::GetField(const StringView &key)	
<b>Function param:</b>	key	key of the field to be accessed.
<b>Return value:</b>	StringView to the value of the field	
<b>Exceptions:</b>	std::invalid_argument if key does not exist	
<b>Header file:</b>	ara/rest/header.h	
<b>Class:</b>	ara::rest::RequestHeader	
<b>Description:</b>	Accesses a field value	

**Table 8.194: ara::rest::RequestHeader::GetField**

[SWS\_REST\_02502] **ara::rest::RequestHeader::GetField** [Table 8.194 describes the interface [ara::rest::RequestHeader::GetField.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.19.9 SetField

<b>Service name:</b>	ara::rest::RequestHeader::SetField	
<b>Type:</b>	Member function	
<b>Syntax:</b>	void ara::rest::RequestHeader::SetField(const StringView &key, const StringView &value) noexcept	
<b>Function param:</b>	key	key of the field to be set.
<b>Function param:</b>	value	value of the field to be set.
<b>Return value:</b>	void	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/header.h	
<b>Class:</b>	ara::rest::RequestHeader	
<b>Description:</b>	Sets a fields value. If field does not exist, it is inserted.	

**Table 8.195: ara::rest::RequestHeader::SetField**

[SWS\_REST\_02503] **ara::rest::RequestHeader::SetField** [Table 8.195 describes the interface [ara::rest::RequestHeader::SetField.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.19.10 NumFields

<b>Service name:</b>	ara::rest::RequestHeader::NumFields
<b>Type:</b>	Member function
<b>Syntax:</b>	std::size_t ara::rest::RequestHeader::NumFields() const noexcept
<b>Return value:</b>	the number of fields.
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::RequestHeader
<b>Description:</b>	Returns the number of fields in this header.

**Table 8.196: ara::rest::RequestHeader::NumFields**

[SWS\_REST\_02505] **ara::rest::RequestHeader::NumFields** [Table 8.196 describes the interface `ara::rest::RequestHeader::NumFields.`] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.19.11 ClearFields

<b>Service name:</b>	ara::rest::RequestHeader::ClearFields
<b>Type:</b>	Member function
<b>Syntax:</b>	void ara::rest::RequestHeader::ClearFields() noexcept
<b>Return value:</b>	void
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::RequestHeader
<b>Description:</b>	Erases all fields in this header.

**Table 8.197: ara::rest::RequestHeader::ClearFields**

[SWS\_REST\_02506] **ara::rest::RequestHeader::ClearFields** [Table 8.197 describes the interface `ara::rest::RequestHeader::ClearFields.`] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.19.12 FieldIteratorRange

<b>Name:</b>	FieldIteratorRange
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::RequestHeader::FieldIteratorRange = ara::rest::IteratorRange<unspecified_iterator_type>
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::RequestHeader
<b>Description:</b>	! Iterator range of header fields

**Table 8.198: ara::rest::RequestHeader::FieldIteratorRange**

[SWS\_REST\_02528] **FieldIteratorRange** [Table 8.198 describes the type alias `ara::rest::RequestHeader::FieldIteratorRange`.] ([RS\\_CM\\_00300](#))

### 8.19.13 ConstFieldIteratorRange

<b>Name:</b>	ConstFieldIteratorRange
<b>Type:</b>	Member type alias
<b>Syntax:</b>	<code>using ara::rest::RequestHeader::ConstFieldIteratorRange = ara::rest::IteratorRange&lt;unspecified_iterator_type&gt;</code>
<b>Header file:</b>	<code>ara/rest/header.h</code>
<b>Class:</b>	<code>ara::rest::RequestHeader</code>
<b>Description:</b>	! Const iterator range of header fields

**Table 8.199: `ara::rest::RequestHeader::ConstFieldIteratorRange`**

[SWS\_REST\_02529] **ConstFieldIteratorRange** [Table 8.199 describes the type alias `ara::rest::RequestHeader::ConstFieldIteratorRange`.] ([RS\\_CM\\_00300](#))

### 8.19.14 FindField

<b>Service name:</b>	<code>ara::rest::RequestHeader::FindField</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>FieldIteratorRange::Iterator</code> <code>ara::rest::RequestHeader::FindField(const StringView &amp; key) noexcept</code>	
<b>Function param:</b>	<code>key</code>	key of the field to be found.
<b>Return value:</b>	an iterator to field	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	<code>ara/rest/header.h</code>	
<b>Class:</b>	<code>ara::rest::RequestHeader</code>	
<b>Description:</b>	Returns an iterator to a field.	

**Table 8.200: `ara::rest::RequestHeader::FindField`**

[SWS\_REST\_02511] **`ara::rest::RequestHeader::FindField`** [Table 8.200 describes the interface `ara::rest::RequestHeader::FindField`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

<b>Service name:</b>	<code>ara::rest::RequestHeader::FindField</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>ConstFieldIteratorRange::Iterator</code> <code>ara::rest::RequestHeader::FindField(const StringView &amp; key) const noexcept</code>	
<b>Function param:</b>	<code>key</code>	key of the field to be found.
<b>Return value:</b>	an iterator to field	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	<code>ara/rest/header.h</code>	
<b>Class:</b>	<code>ara::rest::RequestHeader</code>	



<b>Description:</b>	Returns an iterator to a field.
---------------------	---------------------------------

**Table 8.201: ara::rest::RequestHeader::FindField const**

[SWS\_REST\_02512] **ara::rest::RequestHeader::FindField** [Table 8.201 describes the interface `ara::rest::RequestHeader::FindField`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.19.15 GetFields

<b>Service name:</b>	ara::rest::RequestHeader::GetFields
<b>Type:</b>	Member function
<b>Syntax:</b>	FieldIteratorRange::Iterator ara::rest::RequestHeader::GetFields() noexcept
<b>Return value:</b>	an IteratorRange of header fields
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::RequestHeader
<b>Description:</b>	Returns a range of header fields.

**Table 8.202: ara::rest::RequestHeader::GetFields**

[SWS\_REST\_02513] **ara::rest::RequestHeader::GetFields** [Table 8.202 describes the interface `ara::rest::RequestHeader::GetFields`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

<b>Service name:</b>	ara::rest::RequestHeader::GetFields
<b>Type:</b>	Member function
<b>Syntax:</b>	ConstFieldIteratorRange::Iterator ara::rest::RequestHeader::GetFields() const noexcept
<b>Return value:</b>	an IteratorRange of header fields
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/header.h
<b>Class:</b>	ara::rest::RequestHeader
<b>Description:</b>	Returns a range of header fields.

**Table 8.203: ara::rest::RequestHeader::GetFields const**

[SWS\_REST\_02514] **ara::rest::RequestHeader::GetFields** [Table 8.203 describes the interface `ara::rest::RequestHeader::GetFields`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.19.16 GetStatus

<b>Service name:</b>	ara::rest::RequestHeader::GetStatus
<b>Type:</b>	Member function

<b>Syntax:</b>	<code>int ara::rest::RequestHeader::GetStatus()</code>
<b>Return value:</b>	a status code.
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/header.h</code>
<b>Class:</b>	<code>ara::rest::RequestHeader</code>
<b>Description:</b>	Returns the current message status code. The status codes are binding-specific.

**Table 8.204: `ara::rest::RequestHeader::GetStatus`**

[SWS\_REST\_02507] `ara::rest::RequestHeader::GetStatus` [Table 8.204 describes the interface `ara::rest::RequestHeader::GetStatus`. Status codes are issued by a server in response to a client’s request made to the server.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.19.17 SetStatus

<b>Service name:</b>	<code>ara::rest::RequestHeader::SetStatus</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>void ara::rest::RequestHeader::SetStatus(int code) const</code>	
<b>Function param:</b>	<code>code</code>	an integral status code
<b>Return value:</b>	None	
<b>Exceptions:</b>	<code>noexcept</code>	
<b>Header file:</b>	<code>ara/rest/header.h</code>	
<b>Class:</b>	<code>ara::rest::RequestHeader</code>	
<b>Description:</b>	Sets a message status code. Status codes are binding-specific	

**Table 8.205: `ara::rest::RequestHeader::SetStatus`**

[SWS\_REST\_02508] `ara::rest::RequestHeader::SetStatus` [Table 8.205 describes the interface `ara::rest::RequestHeader::SetStatus`.] ([RS\\_CM\\_00300](#))

## 8.20 `ara::rest::Request`

[SWS\_REST\_02180] [`ara::rest::Request` class shall be declared in the `ara/rest/client.h` header file:

```
1     class ara::rest::Request;
```

] ([RS\\_CM\\_00300](#))

### 8.20.1 Request

<b>Service name:</b>	<code>ara::rest::Request::Request</code>
----------------------	--

<b>Type:</b>	Member function
<b>Syntax:</b>	<code>ara::rest::Request::Request(const Request &amp;)=delete</code>
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/client.h</code>
<b>Class:</b>	<code>ara::rest::Request</code>
<b>Description:</b>	Non-copyable.

**Table 8.206: `ara::rest::Request::Request`**

[SWS\_REST\_02181] `ara::rest::Request::Request` [Table 8.206 describes the interface `ara::rest::Request::Request`.] (*RS\_CM\_00300*)

## 8.20.2 operator=

<b>Service name:</b>	<code>ara::rest::Request::operator=</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Request&amp; ara::rest::Request::operator=(const Request &amp;)=delete</code>
<b>Return value:</b>	a value of type <code>Request &amp;</code>
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/client.h</code>
<b>Class:</b>	<code>ara::rest::Request</code>
<b>Description:</b>	Non-copy-assignable.

**Table 8.207: `ara::rest::Request::operator=`**

[SWS\_REST\_02182] `ara::rest::Request::operator=` [Table 8.207 describes the interface `ara::rest::Request::operator=`.] (*RS\_CM\_00300*)

## 8.20.3 Request

<b>Service name:</b>	<code>ara::rest::Request::Request</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>ara::rest::Request::Request(RequestMethod met, const Uri &amp;uri)</code>	
<b>Function param:</b>	<code>met</code>	one of <code>RequestMethod</code>
<b>Function param:</b>	<code>uri</code>	a <code>Uri</code>
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	<code>ara/rest/client.h</code>	
<b>Class:</b>	<code>ara::rest::Request</code>	
<b>Description:</b>	Constructs a <code>Request</code> .	

**Table 8.208: `ara::rest::Request::Request`**

[SWS\_REST\_02183] `ara::rest::Request::Request` [Table 8.208 describes the interface `ara::rest::Request::Request`.] (*RS\_CM\_00300*)

## 8.20.4 Request

<b>Service name:</b>	<code>ara::rest::Request::Request</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>ara::rest::Request::Request(RequestMethod met, Uri &amp;&amp;uri)</code>	
<b>Function param:</b>	<code>met</code>	one of RequestMethod
<b>Function param:</b>	<code>uri</code>	a Uri
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	<code>ara/rest/client.h</code>	
<b>Class:</b>	<code>ara::rest::Request</code>	
<b>Description:</b>	Constructs a Request.	

Table 8.209: `ara::rest::Request::Request`

[SWS\_REST\_02184] `ara::rest::Request::Request` [Table 8.209 describes the interface `ara::rest::Request::Request`.] (*RS\_CM\_00300*)

## 8.20.5 Request

<b>Service name:</b>	<code>ara::rest::Request::Request</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>ara::rest::Request::Request(RequestMethod met, const Uri &amp;uri, const Pointer&lt; ogm::Object &gt; &amp;obj)</code>	
<b>Function param:</b>	<code>met</code>	one of RequestMethod
<b>Function param:</b>	<code>uri</code>	a Uri
<b>Function param:</b>	<code>obj</code>	data payload of request message
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	<code>ara/rest/client.h</code>	
<b>Class:</b>	<code>ara::rest::Request</code>	
<b>Description:</b>	Constructs a Request.	

Table 8.210: `ara::rest::Request::Request`

[SWS\_REST\_02185] `ara::rest::Request::Request` [Table 8.210 describes the interface `ara::rest::Request::Request`.] (*RS\_CM\_00300*)

## 8.20.6 Request

<b>Service name:</b>	<code>ara::rest::Request::Request</code>	
<b>Type:</b>	Member function	

<b>Syntax:</b>	ara::rest::Request::Request(RequestMethod met, const Uri &uri, Pointer< ogm::Object > &&obj)	
<b>Function param:</b>	met	one of RequestMethod
<b>Function param:</b>	uri	a Uri
<b>Function param:</b>	obj	data payload of request message
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/client.h	
<b>Class:</b>	ara::rest::Request	
<b>Description:</b>	Constructs a Request.	

**Table 8.211: ara::rest::Request::Request**

[SWS\_REST\_02186] **ara::rest::Request::Request** [Table 8.211 describes the interface `ara::rest::Request::Request`.] (*RS\_CM\_00300*)

## 8.20.7 Request

<b>Service name:</b>	ara::rest::Request::Request	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ara::rest::Request::Request(RequestMethod met, Uri &&uri, const Pointer< ogm::Object > &obj)	
<b>Function param:</b>	met	one of RequestMethod
<b>Function param:</b>	uri	a Uri
<b>Function param:</b>	obj	data payload of request message
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/client.h	
<b>Class:</b>	ara::rest::Request	
<b>Description:</b>	Constructs a Request.	

**Table 8.212: ara::rest::Request::Request**

[SWS\_REST\_02187] **ara::rest::Request::Request** [Table 8.212 describes the interface `ara::rest::Request::Request`.] (*RS\_CM\_00300*)

## 8.20.8 Request

<b>Service name:</b>	ara::rest::Request::Request	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ara::rest::Request::Request(RequestMethod met, Uri &&uri, Pointer< ogm::Object > &&obj)	
<b>Function param:</b>	met	one of RequestMethod
<b>Function param:</b>	uri	a Uri
<b>Function param:</b>	obj	data payload of request message
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/client.h	

<b>Class:</b>	ara::rest::Request
<b>Description:</b>	Constructs a Request.

**Table 8.213: ara::rest::Request::Request**

[SWS\_REST\_02188] **ara::rest::Request::Request** [Table 8.213 describes the interface [ara::rest::Request::Request.](#)] ([RS\\_CM\\_00300](#))

## 8.20.9 Request

<b>Service name:</b>	ara::rest::Request::Request	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ara::rest::Request::Request(RequestMethod met, const Uri &uri, Pointer< ara::core::String > &&bin)	
<b>Function param:</b>	met	one of RequestMethod
<b>Function param:</b>	uri	a Uri
<b>Function param:</b>	bin	binary data payload of request message
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/client.h	
<b>Class:</b>	ara::rest::Request	
<b>Description:</b>	Constructs a Request.	

**Table 8.214: ara::rest::Request::Request**

[SWS\_REST\_02989] **ara::rest::Request::Request** [Table 8.214 describes the interface [ara::rest::Request::Request.](#)] ([RS\\_CM\\_00300](#))

## 8.21 ara::rest::Router

[SWS\_REST\_02189] [ara::rest::Router class shall be declared in the ara/rest/routing.h header file:

```
1     class ara::rest::Router;
```

] ([RS\\_CM\\_00300](#))

### 8.21.1 RouteHandlerType

<b>Name:</b>	RouteHandlerType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::Router::RouteHandlerType = Route::Upshot(const ServerRequest&, ServerReply&, const Matches&)
<b>Header file:</b>	ara/rest/routing.h

<b>Class:</b>	ara::rest::Router
<b>Description:</b>	! User-define route handler function type

**Table 8.215: ara::rest::Router::RouteHandlerType**

[SWS\_REST\_02190] **RouteHandlerType** [Table 8.215 describes the type alias `ara::rest::Router::RouteHandlerType`.] ([RS\\_CM\\_00300](#))

## 8.21.2 RouteRange

<b>Name:</b>	RouteRange
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::Router::RouteRange = IteratorRange<unspecified_iterator_type>
<b>Header file:</b>	ara/rest/routing.h
<b>Class:</b>	ara::rest::Router
<b>Description:</b>	! Iterator range of routes

**Table 8.216: ara::rest::Router::RouteRange**

[SWS\_REST\_02191] **RouteRange** [Table 8.216 describes the type alias `ara::rest::Router::RouteRange`.] ([RS\\_CM\\_00300](#))

## 8.21.3 ConstRouteRange

<b>Name:</b>	ConstRouteRange
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::Router::ConstRouteRange = IteratorRange<unspecified_iterator_type>
<b>Header file:</b>	ara/rest/routing.h
<b>Class:</b>	ara::rest::Router
<b>Description:</b>	! Const iterator range of routes

**Table 8.217: ara::rest::Router::ConstRouteRange**

[SWS\_REST\_02192] **ConstRouteRange** [Table 8.217 describes the type alias `ara::rest::Router::ConstRouteRange`.] ([RS\\_CM\\_00300](#))

## 8.21.4 Router

<b>Service name:</b>	ara::rest::Router::Router
<b>Type:</b>	Member function
<b>Syntax:</b>	ara::rest::Router::Router(Allocator *alloc=GetDefaultAllocator())

<b>Function param:</b>	alloc	an allocator for all internal dynamic memory requirements
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/routing.h	
<b>Class:</b>	ara::rest::Router	
<b>Description:</b>	Constructs an empty Router.	

**Table 8.218: ara::rest::Router::Router**

[SWS\_REST\_02193] [ara::rest::Router::Router](#) [Table 8.218 describes the interface [ara::rest::Router::Router](#).] ([RS\\_CM\\_00300](#))

## 8.21.5 Router

<b>Service name:</b>	ara::rest::Router::Router	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>ara::rest::Router::Router(std::initializer_list&lt; Route &gt; routes, Allocator *alloc=GetDefaultAllocator())</pre>	
<b>Function param:</b>	routes	a list of routes
<b>Function param:</b>	alloc	an allocator for all internal dynamic memory requirements
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/routing.h	
<b>Class:</b>	ara::rest::Router	
<b>Description:</b>	Constructs a router from a given list of routes.	

**Table 8.219: ara::rest::Router::Router**

[SWS\_REST\_02194] [ara::rest::Router::Router](#) [Table 8.219 describes the interface [ara::rest::Router::Router](#).] ([RS\\_CM\\_00300](#))

## 8.21.6 operator()

<b>Service name:</b>	ara::rest::Router::operator()	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>void ara::rest::Router::operator()(const ServerRequest &amp;req, ServerReply &amp;rep) const</pre>	
<b>Function param:</b>	req	a request
<b>Function param:</b>	rep	a reply
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/routing.h	
<b>Class:</b>	ara::rest::Router	
<b>Description:</b>	Request handler function. This function serves as the user-defined request handler function passed to Server	



**Table 8.220: ara::rest::Router::operator()**

[SWS\_REST\_02195] **ara::rest::Router::operator()** [Table 8.220 describes the interface `ara::rest::Router::operator()`.] (*RS\_CM\_00300*)

### 8.21.7 InsertRoute

<b>Service name:</b>	ara::rest::Router::InsertRoute	
<b>Type:</b>	Member function	
<b>Syntax:</b>	Router& ara::rest::Router::InsertRoute(const Route &route)	
<b>Function param:</b>	route	a route
<b>Return value:</b>	a reference to this	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/routing.h	
<b>Class:</b>	ara::rest::Router	
<b>Description:</b>	Inserts a route into the set of potential matches. If a route already exists nothing is inserted.	

**Table 8.221: ara::rest::Router::InsertRoute**

[SWS\_REST\_02196] **ara::rest::Router::InsertRoute** [Table 8.221 describes the interface `ara::rest::Router::InsertRoute`.] (*RS\_CM\_00300*)

### 8.21.8 EmplaceRoute

<b>Service name:</b>	ara::rest::Router::EmplaceRoute	
<b>Type:</b>	Member function	
<b>Syntax:</b>	Router& ara::rest::Router::EmplaceRoute(RequestMethod met, Pattern pat, const Function< RouteHandlerType > &hnd)	
<b>Function param:</b>	met	a set of request methods
<b>Function param:</b>	pat	a URI Pattern
<b>Function param:</b>	hnd	a user-defined routing handler
<b>Return value:</b>	a reference to this	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/routing.h	
<b>Class:</b>	ara::rest::Router	
<b>Description:</b>	Constructs a route in-place Similar to Insert except that the route is constructed in-place. The given arguments are forwarded to the internal Route. If such a route already exists nothing is inserted.	

**Table 8.222: ara::rest::Router::EmplaceRoute**

[SWS\_REST\_02197] **ara::rest::Router::EmplaceRoute** [Table 8.222 describes the interface `ara::rest::Router::EmplaceRoute`.] (*RS\_CM\_00300*)

### 8.21.9 SetDefaultHandler

<b>Service name:</b>	ara::rest::Router::SetDefaultHandler	
<b>Type:</b>	Member function	
<b>Syntax:</b>	Router& ara::rest::Router::SetDefaultHandler(const Function< Server::RequestHandlerType > &hnd)	
<b>Function param:</b>	hnd	a user-defined request handler
<b>Return value:</b>	a reference to this.	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/routing.h	
<b>Class:</b>	ara::rest::Router	
<b>Description:</b>	Enables a user to set a default request handler The given handler is called if none of the routes matched of it at least once of the routes called Route::Default().	

