

Package ‘ows4R’

January 27, 2022

Version 0.2-1

Date 2022-01-26

Title Interface to OGC Web-Services (OWS)

Maintainer Emmanuel Blondel <emmanuel.blondel1@gmail.com>

Depends R (>= 3.3.0), methods, geometa, keyring

Imports R6, httr, openssl, XML (>= 3.96-1.1), sf, rgdal, parallel

Suggests testthat

Description Provides an Interface to Web-Services defined as standards by the Open Geospatial Consortium (OGC), including Web Feature Service (WFS) for vector data, Catalogue Service (CSW) for ISO/OGC metadata, and associated standards such as the common web-service specification (OWS) and OGC Filter Encoding. Partial support is provided for the Web Map Service (WMS) and Web Processing Service (WPS). The purpose is to add support for additional OGC service standards such as Web Coverage Service (WCS), Web Coverage Processing Service (WCPS) or OGC API.

License MIT + file LICENSE

URL <https://github.com/eblondel/ows4R>, <https://www.ogc.org/standards>

BugReports <https://github.com/eblondel/ows4R/issues>

LazyLoad yes

RoxygenNote 7.1.0

NeedsCompilation no

Author Emmanuel Blondel [aut, cre] (<<https://orcid.org/0000-0002-5870-5762>>),
Alexandre Bennici [ctb] (<<https://orcid.org/0000-0003-2160-3487>>),
Norbert Billet [ctb]

Repository CRAN

Date/Publication 2022-01-26 23:00:02 UTC

R topics documented:

And	3
BBOX	4
BinaryComparisonOpType	5
BinaryLogicOpType	6
CSWCapabilities	6
CSWClient	7
CSWConstraint	9
CSWDescribeRecord	9
CSWGetRecordById	10
CSWGetRecords	11
CSWHarvest	11
CSWQuery	12
CSWRecordProperty	13
CSWTransaction	14
getOWSNamespace	15
getOWSNamespace	15
Not	16
OGCAbstractObject	16
OGCExpression	17
OGCFilter	18
Or	19
ows4R	19
OWSBoundingBox	20
OWSCapabilities	21
OWSClient	22
OWSCodeType	23
OWSGetCapabilities	23
OWSHttpRequest	24
OWSNamespace	25
OWSOperation	25
OWSOperationsMetadata	26
OWSRequest	27
OWSServiceIdentification	27
OWSServiceProvider	28
OWSUtils	29
OWSWGSS84BoundingBox	30
PropertyIsBetween	30
PropertyIsEqualTo	31
PropertyIsGreaterThan	32
PropertyIsGreaterThanOrEqualTo	32
PropertyIsLessThan	33
PropertyIsLessThanOrEqualTo	34
PropertyIsLike	34
PropertyIsNotEqualTo	35
PropertyIsNull	36
registerOWSNamespace	36

setOWSNamespace	37
UnaryLogicOpType	37
WFSCapabilities	38
WFSClient	38
WFSDescribeFeatureType	39
WFSFeatureType	40
WFSFeatureTypeElement	41
WFSGetFeature	42
WMSCapabilities	42
WMSClient	43
WMSGetFeatureInfo	44
WMSLayer	45
WPSBoundingBoxData	46
WPSCapabilities	46
WPSClient	47
WPSComplexData	48
WPSComplexInputDescription	49
WPSComplexOutputDescription	50
WPSDescribeProcess	50
WPSDescriptionParameter	51
WPSException	52
WPSExecute	52
WPSExecuteResponse	53
WPSFormat	54
WPSInput	54
WPSInputDescription	55
WPSLiteralData	56
WPSLiteralInputDescription	56
WPSLiteralOutputDescription	57
WPSOutput	58
WPSOutputDescription	58
WPSParameter	59
WPSProcess	60
WPSProcessDescription	60
WPSResponseDocument	61
WPSStatus	62