**Table 8.223: ara::rest::Router::SetDefaultHandler**

[SWS\_REST\_02198] [ara::rest::Router::SetDefaultHandler](#) [Table 8.223 describes the interface [ara::rest::Router::SetDefaultHandler.](#)] ([RS\\_CM\\_00300](#))

### 8.21.10 RouteCount

<b>Service name:</b>	ara::rest::Router::RouteCount	
<b>Type:</b>	Member function	
<b>Syntax:</b>	std::size_t ara::rest::Router::RouteCount()	
<b>Function param:</b>	None	
<b>Return value:</b>	the number of user-defined routes	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/routing.h	
<b>Class:</b>	ara::rest::Router	
<b>Description:</b>	Returns the number of routes. Returns the number of specified routes, exclusive of the default route.	

**Table 8.224: ara::rest::Router::RouteCount**

[SWS\_REST\_02199] [ara::rest::Router::RouteCount](#) [Table 8.224 describes the interface [ara::rest::Router::RouteCount.](#)] ([RS\\_CM\\_00300](#))

### 8.21.11 Routes

<b>Service name:</b>	ara::rest::Router::Routes	
<b>Type:</b>	Member function	
<b>Syntax:</b>	RouteRange ara::rest::Router::Routes()	
<b>Function param:</b>	None	
<b>Return value:</b>	an iterator range of routes	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/routing.h	
<b>Class:</b>	ara::rest::Router	

<b>Description:</b>	Provides direc access to the set of routes.
---------------------	---

**Table 8.225: ara::rest::Router::Routes**

[SWS\_REST\_02200] **ara::rest::Router::Routes** [Table 8.225 describes the interface `ara::rest::Router::Routes.`] ([RS\\_CM\\_00300](#))

### 8.21.12 Routes

<b>Service name:</b>	ara::rest::Router::Routes
<b>Type:</b>	Member function
<b>Syntax:</b>	ConstRouteRange ara::rest::Router::Routes() const
<b>Function param:</b>	None
<b>Return value:</b>	an iterator range of routes
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/routing.h
<b>Class:</b>	ara::rest::Router
<b>Description:</b>	Provides direc access to the set of routes.

**Table 8.226: ara::rest::Router::Routes**

[SWS\_REST\_02201] **ara::rest::Router::Routes** [Table 8.226 describes the interface `ara::rest::Router::Routes.`] ([RS\\_CM\\_00300](#))

### 8.21.13 RemoveRoute

<b>Service name:</b>	ara::rest::Router::RemoveRoute
<b>Type:</b>	Member function
<b>Syntax:</b>	void ara::rest::Router::RemoveRoute(RouteRange::Iterator iter)
<b>Function param:</b>	iter   iterator referencing the route to remove
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/routing.h
<b>Class:</b>	ara::rest::Router
<b>Description:</b>	Removes a route from the set.

**Table 8.227: ara::rest::Router::RemoveRoute**

[SWS\_REST\_02202] **ara::rest::Router::RemoveRoute** [Table 8.227 describes the interface `ara::rest::Router::RemoveRoute.`] ([RS\\_CM\\_00300](#))

### 8.21.14 FindRoute

<b>Service name:</b>	ara::rest::Router::FindRoute	
<b>Type:</b>	Member function	
<b>Syntax:</b>	RouteRange::Iterator ara::rest::Router::FindRoute(const Route &route)	
<b>Function param:</b>	route	route to search for
<b>Return value:</b>	an iterator to the route if it exists in the set or Routes.end() if no such route was found.	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/routing.h	
<b>Class:</b>	ara::rest::Router	
<b>Description:</b>	Searches for a given route.	

**Table 8.228: ara::rest::Router::FindRoute**

[SWS\_REST\_02203] **ara::rest::Router::FindRoute** [Table 8.228 describes the interface `ara::rest::Router::FindRoute`.] ([RS\\_CM\\_00300](#))

### 8.21.15 Clear

<b>Service name:</b>	ara::rest::Router::Clear	
<b>Type:</b>	Member function	
<b>Syntax:</b>	void ara::rest::Router::Clear()	
<b>Function param:</b>	None	
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/routing.h	
<b>Class:</b>	ara::rest::Router	
<b>Description:</b>	Removes all routes.	

**Table 8.229: ara::rest::Router::Clear**

[SWS\_REST\_02204] **ara::rest::Router::Clear** [Table 8.229 describes the interface `ara::rest::Router::Clear`.] ([RS\\_CM\\_00300](#))

## 8.22 ara::rest::Route

[SWS\_REST\_02205] [ara::rest::Route class shall be declared in the `ara/rest/routing.h` header file:

```
1     class ara::rest::Route;
```

] ([RS\\_CM\\_00300](#))

### 8.22.1 Upshot

<b>Name:</b>	Upshot
<b>Type:</b>	Member enumeration
<b>Range:</b>	kAccept kYield kDefault
<b>Syntax:</b>	enum class Upshot { kAccept, kYield, kDefault };
<b>Header file:</b>	ara/rest/routing.h
<b>Class:</b>	ara::rest::Route
<b>Description:</b>	Instructions for a Router on how to proceed after a route handler functions returns. A route handler function must return one of these values to instruct the router how to proceed after executing the current handler.

**Table 8.230: ara::rest::Route::Upshot**

[SWS\_REST\_02206] **Upshot** [Table 8.230 describes the enumeration datatype `ara::rest::Route::Upshot`.] ([RS\\_CM\\_00300](#))

## 8.2.2.2 RouteHandlerType

<b>Name:</b>	RouteHandlerType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::Route::RouteHandlerType = Upshot(const ServerRequest&, ServerReply&, const Matches&)
<b>Header file:</b>	ara/rest/routing.h
<b>Class:</b>	ara::rest::Route
<b>Description:</b>	The type of the user-define handler function to be invoked if this Route matches the Pattern.

**Table 8.231: ara::rest::Route::RouteHandlerType**

[SWS\_REST\_02207] **RouteHandlerType** [Table 8.231 describes the type alias `ara::rest::Route::RouteHandlerType`.] ([RS\\_CM\\_00300](#))

## 8.2.2.3 Route

<b>Service name:</b>	ara::rest::Route::Route	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ara::rest::Route::Route(RequestMethod met, const Pattern &pat, const Function< RouteHandlerType > &hnd)	
<b>Function param:</b>	met	a disjunction (logical OR) of RequestMethods to match against
<b>Function param:</b>	pat	a URI Pattern to match against
<b>Function param:</b>	hnd	a user-defined handler

<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/routing.h
<b>Class:</b>	ara::rest::Route
<b>Description:</b>	Constructs a route.

**Table 8.232: ara::rest::Route::Route**

[SWS\_REST\_02208] **ara::rest::Route::Route** [Table 8.232 describes the interface `ara::rest::Route::Route`.] ([RS\\_CM\\_00300](#))

#### 8.22.4 operator()

<b>Service name:</b>	ara::rest::Route::operator()	
<b>Type:</b>	Member function	
<b>Syntax:</b>	Upshot ara::rest::Route::operator() (const ServerRequest &req, ServerReply &rep) const	
<b>Function param:</b>	req	a request
<b>Function param:</b>	rep	a reply
<b>Return value:</b>	a value of type Upshot	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/routing.h	
<b>Class:</b>	ara::rest::Route	
<b>Description:</b>	ara::rest::Server compliant handler function This function is invoked by the Router to test the current Route for a match	

**Table 8.233: ara::rest::Route::operator()**

[SWS\_REST\_02209] **ara::rest::Route::operator()** [Table 8.233 describes the interface `ara::rest::Route::operator()`.] ([RS\\_CM\\_00300](#))

#### 8.22.5 GetRequestMethod

<b>Service name:</b>	ara::rest::Route::GetRequestMethod
<b>Type:</b>	Member function
<b>Syntax:</b>	RequestMethod ara::rest::Route::GetRequestMethod() const
<b>Function param:</b>	None
<b>Return value:</b>	reference to a Pattern
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/routing.h
<b>Class:</b>	ara::rest::Route
<b>Description:</b>	Provides access to the underlying Pattern object.

**Table 8.234: ara::rest::Route::GetRequestMethod**

[SWS\_REST\_02210] `ara::rest::Route::GetRequestMethod` [Table 8.234 describes the interface `ara::rest::Route::GetRequestMethod`.] ([RS\\_CM\\_00300](#))

## 8.22.6 GetPattern

<b>Service name:</b>	<code>ara::rest::Route::GetPattern</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>const Pattern&amp; ara::rest::Route::GetPattern() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	reference to a Pattern
<b>Exceptions:</b>	<code>noexcept</code>
<b>Header file:</b>	<code>ara/rest/routing.h</code>
<b>Class:</b>	<code>ara::rest::Route</code>
<b>Description:</b>	Provides access to the underlying Pattern object.

**Table 8.235: `ara::rest::Route::GetPattern`**

[SWS\_REST\_02211] `ara::rest::Route::GetPattern` [Table 8.235 describes the interface `ara::rest::Route::GetPattern`.] ([RS\\_CM\\_00300](#))

## 8.22.7 operator==

<b>Service name:</b>	<code>ara::rest::Route::operator==</code>	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<code>friend bool operator==(const Route &amp;a, const Route &amp;b)</code>	
<b>Function param:</b>	<code>a</code>	a route
<b>Function param:</b>	<code>b</code>	a route
<b>Return value:</b>	true if equal	
<b>Exceptions:</b>	<code>noexcept</code>	
<b>Header file:</b>	<code>ara/rest/routing.h</code>	
<b>Namespace:</b>	<code>ara::rest::Route</code>	
<b>Description:</b>	Tests for equality.	

**Table 8.236: `ara::rest::Route::operator==`**

[SWS\_REST\_02212] `ara::rest::Route::operator==` [Table 8.236 describes the interface `ara::rest::Route::operator==`.] ([RS\\_CM\\_00300](#))

## 8.22.8 operator!=

<b>Service name:</b>	<code>ara::rest::Route::operator!=</code>	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<code>friend bool operator!=(const Route &amp;a, const Route &amp;b)</code>	
<b>Function param:</b>	<code>a</code>	a route

<b>Function param:</b>	b	a route
<b>Return value:</b>	true if unequal	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/routing.h	
<b>Namespace:</b>	ara::rest::Route	
<b>Description:</b>	Tests for inequality.	

**Table 8.237: ara::rest::Route::operator!=**

[SWS\_REST\_02213] `ara::rest::Route::operator!=` [Table 8.237 describes the interface `ara::rest::Route::operator!=`.] ([RS\\_CM\\_00300](#))

### 8.22.9 operator<

<b>Service name:</b>	ara::rest::Route::operator<	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	friend bool operator<(const Route &a, const Route &b)	
<b>Function param:</b>	a	a route
<b>Function param:</b>	b	a route
<b>Return value:</b>	true if a compares less-than b	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/routing.h	
<b>Namespace:</b>	ara::rest::Route	
<b>Description:</b>	Tests whether a route is less-than another.	

**Table 8.238: ara::rest::Route::operator<**

[SWS\_REST\_02214] `ara::rest::Route::operator<` [Table 8.238 describes the interface `ara::rest::Route::operator<`.] ([RS\\_CM\\_00300](#))

## 8.23 ara::rest::ServerEvent

[SWS\_REST\_02215] [`ara::rest::ServerEvent` class shall be declared in the `ara/rest/server.h` header file:

```
1     class ara::rest::ServerEvent;
```

] ([RS\\_CM\\_00300](#))

### 8.23.1 ServerEvent

<b>Service name:</b>	ara::rest::ServerEvent::ServerEvent	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>ara::rest::ServerEvent::ServerEvent(const ServerEvent &amp;)=delete</code>	



<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::ServerEvent
<b>Description:</b>	Non-copyable.

**Table 8.239: ara::rest::ServerEvent::ServerEvent**

[SWS\_REST\_02216] **ara::rest::ServerEvent::ServerEvent** [Table 8.239 describes the interface `ara::rest::ServerEvent::ServerEvent`.] ([RS\\_CM\\_00300](#))

### 8.23.2 operator=

<b>Service name:</b>	ara::rest::ServerEvent::operator=
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>ServerEvent&amp; ara::rest::ServerEvent::operator=(const ServerEvent &amp;)=delete</code>
<b>Return value:</b>	a value of type <code>ServerEvent &amp;</code>
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::ServerEvent
<b>Description:</b>	Non-copy-assignable.

**Table 8.240: ara::rest::ServerEvent::operator=**

[SWS\_REST\_02217] **ara::rest::ServerEvent::operator=** [Table 8.240 describes the interface `ara::rest::ServerEvent::operator=`.] ([RS\\_CM\\_00300](#))

### 8.23.3 Notify

<b>Service name:</b>	ara::rest::ServerEvent::Notify
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Task&lt;void&gt; ara::rest::ServerEvent::Notify(const Pointer&lt; ogm::Object &gt; &amp;data)</code>
<b>Function param:</b>	data   payload to be notified
<b>Return value:</b>	A task waiting for the notification to complete.
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::ServerEvent
<b>Description:</b>	Issues a change notification to its corresponding Server. Each server-side event has at least one corresponding client-side event. Notify does NOT notify these clients. It notifies its corresponding server of potential changes to the data referenced by the event (URI). It is the server's decision if this event is sent to the clients or not based on the trigger conditions. The data that should be notified is given in the parameter data.

**Table 8.241: ara::rest::ServerEvent::Notify**

[SWS\_REST\_02218] `ara::rest::ServerEvent::Notify` [Table 8.241 describes the interface `ara::rest::ServerEvent::Notify`.] ([RS\\_CM\\_00300](#))

### 8.23.4 Notify

<b>Service name:</b>	<code>ara::rest::ServerEvent::Notify</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Task&lt;void&gt; ara::rest::ServerEvent::Notify()</code>
<b>Return value:</b>	A task waiting for the notification to complete.
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/server.h</code>
<b>Class:</b>	<code>ara::rest::ServerEvent</code>
<b>Description:</b>	Issues a change notification to its corresponding Server. Each server-side event has at least one corresponding client-side event. Notify does NOT notify these clients. It notifies its corresponding server of potential changes to the data referenced by the event (URI). It is the server's decision if this event is send to the clients or not based on the trigger conditions. Note that this function will trigger automatically an GET request to <code>ara::rest::Server</code> application to get the newest data.

**Table 8.242: `ara::rest::ServerEvent::Notify`**

[SWS\_REST\_02889] `ara::rest::ServerEvent::Notify` [Table 8.242 describes the interface `ara::rest::ServerEvent::Notify`.] ([RS\\_CM\\_00300](#))

### 8.23.5 SetSubscriptionState

<b>Service name:</b>	<code>ara::rest::ServerEvent::SetSubscriptionState</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>void ara::rest::ServerEvent::SetSubscriptionState(const ara::rest::SubscriptionState )</code>
<b>Function param:</b>	state   SubscriptionState of corresponding server event
<b>Return value:</b>	void
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/server.h</code>
<b>Class:</b>	<code>ara::rest::ServerEvent</code>
<b>Description:</b>	Set subscription state from server side. Enables the server application to react to event subscriptions received from <code>ara::rest::Client</code>

**Table 8.243: `ara::rest::ServerEvent::SetSubscriptionState`**

[SWS\_REST\_02219] `ara::rest::ServerEvent::SetSubscriptionState` [Table 8.243 describes the interface `ara::rest::ServerEvent::SetSubscriptionState`.] ([RS\\_CM\\_00300](#))

### 8.23.6 GetSubscriptionState

<b>Service name:</b>	ara::rest::ServerEvent::GetSubscriptionState
<b>Type:</b>	Member function
<b>Syntax:</b>	SubscriptionState ara::rest::ServerEvent::GetSubscriptionState() const
<b>Function param:</b>	None
<b>Return value:</b>	the current subscription state as perceived by the client
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::ServerEvent
<b>Description:</b>	Denotes the current subscription state.

**Table 8.244: ara::rest::ServerEvent::GetSubscriptionState**

[SWS\_REST\_02220] [ara::rest::ServerEvent::GetSubscriptionState](#) [Table 8.244 describes the interface [ara::rest::ServerEvent::GetSubscriptionState.](#)] ([RS\\_CM\\_00300](#))

### 8.23.7 GetUri

<b>Service name:</b>	ara::rest::ServerEvent::GetUri
<b>Type:</b>	Member function
<b>Syntax:</b>	const Uri& ara::rest::ServerEvent::GetUri() const
<b>Function param:</b>	None
<b>Return value:</b>	the Uri corresponding to this event subscription
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::ServerEvent
<b>Description:</b>	Returns the event Uri.

**Table 8.245: ara::rest::ServerEvent::GetUri**

[SWS\_REST\_02221] [ara::rest::ServerEvent::GetUri](#) [Table 8.245 describes the interface [ara::rest::ServerEvent::GetUri.](#)] ([RS\\_CM\\_00300](#))

### 8.23.8 SendError

<b>Service name:</b>	ara::rest::ServerEvent::SendError	
<b>Type:</b>	Member function	
<b>Syntax:</b>	void ara::rest::ServerEvent::SendError(const unsigned int errorCode, const StringView& errorMessage);	
<b>Function param:</b>	errorCode	The error code
<b>Function param:</b>	errorMessage	The error code message
<b>Return value:</b>	None	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/server.h	
<b>Class:</b>	ara::rest::ServerEvent	

<b>Description:</b>	During the life time of an event an error can occur. E.g. the subscribed resource is no longer available. With this method the subscribed Client can be notified.
---------------------	---

**Table 8.246: ara::rest::ServerEvent::SendError**

[SWS\_REST\_02805] **ara::rest::ServerEvent::SendError** [Table 8.246 describes the interface `ara::rest::ServerEvent::SendError`.] ([RS\\_CM\\_00300](#))

### 8.23.9 operator==

<b>Service name:</b>	ara::rest::ServerEvent::operator==	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	friend bool operator==(const ServerEvent &a, const ServerEvent &b)	
<b>Function param:</b>	a	an event
<b>Function param:</b>	b	an event
<b>Return value:</b>	true if a and b are equal	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/server.h	
<b>Namespace:</b>	ara::rest::ServerEvent	
<b>Description:</b>	Tests events for equality.	

**Table 8.247: ara::rest::ServerEvent::operator==**

[SWS\_REST\_02222] **ara::rest::ServerEvent::operator==** [Table 8.247 describes the interface `ara::rest::ServerEvent::operator==`.] ([RS\\_CM\\_00300](#))

### 8.23.10 operator!=

<b>Service name:</b>	ara::rest::ServerEvent::operator!=	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	friend bool operator!=(const ServerEvent &a, const ServerEvent &b)	
<b>Function param:</b>	a	an event
<b>Function param:</b>	b	an event
<b>Return value:</b>	true if a and b are unequal	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/server.h	
<b>Namespace:</b>	ara::rest::ServerEvent	
<b>Description:</b>	Tests events for inequality.	

**Table 8.248: ara::rest::ServerEvent::operator!=**

[SWS\_REST\_02223] **ara::rest::ServerEvent::operator!=** [Table 8.248 describes the interface `ara::rest::ServerEvent::operator!=`.] ([RS\\_CM\\_00300](#))

### 8.23.11 operator<

<b>Service name:</b>	ara::rest::ServerEvent::operator<	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	friend bool operator<(const ServerEvent &a, const ServerEvent &b)	
<b>Function param:</b>	a	an event
<b>Function param:</b>	b	an event
<b>Return value:</b>	true if a less-than b	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/server.h	
<b>Namespace:</b>	ara::rest::ServerEvent	
<b>Description:</b>	Tests events for their partial order Order criterion is implementation-defined.	

**Table 8.249: ara::rest::ServerEvent::operator<**

[SWS\_REST\_02224] ara::rest::ServerEvent::operator< [Table 8.249 describes the interface `ara::rest::ServerEvent::operator<.`] (RS\_CM\_00300)

## 8.24 ara::rest::ServerReply

[SWS\_REST\_02225] [ara::rest::ServerReply class shall be declared in the `ara/rest/server.h` header file:

```
1     class ara::rest::ServerReply;
```

] (RS\_CM\_00300)

### 8.24.1 ServerReply

<b>Service name:</b>	ara::rest::ServerReply::ServerReply
<b>Type:</b>	Member function
<b>Syntax:</b>	ara::rest::ServerReply::ServerReply(const ServerReply &)=delete
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::ServerReply
<b>Description:</b>	Non-copyable.

**Table 8.250: ara::rest::ServerReply::ServerReply**

[SWS\_REST\_02226] ara::rest::ServerReply::ServerReply [Table 8.250 describes the interface `ara::rest::ServerReply::ServerReply.`] (RS\_CM\_00300)

### 8.24.2 operator=

<b>Service name:</b>	ara::rest::ServerReply::operator=
<b>Type:</b>	Member function
<b>Syntax:</b>	ServerReply& ara::rest::ServerReply::operator=(const ServerReply &)=delete
<b>Return value:</b>	a value of type ServerReply &
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::ServerReply
<b>Description:</b>	Non-copy-assignable.

**Table 8.251: ara::rest::ServerReply::operator=**

[SWS\_REST\_02227] `ara::rest::ServerReply::operator=` [Table 8.251 describes the interface `ara::rest::ServerReply::operator=.`] ([RS\\_CM\\_00300](#))

### 8.24.3 GetHeader

<b>Service name:</b>	ara::rest::ServerReply::GetHeader
<b>Type:</b>	Member function
<b>Syntax:</b>	ReplyHeader& ara::rest::ServerReply::GetHeader()
<b>Function param:</b>	None
<b>Return value:</b>	a reference to a RequestHeader
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::ServerReply
<b>Description:</b>	Provides access to the reply message header.

**Table 8.252: ara::rest::ServerReply::GetHeader**

[SWS\_REST\_02228] `ara::rest::ServerReply::GetHeader` [Table 8.252 describes the interface `ara::rest::ServerReply::GetHeader.`] ([RS\\_CM\\_00300](#))

### 8.24.4 Send

<b>Service name:</b>	ara::rest::ServerReply::Send
<b>Type:</b>	Member function
<b>Syntax:</b>	Task<void> ara::rest::ServerReply::Send(const Pointer< ogm::Object > &data={})
<b>Function param:</b>	data   payload to be transmitted
<b>Return value:</b>	a task waiting for the transmission to complete
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::ServerReply
<b>Description:</b>	Send a reply to the peer that has issued the request. If this function is not invoked explicitly, the endpoint will transmit a default reply. If <code>Redirect()</code> has been called before, these functions must be used.

**Table 8.253: ara::rest::ServerReply::Send**

[SWS\_REST\_02229] **ara::rest::ServerReply::Send** [Table 8.253 describes the interface `ara::rest::ServerReply::Send`.] (*RS\_CM\_00300*)

### 8.24.5 Send

<b>Service name:</b>	ara::rest::ServerReply::Send	
<b>Type:</b>	Member function	
<b>Syntax:</b>	Task<void> ara::rest::ServerReply::Send(const Pointer< ogm::Object > &&data)	
<b>Function param:</b>	data	payload to be transmitted
<b>Return value:</b>	a task waiting for the transmission to complete	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/server.h	
<b>Class:</b>	ara::rest::ServerReply	
<b>Description:</b>	Send a reply to the peer that has issued the request. Same as other Send(), only with move semantics	

**Table 8.254: ara::rest::ServerReply::Send**

[SWS\_REST\_02230] **ara::rest::ServerReply::Send** [Table 8.254 describes the interface `ara::rest::ServerReply::Send`.] (*RS\_CM\_00300*)

### 8.24.6 Send

<b>Service name:</b>	ara::rest::ServerReply::Send	
<b>Type:</b>	Member function	
<b>Syntax:</b>	Task<void> ara::rest::ServerReply::Send(const StringView &data)	
<b>Function param:</b>	data	binary payload to be transmitted
<b>Return value:</b>	a task waiting for the transmission to complete	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/server.h	
<b>Class:</b>	ara::rest::ServerReply	
<b>Description:</b>	Send a reply with binary data to the peer that has issued the request.	

**Table 8.255: ara::rest::ServerReply::Send**

[SWS\_REST\_02932] **ara::rest::ServerReply::Send** [Table 8.255 describes the interface `ara::rest::ServerReply::Send`.] (*RS\_CM\_00300*)

### 8.24.7 Redirect

<b>Service name:</b>	ara::rest::ServerReply::Redirect	
<b>Type:</b>	Member function	
<b>Syntax:</b>	Task<void> ara::rest::ServerReply::Redirect(const Uri &uri)	
<b>Function param:</b>	uri	location to redirect to
<b>Return value:</b>	a value of type Task< void >	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/server.h	
<b>Class:</b>	ara::rest::ServerReply	
<b>Description:</b>	Issues a redirect command to the connected client. Must not be called after Send().	

**Table 8.256: ara::rest::ServerReply::Redirect**

[SWS\_REST\_02231] **ara::rest::ServerReply::Redirect** [Table 8.256 describes the interface `ara::rest::ServerReply::Redirect`.] (RS\_CM\_00300)

## 8.25 ara::rest::ServerRequest

[SWS\_REST\_02232] [ara::rest::ServerRequest class shall be declared in the ara/rest/server.h header file:

```
1     class ara::rest::ServerRequest;
```

] (RS\_CM\_00300)

### 8.25.1 ServerRequest

<b>Service name:</b>	ara::rest::ServerRequest::ServerRequest	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ara::rest::ServerRequest::ServerRequest(const ServerRequest &)=delete	
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/server.h	
<b>Class:</b>	ara::rest::ServerRequest	
<b>Description:</b>	Non-copyable.	

**Table 8.257: ara::rest::ServerRequest::ServerRequest**

[SWS\_REST\_02233] **ara::rest::ServerRequest::ServerRequest** [Table 8.257 describes the interface `ara::rest::ServerRequest::ServerRequest`.] (RS\_CM\_00300)

### 8.25.2 operator=



<b>Service name:</b>	<code>ara::rest::ServerRequest::operator=</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>ServerRequest&amp;</code> <code>ara::rest::ServerRequest::operator=(const</code> <code>ServerRequest &amp;)=delete</code>
<b>Return value:</b>	a value of type <code>ServerRequest &amp;</code>
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/server.h</code>
<b>Class:</b>	<code>ara::rest::ServerRequest</code>
<b>Description:</b>	Non-copy-assignable.

**Table 8.258: `ara::rest::ServerRequest::operator=`**

[SWS\_REST\_02234] `ara::rest::ServerRequest::operator=` [Table 8.258 describes the interface `ara::rest::ServerRequest::operator=.`] ([RS\\_CM\\_00300](#))

### 8.25.3 GetHeader

<b>Service name:</b>	<code>ara::rest::ServerRequest::GetHeader</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>RequestHeader const&amp;</code> <code>ara::rest::ServerRequest::GetHeader() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a reference to a <code>RequestHeader</code>
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/server.h</code>
<b>Class:</b>	<code>ara::rest::ServerRequest</code>
<b>Description:</b>	Provides access to the message header. Requests the message header from the endpoint. Accessing the message header is always synchronous.

**Table 8.259: `ara::rest::ServerRequest::GetHeader`**

[SWS\_REST\_02235] `ara::rest::ServerRequest::GetHeader` [Table 8.259 describes the interface `ara::rest::ServerRequest::GetHeader.`] ([RS\\_CM\\_00300](#))

### 8.25.4 GetObject

<b>Service name:</b>	<code>ara::rest::ServerRequest::GetObject</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Task&lt;ogm::Object const&amp;&gt;</code> <code>ara::rest::ServerRequest::GetObject() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	returns a task waiting for the message payload to be received.
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/server.h</code>
<b>Class:</b>	<code>ara::rest::ServerRequest</code>
<b>Description:</b>	Obtains the request message payload.

**Table 8.260: ara::rest::ServerRequest::GetObject**

[SWS\_REST\_02236] **ara::rest::ServerRequest::GetObject** [Table 8.260 describes the interface `ara::rest::ServerRequest::GetObject.`] (*RS\_CM\_00300*)

### 8.25.5 ReleaseObject

<b>Service name:</b>	ara::rest::ServerRequest::ReleaseObject
<b>Type:</b>	Member function
<b>Syntax:</b>	Task<Pointer<ogm::Object> > ara::rest::ServerRequest::ReleaseObject()
<b>Function param:</b>	None
<b>Return value:</b>	returns a task waiting for the message payload to be received.
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::ServerRequest
<b>Description:</b>	Obtains the reply message payload.

**Table 8.261: ara::rest::ServerRequest::ReleaseObject**

[SWS\_REST\_02237] **ara::rest::ServerRequest::ReleaseObject** [Table 8.261 describes the interface `ara::rest::ServerRequest::ReleaseObject.`] (*RS\_CM\_00300*)

### 8.25.6 ReleaseBinary

<b>Service name:</b>	ara::rest::ServerRequest::ReleaseBinary
<b>Type:</b>	Member function
<b>Syntax:</b>	Task<Pointer<ara::core::String> > ara::rest::ServerRequest::ReleaseBinary()
<b>Function param:</b>	None
<b>Return value:</b>	returns a task waiting for the message payload to be received.
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::ServerRequest
<b>Description:</b>	Obtains the reply message payload.

**Table 8.262: ara::rest::ServerRequest::ReleaseBinary**

[SWS\_REST\_02991] **ara::rest::ServerRequest::ReleaseBinary** [Table 8.262 describes the interface `ara::rest::ServerRequest::ReleaseBinary.`] (*RS\_CM\_00300*)

## 8.26 ara::rest::Server

[SWS\_REST\_02238] [ara::rest::Server class shall be declared in the ara/rest/server.h header file:

```
1         class ara::rest::Server;
```

](RS\_CM\_00300, RS\_CM\_00301)

### 8.26.1 RequestHandlerType

<b>Name:</b>	RequestHandlerType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::Server::RequestHandlerType = void(const ServerRequest&, ServerReply&)
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::Server
<b>Description:</b>	Type of user-defined request handlers.

**Table 8.263: ara::rest::Server::RequestHandlerType**

[SWS\_REST\_02239] RequestHandlerType [Table 8.263 describes the type alias ara::rest::Server::RequestHandlerType.](RS\_CM\_00300, RS\_CM\_00301)

### 8.26.2 SubscriptionHandlerType

<b>Name:</b>	SubscriptionHandlerType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::Server::SubscriptionHandlerType = void(ServerEvent)
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::Server
<b>Description:</b>	Denotes a subscription handler type.

**Table 8.264: ara::rest::Server::SubscriptionHandlerType**

[SWS\_REST\_02240] SubscriptionHandlerType [Table 8.264 describes the type alias ara::rest::Server::SubscriptionHandlerType.](RS\_CM\_00300, RS\_CM\_00301)

### 8.26.3 SubscriptionStateHandlerType

<b>Name:</b>	SubscriptionStateHandlerType
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::Server::SubscriptionStateHandlerType = void(ServerEvent&, SubscriptionState)

<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::Server
<b>Description:</b>	Denotes a callback to call if subscription status changes.

**Table 8.265: ara::rest::Server::SubscriptionStateHandlerType**

[SWS\_REST\_02241] **SubscriptionStateHandlerType** [Table 8.265 describes the type alias [ara::rest::Server::SubscriptionStateHandlerType](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

## 8.26.4 Server

<b>Service name:</b>	ara::rest::Server::Server
<b>Type:</b>	Member function
<b>Syntax:</b>	ara::rest::Server::Server(const Server &)=delete
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::Server
<b>Description:</b>	Server is non-copy-constructible.

**Table 8.266: ara::rest::Server::Server**

[SWS\_REST\_02242] **ara::rest::Server::Server** [Table 8.266 describes the interface [ara::rest::Server::Server](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

## 8.26.5 operator=

<b>Service name:</b>	ara::rest::Server::operator=
<b>Type:</b>	Member function
<b>Syntax:</b>	Server& ara::rest::Server::operator=(const Server &)=delete
<b>Return value:</b>	a value of type <code>Server &amp;</code>
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::Server
<b>Description:</b>	Server is non-copy-assignable.

**Table 8.267: ara::rest::Server::operator=**

[SWS\_REST\_02243] **ara::rest::Server::operator=** [Table 8.267 describes the interface [ara::rest::Server::operator=](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

## 8.26.6 Server

<b>Service name:</b>	ara::rest::Server::Server	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>ara::rest::Server::Server(const ara::rest::InstanceIdentifier &amp;inst_id, const Function&lt; RequestHandlerType &gt; &amp;hnd, Allocator *alloc=GetDefaultAllocator())</pre>	
<b>Function param:</b>	inst_id	ara::rest::InstanceIdentifier identifies concrete service instace
<b>Function param:</b>	hnd	request handler function
<b>Function param:</b>	alloc	allocator for dynamic memory
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/server.h	
<b>Class:</b>	ara::rest::Server	
<b>Description:</b>	Constructs a server.	

**Table 8.268: ara::rest::Server::Server**

[SWS\_REST\_02244] ara::rest::Server::Server [Table 8.268 describes the interface ara::rest::Server::Server.] (RS\_CM\_00300, RS\_CM\_00301, RS\_CM\_00310)

### 8.26.7 Start

<b>Service name:</b>	ara::rest::Server::Start	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>Task&lt;void&gt; ara::rest::Server::Start(StartupPolicy policy=StartupPolicy::kDetached)</pre>	
<b>Function param:</b>	policy	denotes whether caller is blocked or not.
<b>Return value:</b>	a task waiting for Stop() to be invoked	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/server.h	
<b>Class:</b>	ara::rest::Server	
<b>Description:</b>	Instruct a server to begin serving clients. A server does not serve anything unless Start() is invoked. A server can be started within the execution context of the caller or within its own execution context (usually this is a thread). If StartupPolicy::kAttached, then Start() blocks its caller and only returns of Stop() is called. If Startuppolicy::kDetached, Start() does not block its caller but returns a task for synchronization. The caller may wait on the task, which blocks until Stop() is invoked.	

**Table 8.269: ara::rest::Server::Start**

[SWS\_REST\_02245] ara::rest::Server::Start [Table 8.269 describes the interface ara::rest::Server::Start.] (RS\_CM\_00300, RS\_CM\_00301)

### 8.26.8 Stop

<b>Service name:</b>	ara::rest::Server::Stop
----------------------	-------------------------

<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Task&lt;void&gt; ara::rest::Server::Stop(ShutdownPolicy policy=ShutdownPolicy::kGraceful)</code>
<b>Function param:</b>	policy denotes how server is stopped.
<b>Return value:</b>	return type
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::Server
<b>Description:</b>	Instructs a server to stop serving clients. A client can be stopped either as fast as possible or "gracefully". If ShutdownPolicy::kForced then all connections are terminated instantly and all ongoing processes shall be terminated as fast as possible. If ShutdownPolicy::kGraceful then a server will stop accepting new requests but will wait until all requests have been served.

**Table 8.270: ara::rest::Server::Stop**

[SWS\_REST\_02246] **ara::rest::Server::Stop** [Table 8.270 describes the interface `ara::rest::Server::Stop`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

### 8.26.9 ObserveSubscriptions

<b>Service name:</b>	ara::rest::Server::ObserveSubscriptions
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>void ara::rest::Server::ObserveSubscriptions(const Function&lt; SubscriptionHandlerType &gt; &amp;shnd, const Function&lt; SubscriptionStateHandlerType &gt; &amp;sshnd)</code>
<b>Function param:</b>	shnd a subscription handler function
<b>Function param:</b>	sshnd a subscription state handler function
<b>Return value:</b>	None
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::Server
<b>Description:</b>	Registers a user-defined subscription handler. A server can handle event subscriptions by default. Unless a user-defined handler function is registered explicitly, event subscriptions are not visible to the user. This implies that subscriptions with EventPolicy::kTriggered never receive notifications.

**Table 8.271: ara::rest::Server::ObserveSubscriptions**

[SWS\_REST\_02247] **ara::rest::Server::ObserveSubscriptions** [Table 8.271 describes the interface `ara::rest::Server::ObserveSubscriptions`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

### 8.26.10 GetError

<b>Service name:</b>	ara::rest::Server::GetError
----------------------	-----------------------------

<b>Type:</b>	Member function
<b>Syntax:</b>	<code>ara::core::ErrorCode ara::rest::Server::GetError() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a reference to the server Status
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/server.h
<b>Class:</b>	ara::rest::Server
<b>Description:</b>	Obtain server status.

**Table 8.272: ara::rest::Server::GetError**

[SWS\_REST\_02248] `ara::rest::Server::GetError` [Table 8.272 describes the interface `ara::rest::Server::GetError`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

### 8.26.11 ObserveError

<b>Service name:</b>	<code>ara::rest::Server::ObserveError</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>void ara::rest::Server::ObserveError(const Function&lt; void(ara::core::ErrorCode)&gt; &amp;hnd)</code>	
<b>Function param:</b>	<code>hnd</code>	user-defined handler function to to called on status changes
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/server.h	
<b>Class:</b>	ara::rest::Server	
<b>Description:</b>	Observe status changes.	

**Table 8.273: ara::rest::Server::ObserveError**

[SWS\_REST\_02249] `ara::rest::Server::ObserveError` [Table 8.273 describes the interface `ara::rest::Server::ObserveError`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00301](#))

## 8.27 ara::rest::StdAllocator

[SWS\_REST\_02250] [`ara::rest::StdAllocator` class shall be declared in the `ara/rest/allocator.h` header file:

```
1     template <typename T>
2     class ara::rest::StdAllocator;
```

]([RS\\_CM\\_00300](#))

### 8.27.1 value\_type

<b>Name:</b>	value_type
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::StdAllocator< T >::value_type = T
<b>Header file:</b>	ara/rest/allocator.h
<b>Class:</b>	ara::rest::StdAllocator
<b>Description:</b>	Type this allocator is bound to.

**Table 8.274: ara::rest::StdAllocator::value\_type**

[SWS\_REST\_02251] `value_type` [Table 8.274 describes the type alias `ara::rest::StdAllocator::value_type`.] ([RS\\_CM\\_00300](#))

## 8.27.2 StdAllocator

<b>Service name:</b>	ara::rest::StdAllocator::StdAllocator
<b>Type:</b>	Member function
<b>Syntax:</b>	ara::rest::StdAllocator< T >::StdAllocator()
<b>Function param:</b>	None
<b>Return value:</b>	None
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/allocator.h
<b>Class:</b>	ara::rest::StdAllocator
<b>Description:</b>	Default constructs this allocator See <code>std::pmr::polymorphic_allocator</code> documentation for details.

**Table 8.275: ara::rest::StdAllocator::StdAllocator**

[SWS\_REST\_02252] `ara::rest::StdAllocator::StdAllocator` [Table 8.275 describes the interface `ara::rest::StdAllocator::StdAllocator`.] ([RS\\_CM\\_00300](#))

## 8.27.3 StdAllocator

<b>Service name:</b>	ara::rest::StdAllocator::StdAllocator
<b>Type:</b>	Member function
<b>Syntax:</b>	ara::rest::StdAllocator< T >::StdAllocator(Allocator *a)
<b>Function param:</b>	a   an allocator
<b>Return value:</b>	None
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/allocator.h
<b>Class:</b>	ara::rest::StdAllocator
<b>Description:</b>	Default constructs this allocator See <code>std::pmr::polymorphic_allocator</code> documentation for details.

**Table 8.276: ara::rest::StdAllocator::StdAllocator**



[SWS\_REST\_02253] `ara::rest::StdAllocator::StdAllocator` [Table 8.276 describes the interface `ara::rest::StdAllocator::StdAllocator`.] ([RS\\_CM\\_00300](#))

### 8.27.4 StdAllocator

<b>Service name:</b>	<code>ara::rest::StdAllocator::StdAllocator</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>template &lt;typename U &gt; ara::rest::StdAllocator&lt; T &gt;::StdAllocator(StdAllocator&lt; U &gt; const &amp;do_not_use)</pre>	
<b>Function param:</b>	<code>do_not_use</code>	meaningless.
<b>Return value:</b>	None	
<b>Exceptions:</b>	<code>noexcept</code>	
<b>Header file:</b>	<code>ara/rest/allocator.h</code>	
<b>Class:</b>	<code>ara::rest::StdAllocator</code>	
<b>Description:</b>	Do not call. See detailed API description. This function exists only for the sake of 'noexcept'. never invoked directly.	

**Table 8.277: `ara::rest::StdAllocator::StdAllocator`**

[SWS\_REST\_02254] `ara::rest::StdAllocator::StdAllocator` [Table 8.277 describes the interface `ara::rest::StdAllocator::StdAllocator`.] ([RS\\_CM\\_00300](#))

### 8.27.5 allocate

<b>Service name:</b>	<code>ara::rest::StdAllocator::allocate</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>value_type* ara::rest::StdAllocator&lt; T &gt;::allocate(std::size_t n)</pre>	
<b>Function param:</b>	<code>n</code>	number of bytes to allocator
<b>Return value:</b>	a value of type <code>value_type *</code>	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	<code>ara/rest/allocator.h</code>	
<b>Class:</b>	<code>ara::rest::StdAllocator</code>	
<b>Description:</b>	Allocate	

**Table 8.278: `ara::rest::StdAllocator::allocate`**

[SWS\_REST\_02255] `ara::rest::StdAllocator::allocate` [Table 8.278 describes the interface `ara::rest::StdAllocator::allocate`.] ([RS\\_CM\\_00300](#))

### 8.27.6 deallocate

<b>Service name:</b>	<code>ara::rest::StdAllocator::deallocate</code>	
<b>Type:</b>	Member function	

<b>Syntax:</b>	void ara::rest::StdAllocator< T >::deallocate(value_type *p, std::size_t s)	
<b>Function param:</b>	p	pointer to allocated memory region
<b>Function param:</b>	s	size of allocated memory region
<b>Return value:</b>	None	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/allocator.h	
<b>Class:</b>	ara::rest::StdAllocator	
<b>Description:</b>	Deallocate.	