Index**63**

And

*And***Description**

And

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an And operator

Methods

`new(...)` This method is used to instantiate an And operator

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

Examples

```
expr1 <- PropertyIsEqualTo$new(PropertyName = "property1", Literal = "value1")
expr2 <- PropertyIsEqualTo$new(PropertyName = "property2", Literal = "value2")
and <- And$new(expr1,expr2)
and_xml <- and$encode() #see how it looks like in XML
```

BBOX

BBOX

Description

BBOX

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an BBOX

Methods

`new(bbox, srsName)` This method is used to instantiate an BBOX

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

Examples

```
bbox <- OWSUtils$toBBOX(-180,-90,180,90)
expr <- BBOX$new(bbox)
expr_xml <- expr$encode() #see how it looks like in XML
```

BinaryComparisonOpType

BinaryComparisonOpType

Description

BinaryComparisonOpType

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an BinaryComparisonOpType

Methods

`new(element, namespacePrefix, PropertyName, Literal, matchCase)` This method is used to instantiate an BinaryComparisonOpType

Note

abstract super class of all the property operation classes

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

BinaryLogicOpType *BinaryLogicOpType*

Description

BinaryLogicOpType

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an BinaryLogicOpType

Methods

`new(...)` This method is used to instantiate an BinaryLogicOpType

Note

abstract super class of all the binary logical operation classes

CSWCapabilities *CSWCapabilities*

Description

CSWCapabilities

Format

[R6Class](#) object.

Value

Object of [R6Class](#) with methods for interfacing an OGC Catalogue Service for the Web (CSW) Get Capabilities document.

Methods

`new(url, version, client logger)` This method is used to instantiate a CSWCapabilities object

Note

Class used to read a CSWCapabilities document. The use of CSWClient is recommended instead to benefit from the full set of capabilities associated to a CSW server.

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

CSWClient

CSWClient

Description

CSWClient

Format

[R6Class](#) object.

Value

Object of [R6Class](#) with methods for interfacing an OGC Catalogue Service for the Web.

Methods

`new(url, serviceVersion, user, pwd, token, headers, logger, ...)` This method is used to instantiate a CSWClient with the url of the OGC service. Authentication is supported using basic auth (using user/pwd arguments), bearer token (using token argument), or custom (using headers argument). By default, the logger argument will be set to NULL (no logger). This argument accepts two possible values: INFO: to print only **ows4R** logs, DEBUG: to print more verbose logs

`getCapabilities()` Get service capabilities.

`reloadCapabilities()` Reload service capabilities

`describeRecord(namespace, ...)` Describe records. Retrieves the XML schema for CSW records. By default, returns the XML schema for the CSW records (<http://www.opengis.net/cat/csw/2.0.2>). For other schemas, specify the outputSchema required, e.g. <http://www.isotc211.org/2005/gmd> for ISO 19115/19139 schema

`getRecordById(id, elementSetName, ...)` Get a record by Id. By default, the record will be returned following the CSW schema (<http://www.opengis.net/cat/csw/2.0.2>). For other schemas, specify the outputSchema required, e.g. <http://www.isotc211.org/2005/gmd> for ISO 19115/19139 records. The parameter elementSetName should among values "full", "brief", "summary". The default "full" corresponds to the full metadata sheet returned. "brief" and "summary" will contain only a subset of the metadata content.

`getRecords(query, maxRecords, maxRecordsPerRequest)` Get records based on a query, object of class `CSWQuery`. The maximum number of records can be set either for the full query (`maxRecords`) or per request (`maxRecordsPerRequest`, default set to 10 records) considering this operation is paginated. By default, the record will be returned following the CSW schema (<http://www.opengis.net/cat/csw/2.0.2>). For other schemas, specify the `outputSchema` required, e.g. <http://www.isotc211.org/2005/gmd> for ISO 19115/19139 records.

`transaction(type, record, recordProperty, constraint)` Generic transaction method. Used for inserting, updating or deleting metadata using the transactional CSW service. The `type` gives the type of transaction (Insert, Update, or Delete). The `record`

`insertRecord(record, ...)` Inserts a new record

`updateRecord(record, recordProperty, constraint, ...)` Updates an existing record. It can be a partial update by specifying a `recordProperty`. A `constraint` (object of class `CSWConstraint`) can be specified.

`deleteRecord(record, constraint, ...)` Deletes an existing record. A `constraint` (object of class `CSWConstraint`) can be specified to limit the deletion to some records.

`deleteRecordById(record, ...)` Deletes an existing record by identifier (`constraint` used to identify the record based on its identifier).

`harvestRecord(sourceUrl, resourceType)` Harvests a single record from a `sourceUrl`, given a `resourceType` (by default "<http://www.isotc211.org/2005/gmd>").

`harvestNode(url, query, resourceType, sourceBaseUrl)` Harvests a CSW node (having its endpoint defined by an `url`). A `query` (object of class `CSWQuery`) can be specified if needed to restrain the harvesting to a subset. The `resourceType` defines the type of resources to be harvested (by default "<http://www.isotc211.org/2005/gmd>")

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

Examples

```
#example based on CSW endpoint responding at http://localhost:8000/csw
csw <- CSWClient$new("http://localhost:8000/csw", serviceVersion = "2.0.2")

#get capabilities
caps <- csw$getCapabilities()

#get records
records <- csw$getRecords()

#get record by id
record <- csw$getRecordById("my-metadata-id")

#Advanced examples at https://github.com/eblondel/ows4R/wiki#csw
```

CSWConstraint	<i>CSWConstraint</i>
---------------	----------------------

Description

CSWConstraint

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an CSW Constraint

Methods

`new(cqlText, filter, serviceVersion)` This method is used to instantiate an CSWConstraint object.

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

Examples

```
filter <- OGCFilter$new( PropertyIsEqualTo$new("apiso:Identifier", "12345") )
cons <- CSWConstraint$new(filter = filter)
cons_xml <- cons$encode() #how it looks like in XML
```

CSWDescribeRecord	<i>CSWDescribeRecord</i>
-------------------	--------------------------

Description

CSWDescribeRecord

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a CSW DescribeRecord request

Methods

`new(capabilities, op, url, version, namespace, logger, ...)` This method is used to instantiate a CSWDescribeRecord object

Note

Class used internally by **ows4R** to trigger a CSW DescribeRecord request

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

CSWGetRecordById

CSWGetRecordById

Description

CSWGetRecordById

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a CSW GetRecordById request

Methods

`new(capabilities, op, url, serviceVersion, user, pwd, token, headers, id, elementSetName, logger, ...)`
This method is used to instantiate a CSWGetRecordById object

Note

Class used internally by **ows4R** to trigger a CSW GetRecordById request

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

CSWGetRecords	<i>CSWGetRecords</i>
---------------	----------------------

Description

CSWGetRecords

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a CSW GetRecords request

Methods

`new(capabilities, op, url, serviceVersion, user, pwd, token, headers, query, logger, ...)`
This method is used to instantiate a CSWGetRecords object

Note

Class used internally by **ows4R** to trigger a CSW GetRecords request

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

CSWHarvest	<i>CSWHarvest</i>
------------	-------------------

Description

CSWHarvest

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a CSW Harvest request

Methods

`new(capabilities, op, url, serviceVersion, user, pwd, source, resourceType, resourceFormat, logger, ...)`
This method is used to instantiate a CSWHarvest object

Note

Class used internally by **ows4R** to trigger a CSW Harvest request

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

CSWQuery

CSWQuery

Description

CSWQuery

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an CSW Query

Methods

`new(elementSetName, constraint, typeNames, serviceVersion)` This method is used to instantiate an CSWQuery object. The `elementSetName` can be either "full" (default), "brief" or "summary". A constraint CSWConstraint can be defined for the query. The `typeNames` indicates to query (default "csw:Record"). The `serviceVersion` gives the CSW service version (default "2.0.2")