**Table 8.279: ara::rest::StdAllocator::deallocate**

[SWS\_REST\_02256] `ara::rest::StdAllocator::deallocate` [Table 8.279 describes the interface `ara::rest::StdAllocator::deallocate`.] ([RS\\_CM\\_00300](#))

### 8.27.7 select\_on\_container\_copy\_construction

<b>Service name:</b>	ara::rest::StdAllocator::select_on_container_copy_construction
<b>Type:</b>	Member function
<b>Syntax:</b>	StdAllocator ara::rest::StdAllocator< T >::select_on_container_copy_construction() const
<b>Function param:</b>	None
<b>Return value:</b>	a value of type StdAllocator
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/allocator.h
<b>Class:</b>	ara::rest::StdAllocator
<b>Description:</b>	Returns the allocator to use when a standard container using it is copied. See std::pmr::polymorphic_allocator documentation for details

**Table 8.280: ara::rest::StdAllocator::select\_on\_container\_copy\_construction**

[SWS\_REST\_02257] `ara::rest::StdAllocator::select_on_container_copy_construction` [Table 8.280 describes the interface `ara::rest::StdAllocator::select_on_container_copy_construction`.] ([RS\\_CM\\_00300](#))

### 8.27.8 resource

<b>Service name:</b>	ara::rest::StdAllocator::resource
<b>Type:</b>	Member function
<b>Syntax:</b>	Allocator* ara::rest::StdAllocator< T >::resource() const
<b>Function param:</b>	None
<b>Return value:</b>	a value of type Allocator *
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/allocator.h
<b>Class:</b>	ara::rest::StdAllocator
<b>Description:</b>	Returns the Allocator behind this adapter. See std::pmr::polymorphic_allocator documentation for details

**Table 8.281: ara::rest::StdAllocator::resource**

[SWS\_REST\_02258] **ara::rest::StdAllocator::resource** [Table 8.281 describes the interface [ara::rest::StdAllocator::resource](#).] ([RS\\_CM\\_00300](#))

## 8.28 ara::rest::Uri::Builder

[SWS\_REST\_02259] [ara::rest::Uri::Builder class shall be declared in the ara/rest/uri.h header file:

```
1     class ara::rest::Uri::Builder;
```

] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.1 Builder

<b>Service name:</b>	ara::rest::Uri::Builder::Builder	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ara::rest::Uri::Builder::Builder(Allocator *alloc=GetDefaultAllocator())	
<b>Function param:</b>	alloc	an allocator
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Builder	
<b>Description:</b>	Default-constructs a builder.	

**Table 8.282: ara::rest::Uri::Builder::Builder**

[SWS\_REST\_02260] **ara::rest::Uri::Builder::Builder** [Table 8.282 describes the interface [ara::rest::Uri::Builder::Builder](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.2 Builder

<b>Service name:</b>	ara::rest::Uri::Builder::Builder	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ara::rest::Uri::Builder::Builder(StringView uri, Allocator *alloc=GetDefaultAllocator())	
<b>Function param:</b>	uri	an URI to initialize from
<b>Function param:</b>	alloc	an allocator
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	

<b>Class:</b>	ara::rest::Uri::Builder
<b>Description:</b>	Parses a URI in string format.

**Table 8.283: ara::rest::Uri::Builder::Builder**

[SWS\_REST\_02261] **ara::rest::Uri::Builder::Builder** [Table 8.283 describes the interface `ara::rest::Uri::Builder::Builder`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.3 Builder

<b>Service name:</b>	ara::rest::Uri::Builder::Builder	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>ara::rest::Uri::Builder::Builder(const Uri &amp;uri, Allocator *alloc=GetDefaultAllocator())</code>	
<b>Function param:</b>	uri	an URI to initiazlize from
<b>Function param:</b>	alloc	an allocator
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Builder	
<b>Description:</b>	Initializes this builder with an existing Uri.	

**Table 8.284: ara::rest::Uri::Builder::Builder**

[SWS\_REST\_02262] **ara::rest::Uri::Builder::Builder** [Table 8.284 describes the interface `ara::rest::Uri::Builder::Builder`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.4 Builder

<b>Service name:</b>	ara::rest::Uri::Builder::Builder	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<code>ara::rest::Uri::Builder::Builder(Uri &amp;&amp;uri, Allocator *alloc=GetDefaultAllocator())</code>	
<b>Function param:</b>	uri	an URI to initiazlize from
<b>Function param:</b>	alloc	an allocator
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Builder	
<b>Description:</b>	Initializes this builder with an existing Uri.	

**Table 8.285: ara::rest::Uri::Builder::Builder**

[SWS\_REST\_02263] **ara::rest::Uri::Builder::Builder** [Table 8.285 describes the interface [ara::rest::Uri::Builder::Builder](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.5 Scheme

<b>Service name:</b>	ara::rest::Uri::Builder::Scheme	
<b>Type:</b>	Member function	
<b>Syntax:</b>	template <typename T > Builder& ara::rest::Uri::Builder::Scheme(T &&value)	
<b>Function param:</b>	value	a value of an output-streamable type
<b>Return value:</b>	a reference to this builder	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Builder	
<b>Description:</b>	Set scheme by parsing the given argument Throws std::invalid_argument if parsing fails.	

**Table 8.286: ara::rest::Uri::Builder::Scheme**

[SWS\_REST\_02264] **ara::rest::Uri::Builder::Scheme** [Table 8.286 describes the interface [ara::rest::Uri::Builder::Scheme](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.6 UserInfo

<b>Service name:</b>	ara::rest::Uri::Builder::UserInfo	
<b>Type:</b>	Member function	
<b>Syntax:</b>	template <typename T > Builder& ara::rest::Uri::Builder::UserInfo(T &&value)	
<b>Function param:</b>	value	a value of an output-streamable type
<b>Return value:</b>	a reference to this builder	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Builder	
<b>Description:</b>	Set user info by parsing the given argument Throws std::invalid_argument if parsing fails.	

**Table 8.287: ara::rest::Uri::Builder::UserInfo**

[SWS\_REST\_02265] **ara::rest::Uri::Builder::UserInfo** [Table 8.287 describes the interface [ara::rest::Uri::Builder::UserInfo](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.7 Host

<b>Service name:</b>	ara::rest::Uri::Builder::Host	
----------------------	-------------------------------	--

<b>Type:</b>	Member function
<b>Syntax:</b>	template <typename T > Builder& ara::rest::Uri::Builder::Host(T &&value)
<b>Function param:</b>	value   a value of an output-streamable type
<b>Return value:</b>	a reference to this builder
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri::Builder
<b>Description:</b>	Sets host by parsing the given argument Throws std::invalid_argument if parsing fails.

**Table 8.288: ara::rest::Uri::Builder::Host**

[SWS\_REST\_02266] [ara::rest::Uri::Builder::Host](#) [Table 8.288 describes the interface [ara::rest::Uri::Builder::Host](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

## 8.28.8 Port

<b>Service name:</b>	ara::rest::Uri::Builder::Port
<b>Type:</b>	Member function
<b>Syntax:</b>	template <typename T > Builder& ara::rest::Uri::Builder::Port(T &&value)
<b>Function param:</b>	value   a value of an output-streamable type
<b>Return value:</b>	a reference to this builder
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri::Builder
<b>Description:</b>	Sets the the Uri port from the given argument Throws std::invalid_argument if parsing fails.

**Table 8.289: ara::rest::Uri::Builder::Port**

[SWS\_REST\_02267] [ara::rest::Uri::Builder::Port](#) [Table 8.289 describes the interface [ara::rest::Uri::Builder::Port](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

## 8.28.9 Path

<b>Service name:</b>	ara::rest::Uri::Builder::Path
<b>Type:</b>	Member function
<b>Syntax:</b>	Builder& ara::rest::Uri::Builder::Path(StringView value)
<b>Function param:</b>	value   a value of an output-streamable type
<b>Return value:</b>	a reference to this builder
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri::Builder
<b>Description:</b>	Sets the URI path by parsing the given argument Throws std::invalid_argument, if parsing fails.

**Table 8.290: ara::rest::Uri::Builder::Path**

[SWS\_REST\_02268] **ara::rest::Uri::Builder::Path** [Table 8.290 describes the interface `ara::rest::Uri::Builder::Path`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.10 Path

<b>Service name:</b>	ara::rest::Uri::Builder::Path	
<b>Type:</b>	Member function	
<b>Syntax:</b>	Builder& ara::rest::Uri::Builder::Path(const Uri::Path &value)	
<b>Function param:</b>	value	a value of an output-streamable type
<b>Return value:</b>	a reference to this builder	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Builder	
<b>Description:</b>	Sets a path from an existing Uri::Path components Throws std::invalid_argument if parsing fails.	

**Table 8.291: ara::rest::Uri::Builder::Path**

[SWS\_REST\_02269] **ara::rest::Uri::Builder::Path** [Table 8.291 describes the interface `ara::rest::Uri::Builder::Path`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.11 PathSegment

<b>Service name:</b>	ara::rest::Uri::Builder::PathSegment	
<b>Type:</b>	Member function	
<b>Syntax:</b>	template <typename T > Builder& ara::rest::Uri::Builder::PathSegment(T &&seg)	
<b>Function param:</b>	seg	of a path segment
<b>Return value:</b>	a reference to this builder	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Builder	
<b>Description:</b>	Inserts a path segment to the end of the path.	

**Table 8.292: ara::rest::Uri::Builder::PathSegment**

[SWS\_REST\_02425] **ara::rest::Uri::Builder::PathSegment** [Table 8.292 describes the interface `ara::rest::Uri::Builder::PathSegment`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.12 PathSegments

<b>Service name:</b>	ara::rest::Uri::Builder::PathSegments	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>template &lt;typename... Ts&gt; Builder&amp; ara::rest::Uri::Builder::PathSegments(Ts &amp;&amp;...values)</pre>	
<b>Function param:</b>	values	values of output-streamable types
<b>Return value:</b>	a reference to this builder	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Builder	
<b>Description:</b>	Constructs a path from the given function arguments Throws std::invalid_argument if parsing fails.	

**Table 8.293: ara::rest::Uri::Builder::PathSegments**

[SWS\_REST\_02270] **ara::rest::Uri::Builder::PathSegments** [Table 8.293 describes the interface `ara::rest::Uri::Builder::PathSegments`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.13 PathSegmentsFrom

<b>Service name:</b>	ara::rest::Uri::Builder::PathSegmentsFrom	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>template &lt;typename... Ts&gt; Builder&amp; ara::rest::Uri::Builder::PathSegmentsFrom(std::size_t pos, Ts &amp;&amp;...values)</pre>	
<b>Function param:</b>	pos	index to start from
<b>Function param:</b>	values	values of output-streamable type
<b>Return value:</b>	a reference to this builder	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Builder	
<b>Description:</b>	Constructs a path from the given function arguments starting from the n'th path segment.	

**Table 8.294: ara::rest::Uri::Builder::PathSegmentsFrom**

[SWS\_REST\_02271] **ara::rest::Uri::Builder::PathSegmentsFrom** [Table 8.294 describes the interface `ara::rest::Uri::Builder::PathSegmentsFrom`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.14 PathSegmentAt

<b>Service name:</b>	ara::rest::Uri::Builder::PathSegmentAt
<b>Type:</b>	Member function



<b>Syntax:</b>	template <typename T , typename U > Builder& ara::rest::Uri::Builder::PathSegmentAt (T &&oldseg, U &&newseg)	
<b>Function param:</b>	oldseg	replaced segment
<b>Function param:</b>	newseg	replacing segment
<b>Return value:</b>	a reference to this builder	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Builder	
<b>Description:</b>	Replaces an existing path segment. If old exists, then it is replaced by new. Otherwise no action is performed.	

**Table 8.295: ara::rest::Uri::Builder::PathSegmentAt**

[SWS\_REST\_02426] [ara::rest::Uri::Builder::PathSegmentAt](#) [Table 8.295 describes the interface [ara::rest::Uri::Builder::PathSegmentAt.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.15 PathSegmentAt

<b>Service name:</b>	ara::rest::Uri::Builder::PathSegmentAt	
<b>Type:</b>	Member function	
<b>Syntax:</b>	template <typename T > Builder& ara::rest::Uri::Builder::PathSegmentAt (T &&seg)	
<b>Function param:</b>	seg	segment to remove
<b>Return value:</b>	a reference to this builder	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Builder	
<b>Description:</b>	Removes a path segment. If segment exists, removes it. Otherwise no action is performed.	

**Table 8.296: ara::rest::Uri::Builder::PathSegmentAt**

[SWS\_REST\_02427] [ara::rest::Uri::Builder::PathSegmentAt](#) [Table 8.296 describes the interface [ara::rest::Uri::Builder::PathSegmentAt.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.16 Query

<b>Service name:</b>	ara::rest::Uri::Builder::Query	
<b>Type:</b>	Member function	
<b>Syntax:</b>	Builder& ara::rest::Uri::Builder::Query (const Uri::Query &q)	
<b>Function param:</b>	q	a query
<b>Return value:</b>	a reference to this builder	

<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri::Builder
<b>Description:</b>	Needs documentation.

**Table 8.297: ara::rest::Uri::Builder::Query**

[SWS\_REST\_02272] **ara::rest::Uri::Builder::Query** [Table 8.297 describes the interface `ara::rest::Uri::Builder::Query`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.17 QueryParameter

<b>Service name:</b>	ara::rest::Uri::Builder::QueryParameter	
<b>Type:</b>	Member function	
<b>Syntax:</b>	template <typename T > Builder& ara::rest::Uri::Builder::QueryParameter (T &&key)	
<b>Function param:</b>	key	of a query parameter
<b>Return value:</b>	a reference to this builder	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Builder	
<b>Description:</b>	Inserts a query parameter (key only) If the given key already exists, no action is performed.	

**Table 8.298: ara::rest::Uri::Builder::QueryParameter**

[SWS\_REST\_02273] **ara::rest::Uri::Builder::QueryParameter** [Table 8.298 describes the interface `ara::rest::Uri::Builder::QueryParameter`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.18 QueryParameter

<b>Service name:</b>	ara::rest::Uri::Builder::QueryParameter	
<b>Type:</b>	Member function	
<b>Syntax:</b>	template <typename T , typename U > Builder& ara::rest::Uri::Builder::QueryParameter (T &&key, U &&value)	
<b>Function param:</b>	key	a key
<b>Function param:</b>	value	a value
<b>Return value:</b>	a reference to this builder	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Builder	
<b>Description:</b>	Inserts a query parameter (key and value) If the given key already exists, no action is performed.	

**Table 8.299: ara::rest::Uri::Builder::QueryParameter**

[SWS\_REST\_02274] `ara::rest::Uri::Builder::QueryParameter` [Table 8.299 describes the interface `ara::rest::Uri::Builder::QueryParameter.`] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.19 QueryParameterAt

<b>Service name:</b>	<code>ara::rest::Uri::Builder::QueryParameterAt</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	template <typename T , typename U > <code>Builder&amp; ara::rest::Uri::Builder::QueryParameterAt (T &amp;&amp;oldkey, U &amp;&amp;newkey)</code>	
<b>Function param:</b>	<code>oldkey</code>	a key
<b>Function param:</b>	<code>newkey</code>	a value
<b>Return value:</b>	a reference to this builder	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	<code>ara/rest/uri.h</code>	
<b>Class:</b>	<code>ara::rest::Uri::Builder</code>	
<b>Description:</b>	Replaces an existing paramater (key only) If old exists, then it is replaced by new. Otherwise no action is performed. If the existing key is part of a key/value pair, the entire pair is replaced.	

Table 8.300: `ara::rest::Uri::Builder::QueryParameterAt`

[SWS\_REST\_02275] `ara::rest::Uri::Builder::QueryParameterAt` [Table 8.300 describes the interface `ara::rest::Uri::Builder::QueryParameterAt.`] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.20 QueryParameterAt

<b>Service name:</b>	<code>ara::rest::Uri::Builder::QueryParameterAt</code>	
<b>Type:</b>	Member function	
<b>Syntax:</b>	template <typename T > <code>Builder&amp; ara::rest::Uri::Builder::QueryParameterAt (T &amp;&amp;key)</code>	
<b>Function param:</b>	<code>key</code>	a key
<b>Return value:</b>	a reference to this builder	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	<code>ara/rest/uri.h</code>	
<b>Class:</b>	<code>ara::rest::Uri::Builder</code>	
<b>Description:</b>	Removes a query parameter (by key) If key exists, removes it. Otherwise no action is performed. If key belong to a key/value pair, the pair is removed. Throws <code>std::invalid_argument</code> if parsing fails.	

Table 8.301: `ara::rest::Uri::Builder::QueryParameterAt`

[SWS\_REST\_02276] `ara::rest::Uri::Builder::QueryParameterAt` [Table 8.301 describes the interface `ara::rest::Uri::Builder::QueryParameterAt.`] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.21 QueryParameterAt

<b>Service name:</b>	ara::rest::Uri::Builder::QueryParameterAt	
<b>Type:</b>	Member function	
<b>Syntax:</b>	template <typename T , typename U , typename V > Builder& ara::rest::Uri::Builder::QueryParameterAt (T &&oldkey, U &&newkey, V &&newvalue)	
<b>Function param:</b>	oldkey	a key
<b>Function param:</b>	newkey	a value
<b>Function param:</b>	newvalue	a value
<b>Return value:</b>	a reference to this builder	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Builder	
<b>Description:</b>	Replaces an existing paramater (key + value) If oldkey exists, then it is replaced (including its value) by newkey and newvalue. Otherwise no action is performed. If no old value exists, it is set.	

**Table 8.302: ara::rest::Uri::Builder::QueryParameterAt**

[SWS\_REST\_02277] [ara::rest::Uri::Builder::QueryParameterAt](#) [Table 8.302 describes the interface [ara::rest::Uri::Builder::QueryParameterAt.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.22 Fragment

<b>Service name:</b>	ara::rest::Uri::Builder::Fragment	
<b>Type:</b>	Member function	
<b>Syntax:</b>	template <typename T > Builder& ara::rest::Uri::Builder::Fragment (T &&value)	
<b>Function param:</b>	value	a value of an output-streamable type
<b>Return value:</b>	a reference to this builder	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Builder	
<b>Description:</b>	Sets the fragment component of a URI Throws std::invalid_argument if parsing fails.	

**Table 8.303: ara::rest::Uri::Builder::Fragment**

[SWS\_REST\_02278] [ara::rest::Uri::Builder::Fragment](#) [Table 8.303 describes the interface [ara::rest::Uri::Builder::Fragment.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.23 Fragment

<b>Service name:</b>	ara::rest::Uri::Builder::Fragment
<b>Type:</b>	Member function

<b>Syntax:</b>	<code>Builder&amp; ara::rest::Uri::Builder::Fragment()</code>
<b>Function param:</b>	None
<b>Return value:</b>	a reference to this builder
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/uri.h</code>
<b>Class:</b>	<code>ara::rest::Uri::Builder</code>
<b>Description:</b>	Clears the fragment component.

**Table 8.304: `ara::rest::Uri::Builder::Fragment`**

[SWS\_REST\_02279] `ara::rest::Uri::Builder::Fragment` [Table 8.304 describes the interface `ara::rest::Uri::Builder::Fragment`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.24 ToUri

<b>Service name:</b>	<code>ara::rest::Uri::Builder::ToUri</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Uri ara::rest::Uri::Builder::ToUri() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a value of type <code>Uri</code>
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/uri.h</code>
<b>Class:</b>	<code>ara::rest::Uri::Builder</code>
<b>Description:</b>	Returns a <code>Uri</code> .

**Table 8.305: `ara::rest::Uri::Builder::ToUri`**

[SWS\_REST\_02280] `ara::rest::Uri::Builder::ToUri` [Table 8.305 describes the interface `ara::rest::Uri::Builder::ToUri`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.28.25 ToPath

<b>Service name:</b>	<code>ara::rest::Uri::Builder::ToPath</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>Uri::Path ara::rest::Uri::Builder::ToPath() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a value of type <code>Uri::Path</code>
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	<code>ara/rest/uri.h</code>
<b>Class:</b>	<code>ara::rest::Uri::Builder</code>
<b>Description:</b>	Returns a <code>Uri</code> path.

**Table 8.306: `ara::rest::Uri::Builder::ToPath`**

[SWS\_REST\_02422] `ara::rest::Uri::Builder::ToPath` [Table 8.306 describes the interface `ara::rest::Uri::Builder::ToPath`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

## 8.28.26 ToQuery

<b>Service name:</b>	ara::rest::Uri::Builder::ToQuery
<b>Type:</b>	Member function
<b>Syntax:</b>	Uri::Query ara::rest::Uri::Builder::ToQuery() const
<b>Function param:</b>	None
<b>Return value:</b>	a value of type Uri::Query
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri::Builder
<b>Description:</b>	Returns a Uri query.

**Table 8.307: ara::rest::Uri::Builder::ToQuery**

[SWS\_REST\_02424] ara::rest::Uri::Builder::ToQuery [Table 8.307 describes the interface ara::rest::Uri::Builder::ToQuery.] (RS\_CM\_00300, RS\_CM\_00304)

## 8.29 ara::rest::Uri::Path::Segment

[SWS\_REST\_02281] [ara::rest::Uri::Path::Segment class shall be declared in the ara/rest/uri.h header file:

```
1     class ara::rest::Uri::Path::Segment;
```

](RS\_CM\_00300, RS\_CM\_00304)

### 8.29.1 Get

<b>Service name:</b>	ara::rest::Uri::Path::Segment::Get
<b>Type:</b>	Member function
<b>Syntax:</b>	StringView ara::rest::Uri::Path::Segment::Get()
<b>Function param:</b>	None
<b>Return value:</b>	a string representation of this path segment
<b>Exceptions:</b>	noexcept (std::is_nothrow_constructible< StringView >::value)
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri::Path::Segment
<b>Description:</b>	Returns a string representation of this path segment. The representation is non-percent-encoded

**Table 8.308: ara::rest::Uri::Path::Segment::Get**

[SWS\_REST\_02282] ara::rest::Uri::Path::Segment::Get [Table 8.308 describes the interface ara::rest::Uri::Path::Segment::Get.] (RS\_CM\_00300, RS\_CM\_00304)

## 8.29.2 GetAs

<b>Service name:</b>	ara::rest::Uri::Path::Segment::GetAs	
<b>Type:</b>	Member function template	
<b>Syntax:</b>	template <typename T > T ara::rest::Uri::Path::Segment::GetAs (T &&def={}) const	
<b>Function param:</b>	def	a default value
<b>Return value:</b>	an instance of type T that represents this URI component.	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Path::Segment	
<b>Description:</b>	Returns this segment converted to a user-defined type. The conversion result is assigned to the function argument which is subsequently returned. If conversion fails the function argument is returned unchanged. So either form is valid: GetAs<string>(), GetAs(string{my_allocator}), GetAs<string>("conversion failed")	

**Table 8.309: ara::rest::Uri::Path::Segment::GetAs**

[SWS\_REST\_02283] **ara::rest::Uri::Path::Segment::GetAs** [Table 8.309 describes the interface `ara::rest::Uri::Path::Segment::GetAs`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

## 8.29.3 operator==

<b>Service name:</b>	ara::rest::Uri::Path::Segment::operator==	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	friend bool operator==(const Segment &a, const Segment &b)	
<b>Function param:</b>	a	object to compare
<b>Function param:</b>	b	object to compare
<b>Return value:</b>	true if segments compare equal	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/uri.h	
<b>Namespace:</b>	ara::rest::Uri::Path::Segment	
<b>Description:</b>	Tests two segments for equality.	