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

Examples

```
#CSWQuery - elementSetName
query_full <- CSWQuery$new()
query_brief <- CSWQuery$new(elementSetName = "brief")
query_summary <- CSWQuery$new(elementSetName = "summary")

#CSWQuery - cqlText with title
cons <- CSWConstraint$new(cqlText = "dc:title like '%ips%'")
query <- CSWQuery$new(constraint = cons)

#CSW 2.0.2 - Query - Filter / AnyText
filter <- OGCFilter$new( PropertyIsLike$new("csw:AnyText", "%Physio%"))
cons <- CSWConstraint$new(filter = filter)
query <- CSWQuery$new(constraint = cons)
```

```

#CSW 2.0.2 - Query - Filter / AnyText Equal
filter <- OGCFilter$new( PropertyIsEqualTo$new("csw:AnyText", "species"))
cons <- CSWConstraint$new(filter = filter)
query <- CSWQuery$new(constraint = cons)

#CSW 2.0.2 - Query - Filter / AnyText And Not
filter <- OGCFilter$new(And$new(
  PropertyIsLike$new("csw:AnyText", "%lorem%"),
  PropertyIsLike$new("csw:AnyText", "%ipsum%"),
  Not$new(
    PropertyIsLike$new("csw:AnyText", "%dolor%")
  )
))
cons <- CSWConstraint$new(filter = filter)
query <- CSWQuery$new(constraint = cons)

#CSW 2.0.2 - Query - Filter / AnyText And nested Or
filter <- OGCFilter$new(And$new(
  PropertyIsEqualTo$new("dc:title", "Aliquam fermentum purus quis arcu"),
  PropertyIsEqualTo$new("dc:format", "application/pdf"),
  Or$new(
    PropertyIsEqualTo$new("dc:type", "http://purl.org/dc/dcmitype/Dataset"),
    PropertyIsEqualTo$new("dc:type", "http://purl.org/dc/dcmitype/Service"),
    PropertyIsEqualTo$new("dc:type", "http://purl.org/dc/dcmitype/Image"),
    PropertyIsEqualTo$new("dc:type", "http://purl.org/dc/dcmitype/Text")
  )
))
cons <- CSWConstraint$new(filter = filter)
query <- CSWQuery$new(elementSetName = "brief", constraint = cons)

#CSW 2.0.2 - Query - Filter / BBOX
bbox <- matrix(c(-180,180,-90,90), nrow = 2, ncol = 2, byrow = TRUE,
  dimnames = list(c("x", "y"), c("min","max")))
filter <- OGCFilter$new( BBOX$new(bbox = bbox) )
cons <- CSWConstraint$new(filter = filter)
query <- CSWQuery$new(elementSetName = "brief", constraint = cons)

```

CSWRecordProperty

CSWRecordProperty

Description

CSWRecordProperty

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an CSW RecordProperty

Methods

`new(name, value, cswVersion)` This method is used to instantiate an CSWRecordProperty object.

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

Examples

```
rp <- CSWRecordProperty$new(name = "NAME", value = "VALUE")
rp_xml <- rp$encode() #see how it looks in XML
```

CSWTransaction

CSWTransaction

Description

CSWTransaction

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a CSW Transaction request

Methods

`new(capabilities, op, url, serviceVersion, type, user, pwd, token, headers, record, recordProperty, const)`
This method is used to instantiate a CSWTransaction object

Note

Class used internally by **ows4R** to trigger a CSW Transaction request

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

getOWSNamespace *getOWSNamespace*

Description

getOWSNamespace gets a namespace given its id

Usage

getOWSNamespace(id)

Arguments

id namespace prefix

Author(s)