**Table 8.310: ara::rest::Uri::Path::Segment::operator==**

[SWS\_REST\_02284] **ara::rest::Uri::Path::Segment::operator==** [Table 8.310 describes the interface `ara::rest::Uri::Path::Segment::operator==`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

## 8.29.4 operator!=

<b>Service name:</b>	ara::rest::Uri::Path::Segment::operator!=	
<b>Type:</b>	Non-member function	

<b>Syntax:</b>	friend bool operator!=(const Segment &a, const Segment &b)	
<b>Function param:</b>	a	object to compare
<b>Function param:</b>	b	object to compare
<b>Return value:</b>	true if segments compare unequal	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/uri.h	
<b>Namespace:</b>	ara::rest::Uri::Path::Segment	
<b>Description:</b>	Tests two segments for inequality.	

**Table 8.311: ara::rest::Uri::Path::Segment::operator!=**

[SWS\_REST\_02285] **ara::rest::Uri::Path::Segment::operator!=** [Table 8.311 describes the interface `ara::rest::Uri::Path::Segment::operator!=.`] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.29.5 operator<

<b>Service name:</b>	ara::rest::Uri::Path::Segment::operator<	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	friend bool operator<(const Segment &a, const Segment &b)	
<b>Function param:</b>	a	object to compare
<b>Function param:</b>	b	object to compare
<b>Return value:</b>	true if a compares less-than b	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/uri.h	
<b>Namespace:</b>	ara::rest::Uri::Path::Segment	
<b>Description:</b>	Compares two path segments according to their lexicographical order.	

**Table 8.312: ara::rest::Uri::Path::Segment::operator<**

[SWS\_REST\_02286] **ara::rest::Uri::Path::Segment::operator<** [Table 8.312 describes the interface `ara::rest::Uri::Path::Segment::operator<.`] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

## 8.30 ara::rest::Uri::Path

[SWS\_REST\_02287] [ara::rest::Uri::Path class shall be declared in the `ara/rest/uri.h` header file:

```
1     class ara::rest::Uri::Path;
```

] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))



### 8.30.1 IteratorRange

<b>Name:</b>	IteratorRange
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::Uri::Path::IteratorRange = ara::rest::IteratorRange<unspecified_iterator_type>
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri::Path
<b>Description:</b>	Iterator range of path segments.

**Table 8.313: ara::rest::Uri::Path::IteratorRange**

[SWS\_REST\_02288] **IteratorRange** [Table 8.313 describes the type alias `ara::rest::Uri::Path::IteratorRange`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.30.2 NumSegments

<b>Service name:</b>	ara::rest::Uri::Path::NumSegments
<b>Type:</b>	Member function
<b>Syntax:</b>	std::size_t ara::rest::Uri::Path::NumSegments() const
<b>Function param:</b>	None
<b>Return value:</b>	a number of path segments
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri::Path
<b>Description:</b>	Returns the number of path segments.

**Table 8.314: ara::rest::Uri::Path::NumSegments**

[SWS\_REST\_02289] **ara::rest::Uri::Path::NumSegments** [Table 8.314 describes the interface `ara::rest::Uri::Path::NumSegments`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.30.3 GetSegments

<b>Service name:</b>	ara::rest::Uri::Path::GetSegments
<b>Type:</b>	Member function
<b>Syntax:</b>	IteratorRange ara::rest::Uri::Path::GetSegments() const
<b>Function param:</b>	None
<b>Return value:</b>	an iterator range of path segments
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri::Path
<b>Description:</b>	Returns a range of path segments.

**Table 8.315: ara::rest::Uri::Path::GetSegments**

[SWS\_REST\_02290] **ara::rest::Uri::Path::GetSegments** [Table 8.315 describes the interface `ara::rest::Uri::Path::GetSegments`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.30.4 operator==

<b>Service name:</b>	ara::rest::Uri::Path::operator==	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	friend bool operator==(const Path &a, const Path &b)	
<b>Function param:</b>	a	object to compare
<b>Function param:</b>	b	object to compare
<b>Return value:</b>	true if equal	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/uri.h	
<b>Namespace:</b>	ara::rest::Uri::Path	
<b>Description:</b>	Tests two paths for equality.	

**Table 8.316: ara::rest::Uri::Path::operator==**

[SWS\_REST\_02291] **ara::rest::Uri::Path::operator==** [Table 8.316 describes the interface `ara::rest::Uri::Path::operator==`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.30.5 operator!=

<b>Service name:</b>	ara::rest::Uri::Path::operator!=	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	friend bool operator!=(const Path &a, const Path &b)	
<b>Function param:</b>	a	object to compare
<b>Function param:</b>	b	object to compare
<b>Return value:</b>	true if not equal	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/uri.h	
<b>Namespace:</b>	ara::rest::Uri::Path	
<b>Description:</b>	Tests two paths for inequality.	

**Table 8.317: ara::rest::Uri::Path::operator!=**

[SWS\_REST\_02292] **ara::rest::Uri::Path::operator!=** [Table 8.317 describes the interface `ara::rest::Uri::Path::operator!=`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.30.6 operator<

<b>Service name:</b>	ara::rest::Uri::Path::operator<	
----------------------	---------------------------------	--

<b>Type:</b>	Non-member function	
<b>Syntax:</b>	friend bool operator<(const Path &a, const Path &b)	
<b>Function param:</b>	a	object to compare
<b>Function param:</b>	b	object to compare
<b>Return value:</b>	true if a compares less-than b	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/uri.h	
<b>Namespace:</b>	ara::rest::Uri::Path	
<b>Description:</b>	Relates two paths according to their lexicographical order.	

**Table 8.318: ara::rest::Uri::Path::operator<**

[SWS\_REST\_02293] **ara::rest::Uri::Path::operator<** [Table 8.318 describes the interface `ara::rest::Uri::Path::operator<`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

## 8.31 ara::rest::Uri::Query::Parameter

[SWS\_REST\_02294] [ara::rest::Uri::Query::Parameter class shall be declared in the `ara/rest/uri.h` header file:

```
1     class ara::rest::Uri::Query::Parameter;
```

] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.31.1 GetKey

<b>Service name:</b>	ara::rest::Uri::Query::Parameter::GetKey
<b>Type:</b>	Member function
<b>Syntax:</b>	StringView ara::rest::Uri::Query::Parameter::GetKey() const
<b>Function param:</b>	None
<b>Return value:</b>	a string representation
<b>Exceptions:</b>	noexcept (std::is_nothrow_constructible< StringView >::value)
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri::Query::Parameter
<b>Description:</b>	Returns a string representation of the parameter key The representation is non-percent-encoded.

**Table 8.319: ara::rest::Uri::Query::Parameter::GetKey**

[SWS\_REST\_02295] **ara::rest::Uri::Query::Parameter::GetKey** [Table 8.319 describes the interface `ara::rest::Uri::Query::Parameter::GetKey`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.31.2 GetKeyAs

<b>Service name:</b>	ara::rest::Uri::Query::Parameter::GetKeyAs	
<b>Type:</b>	Member function template	
<b>Syntax:</b>	template <typename T > T ara::rest::Uri::Query::Parameter::GetKeyAs (T &&def={}) const	
<b>Function param:</b>	def	a default value
<b>Return value:</b>	an instance of type T that represents this URI component.	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Query::Parameter	
<b>Description:</b>	Converts a query parameter key to the specified type. The conversion result is assigned to the function argument which is subsequently returned. If conversion fails the function argument is returned unchanged. So either form is valid: GetAs<string>(), GetAs(string{my_allocator}), GetAs<string>("conversion failed")	

**Table 8.320: ara::rest::Uri::Query::Parameter::GetKeyAs**

[SWS\_REST\_02296] [ara::rest::Uri::Query::Parameter::GetKeyAs](#) [Table 8.320 describes the interface [ara::rest::Uri::Query::Parameter::GetKeyAs.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.31.3 HasValue

<b>Service name:</b>	ara::rest::Uri::Query::Parameter::HasValue	
<b>Type:</b>	Member function	
<b>Syntax:</b>	bool ara::rest::Uri::Query::Parameter::HasValue () const	
<b>Function param:</b>	None	
<b>Return value:</b>	true if this query paramater has a value component	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Query::Parameter	
<b>Description:</b>	Needs documentation.	

**Table 8.321: ara::rest::Uri::Query::Parameter::HasValue**

[SWS\_REST\_02297] [ara::rest::Uri::Query::Parameter::HasValue](#) [Table 8.321 describes the interface [ara::rest::Uri::Query::Parameter::HasValue.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.31.4 GetValue

<b>Service name:</b>	ara::rest::Uri::Query::Parameter::GetValue	
<b>Type:</b>	Member function	

<b>Syntax:</b>	StringView ara::rest::Uri::Query::Parameter::GetValue() const
<b>Function param:</b>	None
<b>Return value:</b>	a string representation of the value
<b>Exceptions:</b>	noexcept (std::is_nothrow_constructible< StringView >::value)
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri::Query::Parameter
<b>Description:</b>	Obtains the value of a query parameter If none exists the result is undefined.

**Table 8.322: ara::rest::Uri::Query::Parameter::GetValue**

[SWS\_REST\_02298] [ara::rest::Uri::Query::Parameter::GetValue](#) [Table 8.322 describes the interface [ara::rest::Uri::Query::Parameter::GetValue.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.31.5 GetValueAs

<b>Service name:</b>	ara::rest::Uri::Query::Parameter::GetValueAs
<b>Type:</b>	Member function template
<b>Syntax:</b>	template <typename T > T ara::rest::Uri::Query::Parameter::GetValueAs (T &&def={}) const
<b>Function param:</b>	def   a default value
<b>Return value:</b>	an instance of type T that represents this URI component.
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri::Query::Parameter
<b>Description:</b>	Converts a query parameter value to the specified type. The conversion result is assigned to the function argument which is subsequently returned. If conversion fails the function argument is returned unchanged. So either form is valid: <code>GetAs&lt;string&gt;()</code> , <code>GetAs(string{my_allocator})</code> , <code>GetAs&lt;string&gt;("conversion failed")</code>

**Table 8.323: ara::rest::Uri::Query::Parameter::GetValueAs**

[SWS\_REST\_02299] [ara::rest::Uri::Query::Parameter::GetValueAs](#) [Table 8.323 describes the interface [ara::rest::Uri::Query::Parameter::GetValueAs.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

## 8.32 ara::rest::Uri::Query

[SWS\_REST\_02300] [ara::rest::Uri::Query class shall be declared in the ara/rest/uri.h header file:

```
1     class ara::rest::Uri::Query;
```

|(RS\_CM\_00300, RS\_CM\_00304)

### 8.32.1 IteratorRange

<b>Name:</b>	IteratorRange
<b>Type:</b>	Member type alias
<b>Syntax:</b>	using ara::rest::Uri::Query::IteratorRange = ara::rest::IteratorRange<unspecified_iterator_type>
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri::Query
<b>Description:</b>	A range of query parameters.

**Table 8.324: ara::rest::Uri::Query::IteratorRange**

[SWS\_REST\_02301] **IteratorRange** [Table 8.324 describes the type alias `ara::rest::Uri::Query::IteratorRange`.](RS\_CM\_00300, RS\_CM\_00304)

### 8.32.2 NumParameters

<b>Service name:</b>	ara::rest::Uri::Query::NumParameters
<b>Type:</b>	Member function
<b>Syntax:</b>	std::size_t ara::rest::Uri::Query::NumParameters() const
<b>Function param:</b>	None
<b>Return value:</b>	the number of query parameters
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri::Query
<b>Description:</b>	Returns the number of query parameters.

**Table 8.325: ara::rest::Uri::Query::NumParameters**

[SWS\_REST\_02302] **ara::rest::Uri::Query::NumParameters** [Table 8.325 describes the interface `ara::rest::Uri::Query::NumParameters`.](RS\_CM\_00300, RS\_CM\_00304)

### 8.32.3 GetParameters

<b>Service name:</b>	ara::rest::Uri::Query::GetParameters
<b>Type:</b>	Member function
<b>Syntax:</b>	IteratorRange ara::rest::Uri::Query::GetParameters() const
<b>Function param:</b>	None
<b>Return value:</b>	an iterator range of query parameters
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/uri.h

<b>Class:</b>	ara::rest::Uri::Query
<b>Description:</b>	Returns the range of all query parameters.

**Table 8.326: ara::rest::Uri::Query::GetParameters**

[SWS\_REST\_02303] **ara::rest::Uri::Query::GetParameters** [Table 8.326 describes the interface `ara::rest::Uri::Query::GetParameters`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.32.4 GetParameter

<b>Service name:</b>	ara::rest::Uri::Query::GetParameter	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>const Parameter&amp; ara::rest::Uri::Query::GetParameter(std::size_t i) const</pre>	
<b>Function param:</b>	i	an index
<b>Return value:</b>	a reference to the query parameter	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Query	
<b>Description:</b>	Returns a specific query parameter by index.	

**Table 8.327: ara::rest::Uri::Query::GetParameter**

[SWS\_REST\_02304] **ara::rest::Uri::Query::GetParameter** [Table 8.327 describes the interface `ara::rest::Uri::Query::GetParameter`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.32.5 Find

<b>Service name:</b>	ara::rest::Uri::Query::Find	
<b>Type:</b>	Member function	
<b>Syntax:</b>	<pre>IteratorRange::Iterator ara::rest::Uri::Query::Find(StringView key) const</pre>	
<b>Function param:</b>	key	a key
<b>Return value:</b>	an iterator to the respective query parameter	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Query	
<b>Description:</b>	Searches for a query parameter by key.	

**Table 8.328: ara::rest::Uri::Query::Find**

[SWS\_REST\_02305] **ara::rest::Uri::Query::Find** [Table 8.328 describes the interface `ara::rest::Uri::Query::Find`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.32.6 HasKey

<b>Service name:</b>	ara::rest::Uri::Query::HasKey	
<b>Type:</b>	Member function	
<b>Syntax:</b>	bool ara::rest::Uri::Query::HasKey(StringView key)	
<b>Function param:</b>	key	a key
<b>Return value:</b>	true if key exists	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/uri.h	
<b>Class:</b>	ara::rest::Uri::Query	
<b>Description:</b>	Tests whether a query parameter of a given key exists.	

**Table 8.329: ara::rest::Uri::Query::HasKey**

[SWS\_REST\_02306] `ara::rest::Uri::Query::HasKey` [Table 8.329 describes the interface `ara::rest::Uri::Query::HasKey`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

## 8.33 ara::rest::Uri

[SWS\_REST\_02307] [code `ara::rest::Uri` class shall be declared in the `ara/rest/uri.h` header file:

```
1     class ara::rest::Uri;
```

] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.1 Part

<b>Name:</b>	Part	
<b>Type:</b>	Member enumeration	
<b>Range:</b>	kScheme	= 1 « 1
	kUserInfo	= 1 « 2
	kHost	= 1 « 3
	kPort	= 1 « 4
	kPath	= 1 « 5
	kQuery	= 1 « 6
	kFragment	= 1 « 7
	kPathAndQuery	= Part::kPath   Part::kQuery
	kPathEtc	= Part::kPath   Part::kQuery   Part::kFragment
	kAll	= ~std::underlying_type<Part>::type{0}



<b>Syntax:</b>	<pre>enum class Part : std::uint32_t {     kScheme = 1 « 1,     kUserInfo = 1 « 2,     kHost = 1 « 3,     kPort = 1 « 4,     kPath = 1 « 5,     kQuery = 1 « 6,     kFragment = 1 « 7,     kPathAndQuery = Part::kPath  Part::kQuery,     kPathEtc = Part::kPath  Part::kQuery      Part::kFragment,     kAll = ~std::underlying_type&lt;Part&gt;::type{0} };</pre>
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri
<b>Description:</b>	Used to specify a subset of a URI. Part defines components of a

**Table 8.330: ara::rest::Uri::Part**

[SWS\_REST\_02308] Part [Table 8.330 describes the enumeration datatype `ara::rest::Uri::Part`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.2 LENGTH\_MAX

<b>Name:</b>	LENGTH_MAX
<b>Type:</b>	Member variable
<b>Syntax:</b>	static constexpr std::size_t LENGTH_MAX = 2048;
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri
<b>Description:</b>	The maximum length of a URI. The suggested length maximum is around 2000 characters which roughly conforms to the typical limit that mainstream webbrowsers support. A bound is specified to enable optimization potential in the internal encoding.

**Table 8.331: ara::rest::Uri::LENGTH\_MAX**

[SWS\_REST\_02309] `ara::rest::Uri::LENGTH_MAX` [Table 8.331 describes the variable `ara::rest::Uri::LENGTH_MAX`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.3 operator|

<b>Service name:</b>	ara::rest::Uri::operator	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	friend constexpr Part operator (Part a, Part b)	
<b>Function param:</b>	a	(set of) Part enumerator(s)
<b>Function param:</b>	b	(set of) Part enumerator(s)
<b>Return value:</b>	a set of Part enumerators	
<b>Exceptions:</b>	noexcept	

<b>Header file:</b>	ara/rest/uri.h
<b>Namespace:</b>	ara::rest::Uri
<b>Description:</b>	Computes a set of Part enumerators.

**Table 8.332: ara::rest::Uri::operator|**

[SWS\_REST\_02310] **ara::rest::Uri::operator|** [Table 8.332 describes the interface `ara::rest::Uri::operator|`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.4 Uri

<b>Service name:</b>	ara::rest::Uri::Uri
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>ara::rest::Uri::Uri() =default</code>
<b>Function param:</b>	None
<b>Return value:</b>	None
<b>Exceptions:</b>	<code>noexcept=default</code>
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri
<b>Description:</b>	Constructs a default URI. A default-constructed URI is empty. To populate an existing URI, <code>Uri::Builder</code> must be used. Uri member functions must not throw unless in those cases where <code>StringView</code> is used and <code>StringView</code> is not <code>'nothrow_constructible'</code> .

**Table 8.333: ara::rest::Uri::Uri**

[SWS\_REST\_02311] **ara::rest::Uri::Uri** [Table 8.333 describes the interface `ara::rest::Uri::Uri`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.5 HasScheme

<b>Service name:</b>	ara::rest::Uri::HasScheme
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>bool ara::rest::Uri::HasScheme() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a value of type <code>bool</code>
<b>Exceptions:</b>	<code>noexcept</code>
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri
<b>Description:</b>	Has scheme.

**Table 8.334: ara::rest::Uri::HasScheme**

[SWS\_REST\_02312] **ara::rest::Uri::HasScheme** [Table 8.334 describes the interface `ara::rest::Uri::HasScheme`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.6 GetScheme

<b>Service name:</b>	ara::rest::Uri::GetScheme
<b>Type:</b>	Member function
<b>Syntax:</b>	StringView ara::rest::Uri::GetScheme() const
<b>Function param:</b>	None
<b>Return value:</b>	a value of type StringView
<b>Exceptions:</b>	noexcept (std::is_nothrow_constructible< StringView >::value)
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri
<b>Description:</b>	gets scheme.

**Table 8.335: ara::rest::Uri::GetScheme**

[SWS\_REST\_02313] **ara::rest::Uri::GetScheme** [Table 8.335 describes the interface `ara::rest::Uri::GetScheme.`] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.7 HasUserInfo

<b>Service name:</b>	ara::rest::Uri::HasUserInfo
<b>Type:</b>	Member function
<b>Syntax:</b>	bool ara::rest::Uri::HasUserInfo() const
<b>Function param:</b>	None
<b>Return value:</b>	a value of type bool
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri
<b>Description:</b>	has user info.

**Table 8.336: ara::rest::Uri::HasUserInfo**

[SWS\_REST\_02314] **ara::rest::Uri::HasUserInfo** [Table 8.336 describes the interface `ara::rest::Uri::HasUserInfo.`] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.8 GetUserinfo

<b>Service name:</b>	ara::rest::Uri::GetUserinfo
<b>Type:</b>	Member function
<b>Syntax:</b>	StringView ara::rest::Uri::GetUserinfo() const
<b>Function param:</b>	None
<b>Return value:</b>	a value of type StringView
<b>Exceptions:</b>	noexcept (std::is_nothrow_constructible< StringView >::value)
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri
<b>Description:</b>	get user info.