Emmanuel Blondel, <emmanuel.blondel1@gmail.com>

Examples

getOWSNamespace("GMD")

*getOWSNamespace*s *getOWSNamespace*s

Description

*getOWSNamespace*s gets the list of namespaces registered

Usage

*getOWSNamespace*s()

Author(s)

Emmanuel Blondel, <emmanuel.blondel1@gmail.com>

Examples

*getOWSNamespace*s()

Not	<i>Not</i>
-----	------------

Description

Not

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an Not

Methods

`new(...)` This method is used to instantiate an Not operator

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

Examples

```
expr <- PropertyIsEqualTo$new(PropertyName = "property", Literal = "value")
not <- Not$new(expr)
not_xml <- not$encode() #see how it looks like in XML
```

OGCAbstractObject	<i>OGCAbstractObject</i>
-------------------	--------------------------

Description

OGCAbstractObject

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an OGCAbstractObject

Methods

`new(xml, element, namespacePrefix, attrs, defaults, wrap, logger)` This method is used to instantiate an `OGCAbstractObject`

`getClassName()` Get class name

`getClass()` Get class

`encode(addNS, geometa_validate, geometa_inspire, geometa_inspireValidator)` Encodes as XML. The `addNS` controls the addition of XML namespaces. Extra parameters related to **geometa** objects: `geometa_validate` (TRUE by default) and `geometa_inspire` (FALSE by default) can be used to perform ISO and INSPIRE validation respectively. In that case on object of class `geometa::INSPIREMetadataValidator`, with a proper user API key, should be specified as `geometa_inspireValidator` argument.

Note

abstract class used by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

OGCExpression

OGCExpression

Description

OGCExpression

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an OGC Expression

Methods

`new(element, namespacePrefix, attrs, defaults, exprVersion)` This method is used to instantiate an `OGCExpression` object

`setExprVersion(exprVersion)` Set expression version

`getExprVersion()` Get expression version

Note

abstract class

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

OGCFilter

OGCFilter

Description

OGCFilter

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an OGC Filter

Methods

`new(expr, filterVersion)` This method is used to instantiate an OGCFilter object. The unique argument should be an object of class [OGCExpression](#)

`setFilterVersion(filterVersion)` Set filter version

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

Examples

```
expr <- PropertyIsEqualTo$new(PropertyName = "property", Literal = "value")
not <- Not$new(expr)
not_xml <- not$encode() #see how it looks like in XML
```

Or

Or

Description

Or

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an Or operator

Methods

`new(...)` This method is used to instantiate an Or operator

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

Examples

```
expr1 <- PropertyIsEqualTo$new(PropertyName = "property1", Literal = "value1")
expr2 <- PropertyIsEqualTo$new(PropertyName = "property2", Literal = "value2")
or <- Or$new(expr1,expr2)
or_xml <- or$encode() #see how it looks like in XML
```

ows4R

Interface to OGC Web-Services

Description

Provides an Interface to Web-Services defined as standards by the Open Geospatial Consortium (OGC), including Web Feature Service (WFS) for vector data, Catalogue Service (CSW) for ISO/OGC metadata, and associated standards such as the common web-service specification (OWS) and OGC Filter Encoding. Partial support is provided for the Web Map Service (WMS) and Web Processing Service (WPS). The purpose is to add support for additional OGC service standards such as Web Coverage Service (WCS), Web Coverage Processing Service (WCPS) or OGC API.

Details

Package: ows4R
Type: Package
Version : 0.2-1
Date: 2022-01-26
License: MIT
LazyLoad: yes

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

OWSBoundingBox

OWSBoundingBox

Description

OWSBoundingBox

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an OGC Bounding Box

Methods

`new(xml, element, namespacePrefix, owsVersion, serviceVersion, logger)` This method is used to instantiate an OWSBoundingBox object

`getBBOX()` Get BBOX as object of class `bbox` from **sf** package

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

OWSCapabilities

OWSGetCapabilities

Description

OWSGetCapabilities

Format

[R6Class](#) object.