**Table 8.337: ara::rest::Uri::GetUserinfo**

[SWS\_REST\_02315] `ara::rest::Uri::GetUserinfo` [Table 8.337 describes the interface `ara::rest::Uri::GetUserinfo`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.9 HasHost

<b>Service name:</b>	<code>ara::rest::Uri::HasHost</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>bool ara::rest::Uri::HasHost() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a value of type <code>bool</code>
<b>Exceptions:</b>	<code>noexcept</code>
<b>Header file:</b>	<code>ara/rest/uri.h</code>
<b>Class:</b>	<code>ara::rest::Uri</code>
<b>Description:</b>	has host.

**Table 8.338: `ara::rest::Uri::HasHost`**

[SWS\_REST\_02316] `ara::rest::Uri::HasHost` [Table 8.338 describes the interface `ara::rest::Uri::HasHost`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.10 GetHost

<b>Service name:</b>	<code>ara::rest::Uri::GetHost</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>StringView ara::rest::Uri::GetHost() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a value of type <code>StringView</code>
<b>Exceptions:</b>	<code>noexcept(std::is_nothrow_constructible&lt;StringView&gt;::value)</code>
<b>Header file:</b>	<code>ara/rest/uri.h</code>
<b>Class:</b>	<code>ara::rest::Uri</code>
<b>Description:</b>	get host.

**Table 8.339: `ara::rest::Uri::GetHost`**

[SWS\_REST\_02317] `ara::rest::Uri::GetHost` [Table 8.339 describes the interface `ara::rest::Uri::GetHost`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.11 HasPort

<b>Service name:</b>	<code>ara::rest::Uri::HasPort</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>bool ara::rest::Uri::HasPort() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a value of type <code>bool</code>
<b>Exceptions:</b>	<code>noexcept</code>

<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri
<b>Description:</b>	has host.

**Table 8.340: ara::rest::Uri::HasPort**

[SWS\_REST\_02318] **ara::rest::Uri::HasPort** [Table 8.340 describes the interface `ara::rest::Uri::HasPort`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.12 GetPort

<b>Service name:</b>	ara::rest::Uri::GetPort
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>int ara::rest::Uri::GetPort() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a value of type <code>int</code>
<b>Exceptions:</b>	<code>noexcept</code>
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri
<b>Description:</b>	get port.

**Table 8.341: ara::rest::Uri::GetPort**

[SWS\_REST\_02319] **ara::rest::Uri::GetPort** [Table 8.341 describes the interface `ara::rest::Uri::GetPort`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.13 HasPath

<b>Service name:</b>	ara::rest::Uri::HasPath
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>bool ara::rest::Uri::HasPath() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a value of type <code>bool</code>
<b>Exceptions:</b>	<code>noexcept</code>
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri
<b>Description:</b>	has path.

**Table 8.342: ara::rest::Uri::HasPath**

[SWS\_REST\_02320] **ara::rest::Uri::HasPath** [Table 8.342 describes the interface `ara::rest::Uri::HasPath`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.14 GetPath

<b>Service name:</b>	ara::rest::Uri::GetPath
<b>Type:</b>	Member function
<b>Syntax:</b>	const Path& ara::rest::Uri::GetPath() const
<b>Function param:</b>	None
<b>Return value:</b>	a value of type const Path &
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri
<b>Description:</b>	get path.

**Table 8.343: ara::rest::Uri::GetPath**

[SWS\_REST\_02321] **ara::rest::Uri::GetPath** [Table 8.343 describes the interface `ara::rest::Uri::GetPath`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.15 HasQuery

<b>Service name:</b>	ara::rest::Uri::HasQuery
<b>Type:</b>	Member function
<b>Syntax:</b>	bool ara::rest::Uri::HasQuery() const
<b>Function param:</b>	None
<b>Return value:</b>	a value of type bool
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri
<b>Description:</b>	has query.

**Table 8.344: ara::rest::Uri::HasQuery**

[SWS\_REST\_02322] **ara::rest::Uri::HasQuery** [Table 8.344 describes the interface `ara::rest::Uri::HasQuery`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.16 GetQuery

<b>Service name:</b>	ara::rest::Uri::GetQuery
<b>Type:</b>	Member function
<b>Syntax:</b>	const Query& ara::rest::Uri::GetQuery() const
<b>Function param:</b>	None
<b>Return value:</b>	a value of type const Query &
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri
<b>Description:</b>	get query.

**Table 8.345: ara::rest::Uri::GetQuery**

[SWS\_REST\_02323] `ara::rest::Uri::GetQuery` [Table 8.345 describes the interface `ara::rest::Uri::GetQuery`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.17 HasFragment

<b>Service name:</b>	<code>ara::rest::Uri::HasFragment</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>bool ara::rest::Uri::HasFragment() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a value of type <code>bool</code>
<b>Exceptions:</b>	<code>noexcept</code>
<b>Header file:</b>	<code>ara/rest/uri.h</code>
<b>Class:</b>	<code>ara::rest::Uri</code>
<b>Description:</b>	Has Fragment.

**Table 8.346: `ara::rest::Uri::HasFragment`**

[SWS\_REST\_02324] `ara::rest::Uri::HasFragment` [Table 8.346 describes the interface `ara::rest::Uri::HasFragment`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.18 GetFragment

<b>Service name:</b>	<code>ara::rest::Uri::GetFragment</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>StringView ara::rest::Uri::GetFragment() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a value of type <code>StringView</code>
<b>Exceptions:</b>	<code>noexcept(std::is_nothrow_constructible&lt; StringView &gt;::value)</code>
<b>Header file:</b>	<code>ara/rest/uri.h</code>
<b>Class:</b>	<code>ara::rest::Uri</code>
<b>Description:</b>	Get Fragment as string.

**Table 8.347: `ara::rest::Uri::GetFragment`**

[SWS\_REST\_02325] `ara::rest::Uri::GetFragment` [Table 8.347 describes the interface `ara::rest::Uri::GetFragment`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.19 GetFragmentAs

<b>Service name:</b>	<code>ara::rest::Uri::GetFragmentAs</code>
<b>Type:</b>	Member function template
<b>Syntax:</b>	<code>template &lt;typename T &gt;</code> <code>T ara::rest::Uri::GetFragmentAs(T &amp;&amp;def={}) const</code>
<b>Function param:</b>	<code>def</code>   a default value
<b>Return value:</b>	an instance of type <code>T</code> that represents this URI component

<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri
<b>Description:</b>	Converts a URI fragment part to a given type. The conversion result is assigned to the function argument which is subsequently returned. If conversion fails the function argument is returned unchanged. So either form is valid: GetFragmentAs<string>(), GetFragmentAs(string{my_allocator}), GetFragmentAs<string>("conversion failed")

**Table 8.348: ara::rest::Uri::GetFragmentAs**

[SWS\_REST\_02326] **ara::rest::Uri::GetFragmentAs** [Table 8.348 describes the interface `ara::rest::Uri::GetFragmentAs`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.20 IsEmpty

<b>Service name:</b>	ara::rest::Uri::IsEmpty
<b>Type:</b>	Member function
<b>Syntax:</b>	bool ara::rest::Uri::IsEmpty() const
<b>Function param:</b>	None
<b>Return value:</b>	true if empty
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri
<b>Description:</b>	Is URI empty.

**Table 8.349: ara::rest::Uri::IsEmpty**

[SWS\_REST\_02327] **ara::rest::Uri::IsEmpty** [Table 8.349 describes the interface `ara::rest::Uri::IsEmpty`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.21 IsRelative

<b>Service name:</b>	ara::rest::Uri::IsRelative
<b>Type:</b>	Member function
<b>Syntax:</b>	bool ara::rest::Uri::IsRelative() const
<b>Function param:</b>	None
<b>Return value:</b>	true if relative
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/uri.h
<b>Class:</b>	ara::rest::Uri
<b>Description:</b>	Is URI relative. An URI is relative if it does not starts with a scheme.

**Table 8.350: ara::rest::Uri::IsRelative**



[SWS\_REST\_02328] `ara::rest::Uri::IsRelative` [Table 8.350 describes the interface `ara::rest::Uri::IsRelative`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.22 IsOpaque

<b>Service name:</b>	<code>ara::rest::Uri::isOpaque</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>bool ara::rest::Uri::isOpaque() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	true if this URI is opaque
<b>Exceptions:</b>	<code>noexcept</code>
<b>Header file:</b>	<code>ara/rest/uri.h</code>
<b>Class:</b>	<code>ara::rest::Uri</code>
<b>Description:</b>	Denotes whether the URI is opaque. An opaque URI is an absolute URI whose scheme-specific part does not begin with a slash character ('/').

**Table 8.351: `ara::rest::Uri::isOpaque`**

[SWS\_REST\_02329] `ara::rest::Uri::isOpaque` [Table 8.351 describes the interface `ara::rest::Uri::isOpaque`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.33.23 IsHierarchical

<b>Service name:</b>	<code>ara::rest::Uri::isHierarchical</code>
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>bool ara::rest::Uri::isHierarchical() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	true if this URI is hierarchical
<b>Exceptions:</b>	<code>noexcept</code>
<b>Header file:</b>	<code>ara/rest/uri.h</code>
<b>Class:</b>	<code>ara::rest::Uri</code>
<b>Description:</b>	Denotes whether this URI is hierarchical. A hierarchical URI is either an absolute URI whose scheme-specific part begins with a slash character, or a relative URI, that is, a URI that does not specify a scheme.

**Table 8.352: `ara::rest::Uri::isHierarchical`**

[SWS\_REST\_02330] `ara::rest::Uri::isHierarchical` [Table 8.352 describes the interface `ara::rest::Uri::isHierarchical`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

## 8.34 `ara::rest::Uuid`

[SWS\_REST\_02331] [`ara::rest::Uuid` class shall be declared in the `ara/rest/uuid.h` header file:

```
1     class ara::rest::Uuid;
```

]([RS\\_CM\\_00300](#))

### 8.34.1 MakeV1

<b>Service name:</b>	ara::rest::Uuid::MakeV1
<b>Type:</b>	Member function
<b>Syntax:</b>	static Uuid ara::rest::Uuid::MakeV1()
<b>Return value:</b>	Generated Uuid
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/uuid.h
<b>Class:</b>	ara::rest::Uuid
<b>Description:</b>	Creates an UUID with version 1 defined in RFC4122 (date-time and MAC address).

**Table 8.353: ara::rest::Uuid::MakeV1**

[SWS\_REST\_02418] **ara::rest::Uuid::MakeV1** [Table 8.353 describes the interface `ara::rest::Uuid::MakeV1`].([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.34.2 MakeV3

<b>Service name:</b>	ara::rest::Uuid::MakeV3
<b>Type:</b>	Member function
<b>Syntax:</b>	static Uuid ara::rest::Uuid::MakeV3(const String& ns)
<b>Function param:</b>	ns   UUID namespace
<b>Return value:</b>	Generated Uuid
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/uuid.h
<b>Class:</b>	ara::rest::Uuid
<b>Description:</b>	Creates an UUID with version 3 defined in RFC4122 (namespace with MD5).

**Table 8.354: ara::rest::Uuid::MakeV3**

[SWS\_REST\_02419] **ara::rest::Uuid::MakeV3** [Table 8.354 describes the interface `ara::rest::Uuid::MakeV3`].([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.34.3 MakeV4

<b>Service name:</b>	ara::rest::Uuid::MakeV4
<b>Type:</b>	Member function
<b>Syntax:</b>	static Uuid ara::rest::Uuid::MakeV4()
<b>Return value:</b>	Generated Uuid
<b>Exceptions:</b>	Implementation-defined

<b>Header file:</b>	ara/rest/uuid.h
<b>Class:</b>	ara::rest::Uuid
<b>Description:</b>	Creates an UUID with version 4 defined in RFC4122 (random).

**Table 8.355: ara::rest::Uuid::MakeV4**

[SWS\_REST\_02420] **ara::rest::Uuid::MakeV4** [Table 8.355 describes the interface `ara::rest::Uuid::MakeV4`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.34.4 MakeV5

<b>Service name:</b>	ara::rest::Uuid::MakeV5
<b>Type:</b>	Member function
<b>Syntax:</b>	static Uuid ara::rest::Uuid::MakeV5(const String& ns)
<b>Function param:</b>	ns   UUID namespace
<b>Return value:</b>	Generated Uuid
<b>Exceptions:</b>	Implementation-defined
<b>Header file:</b>	ara/rest/uuid.h
<b>Class:</b>	ara::rest::Uuid
<b>Description:</b>	Creates an UUID with version 5 defined in RFC4122 (namespace with SHA1).

**Table 8.356: ara::rest::Uuid::MakeV5**

[SWS\_REST\_02421] **ara::rest::Uuid::MakeV5** [Table 8.356 describes the interface `ara::rest::Uuid::MakeV5`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.34.5 Uuid

<b>Service name:</b>	ara::rest::Uuid::Uuid
<b>Type:</b>	Member function
<b>Syntax:</b>	ara::rest::Uuid::Uuid() =default
<b>Function param:</b>	None
<b>Return value:</b>	None
<b>Exceptions:</b>	noexcept=default
<b>Header file:</b>	ara/rest/uuid.h
<b>Class:</b>	ara::rest::Uuid
<b>Description:</b>	Default constructs a Uuid.

**Table 8.357: ara::rest::Uuid::Uuid**

[SWS\_REST\_02332] **ara::rest::Uuid::Uuid** [Table 8.357 describes the interface `ara::rest::Uuid::Uuid`.] ([RS\\_CM\\_00300](#))

### 8.34.6 Uuid

<b>Service name:</b>	ara::rest::Uuid::Uuid	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ara::rest::Uuid::Uuid(StringView id)	
<b>Function param:</b>	id	a UUID in RFC4122 format
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uuid.h	
<b>Class:</b>	ara::rest::Uuid	
<b>Description:</b>	Constructs a Uuid from a string representation. Throws an std::invalid_argument if parsing fails.	

**Table 8.358: ara::rest::Uuid::Uuid**

[SWS\_REST\_02333] **ara::rest::Uuid::Uuid** [Table 8.358 describes the interface `ara::rest::Uuid::Uuid`.] ([RS\\_CM\\_00300](#))

### 8.34.7 Uuid

<b>Service name:</b>	ara::rest::Uuid::Uuid	
<b>Type:</b>	Member function	
<b>Syntax:</b>	ara::rest::Uuid::Uuid(std::uint32_t timeLow, std::uint16_t timeMid, std::uint16_t timeHighAndVersion, std::uint16_t clockSeq, std::uint64_t node)	
<b>Function param:</b>	timeLow	see RFC 4122
<b>Function param:</b>	timeMid	see RFC 4122
<b>Function param:</b>	timeHighAndVersion	see RFC 4122
<b>Function param:</b>	clockSeq	see RFC 4122
<b>Function param:</b>	node	see RFC 4122
<b>Return value:</b>	None	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/uuid.h	
<b>Class:</b>	ara::rest::Uuid	
<b>Description:</b>	Constructs a Uuid from its components explicitly.	

**Table 8.359: ara::rest::Uuid::Uuid**

[SWS\_REST\_02334] **ara::rest::Uuid::Uuid** [Table 8.359 describes the interface `ara::rest::Uuid::Uuid`.] ([RS\\_CM\\_00300](#))

### 8.34.8 GetTimeLow

<b>Service name:</b>	ara::rest::Uuid::GetTimeLow	
<b>Type:</b>	Member function	
<b>Syntax:</b>	std::uint32_t ara::rest::Uuid::GetTimeLow() const	
<b>Function param:</b>	None	

<b>Return value:</b>	a numeric value
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/uuid.h
<b>Class:</b>	ara::rest::Uuid
<b>Description:</b>	Returns time low.

**Table 8.360: ara::rest::Uuid::GetTimeLow**

[SWS\_REST\_02335] `ara::rest::Uuid::GetTimeLow` [Table 8.360 describes the interface `ara::rest::Uuid::GetTimeLow.`] (*RS\_CM\_00300*)

### 8.34.9 GetTimeMid

<b>Service name:</b>	ara::rest::Uuid::GetTimeMid
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>std::uint16_t ara::rest::Uuid::GetTimeMid() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a numeric value
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/uuid.h
<b>Class:</b>	ara::rest::Uuid
<b>Description:</b>	Returns time mid.

**Table 8.361: ara::rest::Uuid::GetTimeMid**

[SWS\_REST\_02336] `ara::rest::Uuid::GetTimeMid` [Table 8.361 describes the interface `ara::rest::Uuid::GetTimeMid.`] (*RS\_CM\_00300*)

### 8.34.10 GetTimeHighAndVersion

<b>Service name:</b>	ara::rest::Uuid::GetTimeHighAndVersion
<b>Type:</b>	Member function
<b>Syntax:</b>	<code>std::uint16_t ara::rest::Uuid::GetTimeHighAndVersion() const</code>
<b>Function param:</b>	None
<b>Return value:</b>	a numeric value
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/uuid.h
<b>Class:</b>	ara::rest::Uuid
<b>Description:</b>	Returns time high and version.

**Table 8.362: ara::rest::Uuid::GetTimeHighAndVersion**

[SWS\_REST\_02337] `ara::rest::Uuid::GetTimeHighAndVersion` [Table 8.362 describes the interface `ara::rest::Uuid::GetTimeHighAndVersion.`] (*RS\_CM\_00300*)

### 8.34.11 GetClockSeq

<b>Service name:</b>	ara::rest::Uuid::GetClockSeq
<b>Type:</b>	Member function
<b>Syntax:</b>	std::uint16_t ara::rest::Uuid::GetClockSeq() const
<b>Function param:</b>	None
<b>Return value:</b>	a numeric value
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/uuid.h
<b>Class:</b>	ara::rest::Uuid
<b>Description:</b>	Returns clock sequence count.

**Table 8.363: ara::rest::Uuid::GetClockSeq**

[SWS\_REST\_02338] ara::rest::Uuid::GetClockSeq [Table 8.363 describes the interface [ara::rest::Uuid::GetClockSeq.](#)] ([RS\\_CM\\_00300](#))

### 8.34.12 GetNode

<b>Service name:</b>	ara::rest::Uuid::GetNode
<b>Type:</b>	Member function
<b>Syntax:</b>	std::uint64_t ara::rest::Uuid::GetNode() const
<b>Function param:</b>	None
<b>Return value:</b>	return type
<b>Exceptions:</b>	noexcept
<b>Header file:</b>	ara/rest/uuid.h
<b>Class:</b>	ara::rest::Uuid
<b>Description:</b>	Returns node value.

**Table 8.364: ara::rest::Uuid::GetNode**

[SWS\_REST\_02339] ara::rest::Uuid::GetNode [Table 8.364 describes the interface [ara::rest::Uuid::GetNode.](#)] ([RS\\_CM\\_00300](#))

### 8.34.13 operator==

<b>Service name:</b>	ara::rest::Uuid::operator==	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	friend bool operator==(const Uuid &a, const Uuid &b)	
<b>Function param:</b>	a	a uuid
<b>Function param:</b>	b	a uuid
<b>Return value:</b>	true if equal	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/uuid.h	
<b>Namespace:</b>	ara::rest::Uuid	
<b>Description:</b>	Compares UUIDs.	

**Table 8.365: ara::rest::Uuid::operator==**

[SWS\_REST\_02340] `ara::rest::Uuid::operator==` [Table 8.365 describes the interface `ara::rest::Uuid::operator==.`] (*RS\_CM\_00300*)

### 8.34.14 operator!=

<b>Service name:</b>	<code>ara::rest::Uuid::operator!=</code>	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<code>friend bool operator!=(const Uuid &amp;a, const Uuid &amp;b)</code>	
<b>Function param:</b>	<code>a</code>	<code>a uuid</code>
<b>Function param:</b>	<code>b</code>	<code>a uuid</code>
<b>Return value:</b>	true if unequal	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	<code>ara/rest/uuid.h</code>	
<b>Namespace:</b>	<code>ara::rest::Uuid</code>	
<b>Description:</b>	Compares UUIDs.	

**Table 8.366: `ara::rest::Uuid::operator!=`**

[SWS\_REST\_02341] `ara::rest::Uuid::operator!=` [Table 8.366 describes the interface `ara::rest::Uuid::operator!=.`] (*RS\_CM\_00300*)

### 8.34.15 operator<

<b>Service name:</b>	<code>ara::rest::Uuid::operator&lt;</code>	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<code>friend bool operator&lt;(const Uuid &amp;a, const Uuid &amp;b)</code>	
<b>Function param:</b>	<code>a</code>	<code>a uuid</code>
<b>Function param:</b>	<code>b</code>	<code>a uuid</code>
<b>Return value:</b>	true if uuid compares less than lexicographically less by component	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	<code>ara/rest/uuid.h</code>	
<b>Namespace:</b>	<code>ara::rest::Uuid</code>	
<b>Description:</b>	Compares UUIDs.	

**Table 8.367: `ara::rest::Uuid::operator<`**

[SWS\_REST\_02342] `ara::rest::Uuid::operator<` [Table 8.367 describes the interface `ara::rest::Uuid::operator<.`] (*RS\_CM\_00300*)

## 8.35 `ara::rest::ogm`

### 8.35.1 Copy

<b>Service name:</b>	Copy
<b>Type:</b>	Non-member function

<b>Syntax:</b>	<pre>template &lt;typename T &gt; Pointer&lt;T&gt; ara::rest::ogm::Copy(const T &amp;g, Allocator *alloc=GetDefaultAllocator())</pre>	
<b>Function param:</b>	g	OGM graph to copy
<b>Function param:</b>	alloc	allocator to use for the copy
<b>Return value:</b>	a copy	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/copy.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	Copies an object graph. Performs a deep copy of the argument	

**Table 8.368: ara::rest::ogm::Copy**

[SWS\_REST\_02343] **Copy** [Table 8.368 describes the interface [Copy](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.35.2 Copy

<b>Service name:</b>	Copy	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<pre>template &lt;typename T &gt; Pointer&lt;T&gt; ara::rest::ogm::Copy(const Pointer&lt; T &gt; &amp;g, Allocator *alloc=GetDefaultAllocator())</pre>	
<b>Function param:</b>	g	OGM graph to copy
<b>Function param:</b>	alloc	allocator to use for the copy
<b>Return value:</b>	a copy	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/copy.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	Copies an object graph. Performs a deep copy of the argument	

**Table 8.369: ara::rest::ogm::Copy**

[SWS\_REST\_02344] **Copy** [Table 8.369 describes the interface [Copy](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.35.3 Visit

<b>Service name:</b>	Visit	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<pre>template &lt;typename NodeT , typename... Visitors&gt; void ara::rest::ogm::Visit(const NodeT &amp;u, Visitors &amp;&amp;...vis)</pre>	
<b>Function param:</b>	u	OGM node to resolve
<b>Function param:</b>	vis	a set of functors
<b>Return value:</b>	None	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/ogm/visit.h	



<b>Namespace:</b>	ara::rest::ogm
<b>Description:</b>	Resolves the exact type of the OGM node passed to it. The function accepts a set of functors to call with the exact type of the graph node argument. Overload resolution shall apply here. If no visitor matches, the function silently returns without calling any visitor.

**Table 8.370: ara::rest::ogm::Visit**

[SWS\_REST\_02345] **Visit** [Table 8.370 describes the interface [Visit](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.35.4 Visit

<b>Service name:</b>	Visit	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	template <typename NodeT , typename... Visitors> void ara::rest::ogm::Visit(const Pointer< NodeT > &u, Visitors &&...vis)	
<b>Function param:</b>	u	OGM node to resolve
<b>Function param:</b>	vis	a set of functors
<b>Return value:</b>	None	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/ogm/visit.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	See documentation of void Visit(const Node&, Visitors&&);.	

**Table 8.371: ara::rest::ogm::Visit**

[SWS\_REST\_02346] **Visit** [Table 8.371 describes the interface [Visit](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.35.5 Visit

<b>Service name:</b>	Visit	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	template <typename NodeT , typename... Visitors> void ara::rest::ogm::Visit(NodeT &u, Visitors &&...vis)	
<b>Function param:</b>	u	OGM node to resolve
<b>Function param:</b>	vis	a set of functors
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/visit.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	See documentation of void Visit(const Node&, Visitors&&);.	

**Table 8.372: ara::rest::ogm::Visit**

[SWS\_REST\_02347] **Visit** [Table 8.372 describes the interface [Visit](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.35.6 Visit

<b>Service name:</b>	Visit	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	template <typename NodeT , typename... Visitors> void ara::rest::ogm::Visit(Pointer< NodeT > &u, Visitors &&...vis)	
<b>Function param:</b>	u	OGM node to resolve
<b>Function param:</b>	vis	a set of functors
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/visit.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	See documentation of void Visit(const Node&, Visitors&&);.	

**Table 8.373: ara::rest::ogm::Visit**

[SWS\_REST\_02348] **Visit** [Table 8.373 describes the interface [Visit](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.35.7 VisitAll

<b>Service name:</b>	VisitAll	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	template <typename NodeT , typename... Visitors> void ara::rest::ogm::VisitAll(const NodeT &u, Visitors &&...vis)	
<b>Function param:</b>	u	OGM node to resolve
<b>Function param:</b>	vis	a set of functors
<b>Return value:</b>	None	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/ogm/visit.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	Resolves the exact types of the OGM nodes in the graph and performs graph traversal. The function accepts a set of functors to call with the exact type of the graph node argument. Overload resolution shall apply here. If no visitor matches, the function silently returns without calling any visitor.	

**Table 8.374: ara::rest::ogm::VisitAll**

[SWS\_REST\_02411] **VisitAll** [Table 8.374 describes the interface [VisitAll](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.35.8 VisitAll

<b>Service name:</b>	VisitAll	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<pre>template &lt;typename NodeT , typename... Visitors&gt; void ara::rest::ogm::VisitAll(const Pointer&lt; NodeT &gt; &amp;u, Visitors &amp;&amp;...vis)</pre>	
<b>Function param:</b>	u	OGM node to resolve
<b>Function param:</b>	vis	a set of functors
<b>Return value:</b>	None	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/ogm/visit.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	See documentation of void VisitAll(const Node&, Visitors&&);.	

**Table 8.375: ara::rest::ogm::VisitAll**

[SWS\_REST\_02412] VisitAll [Table 8.375 describes the interface VisitAll.]  
 (RS\_CM\_00300, RS\_CM\_00305, RS\_CM\_00306, RS\_CM\_00307, RS\_CM\_00308)

### 8.35.9 VisitAll

<b>Service name:</b>	VisitAll	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<pre>template &lt;typename NodeT , typename... Visitors&gt; void ara::rest::ogm::VisitAll(NodeT &amp;u, Visitors &amp;&amp;...vis)</pre>	
<b>Function param:</b>	u	OGM node to resolve
<b>Function param:</b>	vis	a set of functors
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/visit.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	See documentation of void VisitAll(const Node&, Visitors&&);.	

**Table 8.376: ara::rest::ogm::VisitAll**

[SWS\_REST\_02413] VisitAll [Table 8.376 describes the interface VisitAll.]  
 (RS\_CM\_00300, RS\_CM\_00305, RS\_CM\_00306, RS\_CM\_00307, RS\_CM\_00308)

### 8.35.10 VisitAll

<b>Service name:</b>	VisitAll	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<pre>template &lt;typename NodeT , typename... Visitors&gt; void ara::rest::ogm::VisitAll(Pointer&lt; NodeT &gt; &amp;u, Visitors &amp;&amp;...vis)</pre>	
<b>Function param:</b>	u	OGM node to resolve

<b>Function param:</b>	vis	a set of functors
<b>Return value:</b>	None	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/visit.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	See documentation of void VisitAll(const Node&, Visitors&&);.	

**Table 8.377: ara::rest::ogm::VisitAll**

[SWS\_REST\_02414] **VisitAll** [Table 8.377 describes the interface [VisitAll.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.35.11 Get

<b>Service name:</b>	Get	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	template <typename ValueNodeT , typename NodeT> ValueNodeT& ara::rest::ogm::Get(const NodeT &u, const StringView &n)	
<b>Function param:</b>	u	Root of the search operation
<b>Function param:</b>	n	Field name to search
<b>Return value:</b>	value of the field	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/util.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	Returns a Value object with the given field name and type. Only ascends the graph from the given graph object. Found value is the first suitable value.	

**Table 8.378: ara::rest::ogm::Get**

[SWS\_REST\_02389] **Get** [Table 8.378 describes the interface [Get.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.35.12 Get

<b>Service name:</b>	Get	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	template <typename ValueNodeT , typename NodeT> ValueNodeT& ara::rest::ogm::Get(const Pointer< NodeT > &u, const StringView &n)	
<b>Function param:</b>	u	Root of the search operation
<b>Function param:</b>	n	Field name to search
<b>Return value:</b>	value of the field	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/util.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	See documentation of void Get(const NodeT&, const StringView&);.	

**Table 8.379: ara::rest::ogm::Get**

[SWS\_REST\_02390] **Get** [Table 8.379 describes the interface [Get.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.35.13 GetValue

<b>Service name:</b>	GetValue	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	template <typename ValueNodeT , typename NodeT> typename ValueNodeT::ValueType ara::rest::ogm::GetValue(const NodeT &u, const StringView &n)	
<b>Function param:</b>	u	Root of the search operation
<b>Function param:</b>	n	Field name to search
<b>Return value:</b>	value of the field	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/util.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	Returns a Value object's primitive value with the given field name and type. Only ascends the graph from the given graph object. Found value is the first suitable value.	

**Table 8.380: ara::rest::ogm::GetValue**

[SWS\_REST\_02391] **GetValue** [Table 8.380 describes the interface [Get.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.35.14 GetValue

<b>Service name:</b>	GetValue	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	template <typename ValueNodeT , typename NodeT> typename ValueNodeT::ValueType ara::rest::ogm::GetValue(const Pointer< NodeT > &u, const StringView &n)	
<b>Function param:</b>	u	Root of the search operation
<b>Function param:</b>	n	Field name to search
<b>Return value:</b>	value of the field	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/util.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	See documentation of void GetValue(const NodeT&, const StringView&);.	

**Table 8.381: ara::rest::ogm::GetValue**

[SWS\_REST\_02392] **GetValue** [Table 8.381 describes the interface [Get.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.35.15 Set

<b>Service name:</b>	Set	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<pre>template &lt;typename ValueNodeT , typename ValueT&gt; bool ara::rest::ogm::Set(ValueNodeT &amp;u, ValueT &amp;v)</pre>	
<b>Function param:</b>	u	Node that will be updated
<b>Function param:</b>	v	Value that will be set for the node
<b>Return value:</b>	returns true if value was set, otherwise false	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/util.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	Sets a value to a node if the value differs from the already set value.	

**Table 8.382: ara::rest::ogm::Set**

[SWS\_REST\_02393] **Set** [Table 8.382 describes the interface [Set.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.35.16 Set

<b>Service name:</b>	Set	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<pre>bool ara::rest::ogm::Set(Array &amp;u, Array::MoveRange &amp;v)</pre>	
<b>Function param:</b>	u	Array that will be updated
<b>Function param:</b>	v	Values that will be set for the array
<b>Return value:</b>	returns true if value was set, otherwise false	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/util.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	Sets values to an array if the given values differ from the already set value.	

**Table 8.383: ara::rest::ogm::Set**

[SWS\_REST\_02405] **Set** [Table 8.383 describes the interface [Set.](#)] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

### 8.35.17 Set

<b>Service name:</b>	Set	
<b>Type:</b>	Non-member function	

<b>Syntax:</b>	bool ara::rest::ogm::Set(Object &u, Object::MoveFieldRange &v)	
<b>Function param:</b>	u	Object that will be updated
<b>Function param:</b>	v	Fields that will be set for the object
<b>Return value:</b>	returns true if fields were set, otherwise false	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/util.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	Sets fields to a object if the field values differ from the already set fields.	

**Table 8.384: ara::rest::ogm::Set**

[SWS\_REST\_02406] Set [Table 8.384 describes the interface Set.] (RS\_CM\_00300, RS\_CM\_00305, RS\_CM\_00306, RS\_CM\_00307, RS\_CM\_00308)

### 8.35.18 SetValue

<b>Service name:</b>	SetValue	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	template <typename ValueNodeT , typename NodeT , typename ValueT> bool ara::rest::ogm::SetValue(const ValueNodeT &u, const StringView &n, ValueT &v)	
<b>Function param:</b>	u	Node that will be updated
<b>Function param:</b>	n	Field name to set value to
<b>Function param:</b>	v	Value that will be set for the field
<b>Return value:</b>	returns true if value was set, otherwise false	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/util.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	Sets a primitive value to a node field if the value differs from the already set value.	

**Table 8.385: ara::rest::ogm::SetValue**

[SWS\_REST\_02407] SetValue [Table 8.385 describes the interface SetValue.] (RS\_CM\_00300, RS\_CM\_00305, RS\_CM\_00306, RS\_CM\_00307, RS\_CM\_00308)

### 8.35.19 SetValue

<b>Service name:</b>	SetValue	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	template <typename ValueNodeT , typename NodeT , typename ValueT> bool ara::rest::ogm::SetValue(const Pointer< ValueNodeT > &u, const StringView &n, ValueT &v)	
<b>Function param:</b>	u	Node that will be updated
<b>Function param:</b>	n	Field name to set value to

<b>Function param:</b>	v	Value that will be set for the field
<b>Return value:</b>	returns true if value was set, otherwise false	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/util.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	See documentation of void SetValue(const ValueNodeT&, const StringView&, ValueT);.	

**Table 8.386: ara::rest::ogm::SetValue**

[SWS\_REST\_02409] **SetValue** [Table 8.386 describes the interface [SetValue](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

## 8.35.20 Cast

<b>Service name:</b>	Cast	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	template <typename NodeT> Pointer< NodeT > ara::rest::ogm::Cast (Pointer<Node> n)	
<b>Function param:</b>	n	Node to cast
<b>Return value:</b>	returns pointer to casted node	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/ogm/util.h	
<b>Namespace:</b>	ara::rest::ogm	
<b>Description:</b>	Casts a node to a concrete node type.	

**Table 8.387: ara::rest::ogm::Cast**

[SWS\_REST\_02410] **Cast** [Table 8.387 describes the interface [Cast](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00305](#), [RS\\_CM\\_00306](#), [RS\\_CM\\_00307](#), [RS\\_CM\\_00308](#))

## 8.36 ara::rest

### 8.36.1 RequestMethod

<b>Name:</b>	RequestMethod	
<b>Type:</b>	Non-member enumeration	
<b>Range:</b>	kGet	= 1 « 0
	kPost	= 1 « 1
	kPut	= 1 « 2
	kDelete	= 1 « 3
	kOptions	= 1 « 4
	kHead	= 1 « 5



<b>Syntax:</b>	<pre>enum class RequestMethod : std::uint32_t {     kGet = 1 &lt;&lt; 0,     kPost = 1 &lt;&lt; 1,     kPut = 1 &lt;&lt; 2,     kDelete = 1 &lt;&lt; 3,     kOptions = 1 &lt;&lt; 4,     kHead = 1 &lt;&lt; 5 };</pre>
<b>Header file:</b>	ara/rest/endpoint.h
<b>Namespace:</b>	ara::rest
<b>Description:</b>	Specifies a set possible API access methods. RequestMethod largely corresponds to typical RESTful API access methods.

**Table 8.388: ara::rest::RequestMethod**

[SWS\_REST\_02349] **RequestMethod** [Table 8.388 describes the enumeration datatype `ara::rest::RequestMethod`.] ([RS\\_CM\\_00300](#))

### 8.36.2 SubscriptionState

<b>Name:</b>	SubscriptionState								
<b>Type:</b>	Non-member enumeration								
<b>Range:</b>	<table border="1"> <tr> <td>kSubscribed</td> <td></td> </tr> <tr> <td>kCanceled</td> <td></td> </tr> <tr> <td>kResubscribe</td> <td></td> </tr> <tr> <td>kInvalid</td> <td></td> </tr> </table>	kSubscribed		kCanceled		kResubscribe		kInvalid	
kSubscribed									
kCanceled									
kResubscribe									
kInvalid									
<b>Syntax:</b>	<pre>enum class SubscriptionState {     kSubscribed,     kCanceled,     kResubscribe,     kInvalid };</pre>								
<b>Header file:</b>	ara/rest/endpoint.h								
<b>Namespace:</b>	ara::rest								
<b>Description:</b>	Denotes the state of the subscription relation represented by an Event. The enumerators have the following meaning:								

**Table 8.389: ara::rest::SubscriptionState**

[SWS\_REST\_02350] **SubscriptionState** [Table 8.389 describes the enumeration datatype `ara::rest::SubscriptionState`.] ([RS\\_CM\\_00300](#))

### 8.36.3 EventPolicy

<b>Name:</b>	EventPolicy				
<b>Type:</b>	Non-member enumeration				
<b>Range:</b>	<table border="1"> <tr> <td>kTriggered</td> <td>= 1u &lt;&lt; 0</td> </tr> <tr> <td>kPeriodic</td> <td>= 1u &lt;&lt; 1</td> </tr> </table>	kTriggered	= 1u << 0	kPeriodic	= 1u << 1
kTriggered	= 1u << 0				
kPeriodic	= 1u << 1				

<b>Syntax:</b>	<pre>enum class EventPolicy : std::uint32_t {     kTriggered = 1u &lt;&lt; 0,     kPeriodic = 1u &lt;&lt; 1 };</pre>
<b>Header file:</b>	ara/rest/endpoint.h
<b>Namespace:</b>	ara::rest
<b>Description:</b>	Mode of operation for event subscriptions. Defines the mode of operation for event subscriptions. The modes have the following semantics:

**Table 8.390: ara::rest::EventPolicy**

[SWS\_REST\_02351] **EventPolicy** [Table 8.390 describes the enumeration datatype `ara::rest::EventPolicy`.] ([RS\\_CM\\_00300](#))

### 8.36.4 ShutdownPolicy

<b>Name:</b>	ShutdownPolicy
<b>Type:</b>	Non-member enumeration
<b>Range:</b>	kForced kGraceful
<b>Syntax:</b>	<pre>enum class ShutdownPolicy : std::uint32_t {     kForced,     kGraceful };</pre>
<b>Header file:</b>	ara/rest/endpoint.h
<b>Namespace:</b>	ara::rest
<b>Description:</b>	Specifies shutdown behavior of endpoints. Endpoints can shut down "gracefully", which allows all ongoing transactions to finish while blocking the caller. A forced shutdown must cancel or terminate all transactions as fast as possible does not block the caller for "unreasonably" long period of time. During a forced shutdown I/O is not allowed. Precise semantics of these policies are implementation defined.

**Table 8.391: ara::rest::ShutdownPolicy**

[SWS\_REST\_02352] **ShutdownPolicy** [Table 8.391 describes the enumeration datatype `ara::rest::ShutdownPolicy`.] ([RS\\_CM\\_00300](#))

### 8.36.5 StartupPolicy

<b>Name:</b>	StartupPolicy
<b>Type:</b>	Non-member enumeration
<b>Range:</b>	kDetached kAttached
<b>Syntax:</b>	<pre>enum class StartupPolicy : std::uint32_t {     kDetached,     kAttached };</pre>

<b>Header file:</b>	ara/rest/endpoint.h
<b>Namespace:</b>	ara::rest
<b>Description:</b>	Specifies whether a server will detach itself from its owning context. If a server is started "detached" then ara::rest::Server::Start() does not block. Effectively it will request a separate execution context (such as a thread) from

**Table 8.392: ara::rest::StartupPolicy**

[SWS\_REST\_02353] **StartupPolicy** [Table 8.392 describes the enumeration datatype `ara::rest::StartupPolicy`.] ([RS\\_CM\\_00300](#))

### 8.36.6 Function

<b>Name:</b>	Function
<b>Type:</b>	Non-member type alias
<b>Syntax:</b>	template <typename T> using ara::rest::Function = std::function<T>
<b>Header file:</b>	ara/rest/function.h
<b>Namespace:</b>	ara::rest
<b>Description:</b>	A generalized function pointer.

**Table 8.393: ara::rest::Function**

[SWS\_REST\_02354] **Function** [Table 8.393 describes the type alias `ara::rest::Function`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00311](#))

### 8.36.7 Pointer

<b>Name:</b>	Pointer
<b>Type:</b>	Non-member type alias
<b>Syntax:</b>	template<typename T> using ara::rest::Pointer = std::unique_ptr<T>
<b>Header file:</b>	ara/rest/pointer.h
<b>Namespace:</b>	ara::rest
<b>Description:</b>	The equivalent if <code>std::unique_ptr</code> for ara::rest internal uses.

**Table 8.394: ara::rest::Pointer**

[SWS\_REST\_02355] **Pointer** [Table 8.394 describes the type alias `ara::rest::Pointer`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00311](#))

### 8.36.8 Task

<b>Name:</b>	Task
<b>Type:</b>	Non-member type alias

<b>Syntax:</b>	template<typename T> using ara::rest::Task = ara::core::Future<T>
<b>Header file:</b>	ara/rest/task.h
<b>Namespace:</b>	ara::rest
<b>Description:</b>	Represents an asynchronous task for which a user might want to wait for.

**Table 8.395: ara::rest::Task**

[SWS\_REST\_02360] **Task** [Table 8.395 describes the type alias `ara::rest::Task`.] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00311](#))

### 8.36.9 duration\_t

<b>Name:</b>	duration_t
<b>Type:</b>	Non-member type alias
<b>Syntax:</b>	using ara::rest::duration_t = std::chrono::microseconds
<b>Header file:</b>	ara/rest/types.h
<b>Namespace:</b>	ara::rest
<b>Description:</b>	Specifies an amount of time of granularity of at least microseconds.

**Table 8.396: ara::rest::duration\_t**

[SWS\_REST\_02361] **duration\_t** [Table 8.396 describes the type alias `ara::rest::duration_t`.] ([RS\\_CM\\_00300](#))

### 8.36.10 operator==

<b>Service name:</b>	operator==	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	bool ara::rest::operator==(const Allocator &a, const Allocator &b)	
<b>Function param:</b>	a	an allocator
<b>Function param:</b>	b	an allocator
<b>Return value:</b>	true allocators compare equal	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/allocator.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Tests two allocators for equality.	

**Table 8.397: ara::rest::operator==**

[SWS\_REST\_02362] **operator==** [Table 8.397 describes the interface `operator==`.] ([RS\\_CM\\_00300](#))

### 8.36.11 operator!=

<b>Service name:</b>	operator!=	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	bool ara::rest::operator!=(const Allocator &a, const Allocator &b)	
<b>Function param:</b>	a	an allocator
<b>Function param:</b>	b	an allocator
<b>Return value:</b>	true allocators compare unequal	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/allocator.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Tests two allocators for inequality.	