Value

Object of [R6Class](#) with methods for interfacing an abstract OWS Get Capabilities document.

Methods

`new(url, service, owsVersion, serviceVersion, logger, ...)` This method is used to instantiate a OWSGetCapabilities object

`setClient(client)` Internal method to self-assign the parent client

`getUrl()` Get URL

`getVersion()` Get version

`getRequest()` Get request

`getServiceIdentification()` Get the service identification

`getServiceProvider()` Get the service provider

`getOperationsMetadata()` Get the service operations metadata

Note

abstract class used by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

OWSClient

OWSClient

Description

OWSClient

Format

R6Class object.

Value

Object of R6Class with methods for interfacing a Common OGC web-service.

Fields

url the Base url of OWS service

version the version of OWS service

Methods

new(url, service, serviceVersion, user, pwd, logger) This method is used to instantiate a OWSClient with the url of the OGC service. Authentication is supported using basic auth (using user/pwd arguments), bearer token (using token argument), or custom (using headers argument). By default, the logger argument will be set to NULL (no logger). This argument accepts two possible values: INFO: to print only **ows4R** logs, DEBUG: to print more verbose logs

getUrl() Get the service URL

getVersion() Get the service version

getCapabilities() Get the service capabilities

getUser() Get user

getPwd() Get password

getToken() Get token

getHeaders() Get headers

NoteAbstract class used internally by **ows4R****Author(s)**

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

OWSCodeType	<i>OWSCodeType</i>
-------------	--------------------

Description

OWSCodeType

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an OWS CodeType

Methods

`new(expr)` This method is used to instantiate an OWSCodeType object. The unique argument should be an object of class character

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

OWSGetCapabilities	<i>OWSGetCapabilities</i>
--------------------	---------------------------

Description

OWSGetCapabilities

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a GetCapabilities request

Methods

`new(element, namespacePrefix, url, service, version, ...)` This method is used to instantiate a OWSGetCapabilities object

Note

Abstract class used internally by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

OWSHttpRequest

OWSHttpRequest

Description

OWSHttpRequest

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a generic OWS http request

Methods

`new(capabilities, op, type, url, request, user, pwd, namedParams, attrs, contentType, mimeType, logger)`

This method is used to instantiate a object for doing an OWS HTTP request

`getCapabilities()` Get the capabilities

`getRequest()` Get the request payload

`getRequestHeaders()` Get the request headers

`getStatus()` Get the request status code

`getResponse()` Get the request response

`getException()` Get the exception (in case of request failure)

`getResult()` Get the result TRUE if the request is successful, FALSE otherwise

Note

Abstract class used internally by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

OwsNamespace

OwsNamespace

Description

OwsNamespace

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an OWS Namespace

Methods

`new(id, uri)` This method is used to instantiate an OWS namespace

Note

class used internally by ows4R for specifying XML namespaces

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

OwsOperation

OwsOperation

Description

OwsOperation

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an OGC Operation

Methods

`new(xmlObj, owsVersion, serviceVersion)` This method is used to instantiate an OWSOperation object

`getName()` Get name

`getParameters()` Get the list of parameters

`getParameter(name)` Get a given parameter

Note

Abstract class used internally by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

OWSOperationsMetadata *OWSOperationsMetadata*

Description

OWSOperationsMetadata

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an OGC Operations Metadata

Methods

`new(xmlObj, serviceVersion)` This method is used to instantiate a OWSOperationsMetadata object

`getOperations()` Get operations

Note

Abstract class used internally by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

OWSRequest

OWSRequest

Description

OWSRequest

Format

[R6Class](#) object.