**Table 8.398: ara::rest::operator!=**

[SWS\_REST\_02363] operator!= [Table 8.398 describes the interface [operator!=.](#)] ([RS\\_CM\\_00300](#))

### 8.36.12 NewDeleteAllocator

<b>Service name:</b>	NewDeleteAllocator	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	Allocator* ara::rest::NewDeleteAllocator()	
<b>Function param:</b>	None	
<b>Return value:</b>	a pointer to a NewDeleteAllocator	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/allocator.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Identical to std::pmr::new_delete_resource.	

**Table 8.399: ara::rest::NewDeleteAllocator**

[SWS\_REST\_02364] NewDeleteAllocator [Table 8.399 describes the interface [NewDeleteAllocator.](#)] ([RS\\_CM\\_00300](#))

### 8.36.13 GetDefaultAllocator

<b>Service name:</b>	GetDefaultAllocator	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	Allocator* ara::rest::GetDefaultAllocator()	
<b>Function param:</b>	None	
<b>Return value:</b>	a pointer to the default allocator	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/allocator.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	See std::pmr::get_default_allocator for details.	

**Table 8.400: ara::rest::GetDefaultAllocator**

[SWS\_REST\_02365] **GetDefaultAllocator** [Table 8.400 describes the interface `GetDefaultAllocator`.] ([RS\\_CM\\_00300](#))

### 8.36.14 SetDefaultAllocator

<b>Service name:</b>	SetDefaultAllocator	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	Allocator* ara::rest::SetDefaultAllocator(Allocator *a)	
<b>Function param:</b>	a	an allocator
<b>Return value:</b>	a pointer to the allocator just set	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/allocator.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	See std::pmr::set_default_allocator for details.	

**Table 8.401: ara::rest::SetDefaultAllocator**

[SWS\_REST\_02366] **SetDefaultAllocator** [Table 8.401 describes the interface `SetDefaultAllocator`.] ([RS\\_CM\\_00300](#))

### 8.36.15 operator==

<b>Service name:</b>	operator==	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	template <typename T , typename U > bool ara::rest::operator==(const StdAllocator< T > &a, const StdAllocator< U > &b)	
<b>Function param:</b>	a	an allocator
<b>Function param:</b>	b	an allocator
<b>Return value:</b>	true if memory allocated in one can be freed via other	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/allocator.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Tests allocators for equality.	

**Table 8.402: ara::rest::operator==**

[SWS\_REST\_02367] **operator==** [Table 8.402 describes the interface `operator==`.] ([RS\\_CM\\_00300](#))

### 8.36.16 operator!=

<b>Service name:</b>	operator!=	
<b>Type:</b>	Non-member function	

<b>Syntax:</b>	<pre>template &lt;typename T , typename U &gt; bool ara::rest::operator!=(StdAllocator&lt; T &gt; const &amp;x, StdAllocator&lt; U &gt; const &amp;y)</pre>	
<b>Function param:</b>	x	an allocator
<b>Function param:</b>	y	an allocator
<b>Return value:</b>	true if memory allocated in x cannot be freed via y	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/allocator.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Tests allocators for inequality.	

**Table 8.403: ara::rest::operator!=**

[SWS\_REST\_02368] operator!= [Table 8.403 describes the interface operator!=.]  
 (RS\_CM\_00300)

### 8.36.17 operator|

<b>Service name:</b>	operator	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<pre>constexpr RequestMethod ara::rest::operator (RequestMethod a, RequestMethod b)</pre>	
<b>Function param:</b>	a	a (set of) request method enumerator(s)
<b>Function param:</b>	b	a (set of) request method enumerator(s)
<b>Return value:</b>	a set of request method enumerator(s)	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/endpoint.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Computes a set of RequestMethod enumerators.	

**Table 8.404: ara::rest::operator|**

[SWS\_REST\_02369] operator| [Table 8.404 describes the interface operator|.]  
 (RS\_CM\_00300)

### 8.36.18 operator|

<b>Service name:</b>	operator	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<pre>constexpr EventPolicy ara::rest::operator (EventPolicy a, EventPolicy b)</pre>	
<b>Function param:</b>	a	a (set of) request event policy enumerator(s)
<b>Function param:</b>	b	a (set of) request event policy enumerator(s)
<b>Return value:</b>	a set of request event policy enumerator(s)	
<b>Exceptions:</b>	noexcept	
<b>Header file:</b>	ara/rest/endpoint.h	
<b>Namespace:</b>	ara::rest	

<b>Description:</b>	Computes a set of EventPolicy enumerators.
---------------------	--

**Table 8.405: ara::rest::operator|**

[SWS\_REST\_02370] **operator|** [Table 8.405 describes the interface [operator|.](#)] ([RS\\_CM\\_00300](#))

### 8.36.19 MakeliteratorRange

<b>Service name:</b>	MakeliteratorRange	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<pre>template &lt;typename IterT &gt; IteratorRange&lt;IterT&gt; ara::rest::MakeIteratorRange(IterT a, IterT b)</pre>	
<b>Function param:</b>	a	iterator that denotes the start of the sequence
<b>Function param:</b>	b	iterator that denotes the end of the sequence
<b>Return value:</b>	an IteratorRange	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/iterator.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Helper for type deduction to construct an IteratorRange.	

**Table 8.406: ara::rest::MakeliteratorRange**

[SWS\_REST\_02371] **MakeliteratorRange** [Table 8.406 describes the interface [MakeIteratorRange.](#)] ([RS\\_CM\\_00300](#))

### 8.36.20 MakeMoveIteratorRange

<b>Service name:</b>	MakeMoveIteratorRange	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	<pre>template &lt;typename IterT &gt; MoveIteratorRange&lt;IterT&gt; ara::rest::MakeMoveIteratorRange(IterT a, IterT b)</pre>	
<b>Function param:</b>	a	iterator that denotes the start of the sequence
<b>Function param:</b>	b	iterator that denotes the end of the sequence
<b>Return value:</b>	an MakeMoveIteratorRange	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/iterator.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Helper for type deduction to construct an MoveIteratorRange.	

**Table 8.407: ara::rest::MakeMoveIteratorRange**

[SWS\_REST\_02396] **MakeMoveIteratorRange** [Table 8.407 describes the interface [MakeMoveIteratorRange.](#)] ([RS\\_CM\\_00300](#))



### 8.36.21 Resolve

<b>Service name:</b>	Resolve	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	Uri ara::rest::Resolve(const Uri &base, const Uri &rel, Allocator *alloc=GetDefaultAllocator())	
<b>Function param:</b>	base	the URI to resolve against
<b>Function param:</b>	rel	a relative URI
<b>Function param:</b>	alloc	an allocator
<b>Return value:</b>	a resolved URI	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Resolves a relative URI against a base URI. See section 5.2 of RFC 3986 for the algorithm used.	

**Table 8.408: ara::rest::Resolve**

[SWS\_REST\_02372] **Resolve** [Table 8.408 describes the interface [Resolve](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.36.22 Normalize

<b>Service name:</b>	Normalize	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	Uri ara::rest::Normalize(const Uri &uri, Allocator *alloc=GetDefaultAllocator())	
<b>Function param:</b>	uri	URI to normalize
<b>Function param:</b>	alloc	an allocator
<b>Return value:</b>	a normalized URI	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Normalizes a given URI.	

**Table 8.409: ara::rest::Normalize**

[SWS\_REST\_02373] **Normalize** [Table 8.409 describes the interface [Normalize](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.36.23 Relativize

<b>Service name:</b>	Relativize	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	Uri ara::rest::Relativize(const Uri &base, const Uri &uri, Allocator *alloc=GetDefaultAllocator())	
<b>Function param:</b>	base	a base URI as reference
<b>Function param:</b>	uri	a URI to relativize

<b>Function param:</b>	alloc	an allocator
<b>Return value:</b>	a relative URI	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Relativizes a URI with respect to a given base URI. The relativization of the given URI against this URI is computed as follows:	

**Table 8.410: ara::rest::Relativize**

[SWS\_REST\_02374] **Relativize** [Table 8.410 describes the interface [Relativize](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.36.24 ToString

<b>Service name:</b>	ToString	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	String ara::rest::ToString(const Uri &uri, Uri::Part part, bool encode, Allocator *alloc=GetDefaultAllocator())	
<b>Function param:</b>	uri	URI to encode
<b>Function param:</b>	part	denotes which components of a URI should be encoded
<b>Function param:</b>	encode	if true, then the string will be percent-encoded. If false, the string must not be string encoded.
<b>Function param:</b>	alloc	a user-defined allocator passed to the string object being returned
<b>Return value:</b>	the encoded URI	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Returns a string representation of a Uri.	

**Table 8.411: ara::rest::ToString**

[SWS\_REST\_02375] **ToString** [Table 8.411 describes the interface [ToString](#).] ([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.36.25 ToString

<b>Service name:</b>	ToString	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	String ara::rest::ToString(const Uri &uri, Uri::Part part, Allocator *alloc=GetDefaultAllocator())	
<b>Function param:</b>	uri	URI to encode
<b>Function param:</b>	part	denotes which components of a URI should be encoded

<b>Function param:</b>	alloc	a user-defined allocator passed to the string object being returned
<b>Return value:</b>	the encoded URI	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Returns a string representation of a Uri.	

**Table 8.412: ara::rest::ToString**

[SWS\_REST\_02376] **ToString** [Table 8.412 describes the interface `ToString`.]  
 (RS\_CM\_00300, RS\_CM\_00304)

### 8.36.26 ToString

<b>Service name:</b>	ToString	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	String ara::rest::ToString(const Uri &uri, Allocator *alloc=GetDefaultAllocator())	
<b>Function param:</b>	uri	URI to encode
<b>Function param:</b>	alloc	a user-defined allocator passed to the string object being returned
<b>Return value:</b>	the encoded URI	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Returns a string representation of a Uri.	

**Table 8.413: ara::rest::ToString**

[SWS\_REST\_02377] **ToString** [Table 8.413 describes the interface `ToString`.]  
 (RS\_CM\_00300, RS\_CM\_00304)

### 8.36.27 ToString

<b>Service name:</b>	ToString	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	String ara::rest::ToString(Uri &&uri, Uri::Part part, bool encode, Allocator *alloc=GetDefaultAllocator())	
<b>Function param:</b>	uri	URI to encode
<b>Function param:</b>	part	denotes which components of a URI should be encoded
<b>Function param:</b>	encode	if true, then the string will be percent-encoded. If false, the string must not be string encoded.
<b>Function param:</b>	alloc	a user-defined allocator passed to the string object being returned
<b>Return value:</b>	the encoded URI	
<b>Exceptions:</b>	Implementation-defined	

<b>Header file:</b>	ara/rest/uri.h
<b>Namespace:</b>	ara::rest
<b>Description:</b>	Returns a string representation of a Uri.

**Table 8.414: ara::rest::ToString**

[SWS\_REST\_02378] **ToString** [Table 8.414 describes the interface [ToString](#).]  
 (RS\_CM\_00300, RS\_CM\_00304)

### 8.36.28 ToString

<b>Service name:</b>	ToString	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	String ara::rest::ToString(Uri &uri, Uri::Part part, Allocator *alloc=GetDefaultAllocator())	
<b>Function param:</b>	uri	URI to encode
<b>Function param:</b>	part	denotes which components of a URI should be encoded
<b>Function param:</b>	alloc	a user-defined allocator passed to the string object being returned
<b>Return value:</b>	the encoded URI	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Returns a string representation of a Uri.	

**Table 8.415: ara::rest::ToString**

[SWS\_REST\_02379] **ToString** [Table 8.415 describes the interface [ToString](#).]  
 (RS\_CM\_00300, RS\_CM\_00304)

### 8.36.29 ToString

<b>Service name:</b>	ToString	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	String ara::rest::ToString(Uri &uri, Allocator *alloc=GetDefaultAllocator())	
<b>Function param:</b>	uri	URI to encode
<b>Function param:</b>	alloc	a user-defined allocator passed to the string object being returned
<b>Return value:</b>	the encoded URI	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uri.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Returns a string representation of a Uri.	

**Table 8.416: ara::rest::ToString**

[SWS\_REST\_02380] **ToString** [Table 8.416 describes the interface `ToString`.]  
([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.36.30 ToString

<b>Service name:</b>	ToString	
<b>Type:</b>	Non-member function	
<b>Syntax:</b>	String ara::rest::ToString(const Uuid &uuid, Allocator *alloc=GetDefaultAllocator())	
<b>Function param:</b>	uuid	a UUID
<b>Function param:</b>	alloc	an allocator
<b>Return value:</b>	its canonic textual representation	
<b>Exceptions:</b>	Implementation-defined	
<b>Header file:</b>	ara/rest/uuid.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Converts Uuid into its canonical textual representation.	

**Table 8.417: ara::rest::ToString**

[SWS\_REST\_02381] **ToString** [Table 8.417 describes the interface `ToString`.]  
([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

### 8.36.31 InstanceIdentifier

<b>Service name:</b>	InstanceIdentifier	
<b>Type:</b>	Non-member variable	
<b>Syntax:</b>	using ara::rest::InstanceIdentifier = ara::core::StringView;	
<b>Header file:</b>	ara/rest/endpoint.h	
<b>Namespace:</b>	ara::rest	
<b>Description:</b>	Identifies the concrete <code>ara::rest::Client</code> and <code>ara::rest::Server</code> instance.	

**Table 8.418: ara::rest::InstanceIdentifier**

[SWS\_REST\_10902] **InstanceIdentifier** [Table 8.418 describes the interface `InstanceIdentifier`.]  
([RS\\_CM\\_00300](#), [RS\\_CM\\_00304](#))

## A Mentioned Class Tables

For the sake of completeness, this chapter contains a set of class tables representing meta-classes mentioned in the context of this document but which are not contained directly in the scope of describing specific meta-model semantics.

<b>Enumeration</b>	<b>HttpAcceptEncodingEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::REST::RESTDeployment
<b>Note</b>	This enumeration defines the value for the accept-encoding field of the HTTP header. <b>Tags:</b> atp.Status=draft
<b>Literal</b>	<b>Description</b>
deflate	Use deflate compression. <b>Tags:</b> atp.EnumerationValue=1
gzip	Use gzip pcompression. <b>Tags:</b> atp.EnumerationValue=0

**Table A.1: HttpAcceptEncodingEnum**

<b>Class</b>	<b>Ipv4Configuration</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
<b>Note</b>	Internet Protocol version 4 (IPv4) configuration.			
<b>Base</b>	<i>ARObject, NetworkEndpointAddress</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
assignment Priority	PositiveInteger	0..1	attr	Priority of assignment (1 is highest). If a new address from an assignment method with a higher priority is available, it overwrites the IP address previously assigned by an assignment method with a lower priority.
defaultGateway	Ip4AddressString	0..1	attr	IP address of the default gateway.
dnsServer Address	Ip4AddressString	*	attr	IP addresses of preconfigured DNS servers. <b>Tags:</b> xml.namePlural=DNS-SERVER-ADDRESSES
ipAddressKeep Behavior	IpAddressKeepEnum	0..1	attr	Defines the lifetime of a dynamically fetched IP address.
ipv4Address	Ip4AddressString	0..1	attr	IPv4 Address. Notation: 255.255.255.255. The IP Address shall be declared in case the ipv4Address Source is FIXED and thus no auto-configuration mechanism is used.
ipv4Address Source	Ipv4AddressSource Enum	0..1	attr	Defines how the node obtains its IP address.
networkMask	Ip4AddressString	0..1	attr	Network mask. Notation 255.255.255.255
ttl	PositiveInteger	0..1	attr	Lifespan of data (0..255). The purpose of the TimeToLive field is to avoid a situation in which an undeliverable datagram keeps circulating on a system.

**Table A.2: Ipv4Configuration**

<b>Class</b>	<b>Ipv6Configuration</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
<b>Note</b>	Internet Protocol version 6 (IPv6) configuration.			
<b>Base</b>	ARObject, NetworkEndpointAddress			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
assignment Priority	PositiveInteger	0..1	attr	Priority of assignment (1 is highest). If a new address from an assignment method with a higher priority is available, it overwrites the IP address previously assigned by an assignment method with a lower priority.
defaultRouter	Ip6AddressString	0..1	attr	IP address of the default router.
dnsServer Address	Ip6AddressString	*	attr	IP addresses of pre configured DNS servers. <b>Tags:</b> xml.namePlural=DNS-SERVER-ADDRESSES
enableAnycast	Boolean	0..1	attr	This attribute is used to enable anycast addressing (i.e. to one of multiple receivers).
hopCount	PositiveInteger	0..1	attr	The distance between two hosts. The hop count n means that n gateways separate the source host from the destination host (Range 0..255)
ipAddressKeep Behavior	IpAddressKeepEnum	0..1	attr	Defines the lifetime of a dynamically fetched IP address.
ipAddressPrefix Length	PositiveInteger	0..1	attr	IPv6 prefix length defines the part of the IPv6 address that is the network prefix.
ipv6Address	Ip6AddressString	0..1	attr	IPv6 Address. Notation: FFFF::...:FFFF. The IP Address shall be declared in case the ipv6Address Source is FIXED and thus no auto-configuration mechanism is used.
ipv6Address Source	Ipv6AddressSource Enum	0..1	attr	Defines how the node obtains its IP address.

**Table A.3: Ipv6Configuration**

<b>Class</b>	<b>NetworkEndpoint</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
<b>Note</b>	The network endpoint defines the network addressing (e.g. IP-Address or MAC multicast address). <b>Tags:</b> atp.ManifestKind=MachineManifest			
<b>Base</b>	ARObject, Identifiable, MultilanguageReferrable, Referrable			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
fullyQualified DomainName	String	0..1	attr	Defines the fully qualified domain name (FQDN) e.g. some.example.host.
infrastructure Services	InfrastructureServices	0..1	aggr	Defines the network infrastructure services provided or consumed.
ipSecConfig	IPSecConfig	0..1	aggr	Optional IPSec configuration that provides security services for IP packets. <b>Tags:</b> atp.Status=draft
network Endpoint Address	NetworkEndpoint Address	1..*	aggr	Definition of a Network Address. <b>Tags:</b> xml.name Plural=NETWORK-ENDPOINT-ADDRESSES
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.

**Table A.4: NetworkEndpoint**

<b>Class</b>	<b>RestHttpPortPrototypeMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::REST::RESTDeployment			
<b>Note</b>	<p>This meta-class represents the ability to define pieces of a URI for the REST service that cannot be contributed from the design point of view.</p> <p><b>Tags:</b> atp.ManifestKind=ExecutionManifest atp.Status=draft atp.recommendedPackage=RestHttpPortPrototypeMappings</p>			
<b>Base</b>	<i>ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, UploadablePackageElement</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
acceptsEncoding	HttpAcceptEncoding	*	aggr	<p>This aggregation represents the collection of accepted encodings.</p> <p><b>Tags:</b> atp.Status=draft</p>
host	<a href="#">NetworkEndpoint</a>	0..1	ref	<p>This reference identifies the host configuration of the remote end.</p> <p><b>Tags:</b> atp.Status=draft</p>
portPrototype	PortPrototype	0..1	iref	<p>This reference identifies the instance of the PortPrototype to which the elements of the URI shall be defined.</p> <p><b>Tags:</b> atp.Status=draft</p>
portPrototypeSlugFragment	String	0..1	attr	<p>This attribute contributes a string value to be taken as the slug reference that represents the PortPrototype level of a REST service.</p> <p><b>Tags:</b> atp.Status=draft</p>
process	Process	0..1	ref	<p>This reference represents the process required for context of the mapping.</p> <p><b>Tags:</b> atp.Status=draft</p>
tcpPort	PositiveInteger	1	attr	<p>This attribute represents the value of the TCP port applicable for this mapping.</p> <p><b>Tags:</b> atp.Status=draft</p>
tlsSecureComProps	<a href="#">TlsSecureComProps</a>	0..1	ref	<p>This represents the configuration of TLS applicable for the mapping.</p> <p><b>Tags:</b> atp.Status=draft</p>

**Table A.5: RestHttpPortPrototypeMapping**

<b>Class</b>	<b>RestServiceInterface</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::REST::RESTDesign			
<b>Note</b>	<p>This meta-class represents a REST service.</p> <p><b>Tags:</b> atp.Status=draft atp.recommendedPackage=RestServiceInterfaces</p>			
<b>Base</b>	<i>ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, PortInterface, Referrable</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
resource	RestResourceDef	*	aggr	<p>This aggregation represents the collection of resources owned by the enclosing REST service.</p> <p><b>Tags:</b> atp.Status=draft</p>

**Table A.6: RestServiceInterface**



<b>Class</b>	<b>TlsSecureComProps</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::ServiceInstanceManifest::SecureCommunication			
<b>Note</b>	Configuration of the Transport Layer Security protocol (TLS). <b>Tags:</b> atp.ManifestKind=ServiceInstanceManifest atp.Status=draft			
<b>Base</b>	<i>ARObject, Identifiable, MultilanguageReferrable, Referrable, SecureComProps</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
keyExchange	CryptoServicePrimitive	*	ref	This reference identifies the shared (i.e. applicable for each of the aggregated cipher suites) crypto service primitive for the execution of key exchange during the handshake phase. <b>Tags:</b> atp.Status=draft
tlsCipherSuite	TlsCryptoCipherSuite	*	aggr	Collection of supported cipher suites that are used to negotiate the security settings for a network connection defined by the ServiceInstanceToMachineMapping. <b>Tags:</b> atp.Status=draft

**Table A.7: TlsSecureComProps**