Value

Object of [R6Class](#) modelling a OWS Service Capability Request

Methods

`new(xmlObj, capabilities, version, logger)` This method is used to instantiate a OWSRequest object

`getName()` Get request name

`getFormats()` Get request formats

Note

Abstract class used by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

OWSServiceIdentification

OWSServiceIdentification

Description

OWSServiceIdentification

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an OGC Service Identification

Methods

`new(xmlObj, serviceVersion)` This method is used to instantiate a `OWSServiceIdentification` object

`getName()` Get service name

`getTitle()` Get service title

`getAbstract()` Get service abstract

`getKeywords()` Get service keywords

`getOnlineResource()` Get service online resource

`getServiceType` Get service type

`getServiceTypeVersion` Get service type version

Note

Abstract class used internally by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

OWSServiceProvider *OWSServiceProvider*

Description

OWSServiceProvider

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an OGC Service Provider

Methods

`new(xmlObj, owsVersion, serviceVersion)` This method is used to instantiate a `OWSService-Provider` object

`getProviderName()` Get the provider name

`getProviderSite()` Get the provide site

`getServiceContact()` Get the service contact, as object of class `ISOResponsibleParty` from package **geometa**

Note

Abstract class used internally by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

OWSUtils

OWSUtils

Description

OWSUtils

Usage

OWSUtils

Format

[R6Class](#) object.

Value

set of OWS Utilities

Static methods

`getNamespaces()` Get the namespaces associated to a given XML object
`findNamespace(namespaces, id, uri)` Finds a namespace by id or by URI
`toBBOX(xmin, xmax, ymin, ymax)` Creates a bbox matrix from min/max x/y coordinates
`findP4s(srsName, morphToESRI)` Finds the PROJ4 string definition for a given srsName
`toCRS(srsName)` Converts a srsName into a CRS object
`toEPSG(crs)` Get the EPSG code from a CRS object
`getAspectRatio(bbox)` Get the aspect ratio for a given bbox

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

Examples

```
#toBBOX
bbox <- OWSUtils$toBBOX(-180,-90,180,90)

#toCRS
crs <- OWSUtils$toCRS("EPSG:4326")
```

OWSWGS84BoundingBox *OWSWGS84BoundingBox*

Description

OWSWGS84BoundingBox

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an OGC WS84 BoundingBox

Methods

`new(xml, owsVersion, serviceVersion, logger)` This method is used to instantiate an OWSWGS84BoundingBox object

Note

Class used internally by **geometa**

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

PropertyIsBetween *PropertyIsBetween*

Description

PropertyIsBetween

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an PropertyIsBetween

Methods

`new(PropertyName, lower, upper)` This method is used to instantiate an PropertyIsBetween

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

Examples

```
expr <- PropertyIsBetween$new(PropertyName = "property", lower = 1, upper = 10)
expr_xml <- expr$encode() #see how it looks like in XML
```

PropertyIsEqualTo	<i>PropertyIsEqualTo</i>
-------------------	--------------------------

Description

PropertyIsEqualTo

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an PropertyIsEqualTo

Methods

`new(PropertyName, Literal, matchCase)` This method is used to instantiate an PropertyIsEqualTo

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

Examples

```
expr <- PropertyIsEqualTo$new(PropertyName = "property", Literal = "value")
expr_xml <- expr$encode() #see how it looks like in XML
```

PropertyIsGreaterThan *PropertyIsGreaterThan*

Description

PropertyIsGreaterThan

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an PropertyIsGreaterThan

Methods

`new(PropertyName, Literal, matchCase)` This method is used to instantiate an PropertyIsGreaterThan

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

Examples

```
expr <- PropertyIsGreaterThan$new(PropertyName = "property", Literal = "value")
expr_xml <- expr$encode() #see how it looks like in XML
```

PropertyIsGreaterThanOrEqualTo

PropertyIsGreaterThanOrEqualTo

Description

PropertyIsGreaterThanOrEqualTo

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an PropertyIsGreaterThanOrEqualTo

Methods

`new(PropertyName, Literal, matchCase)` This method is used to instantiate an `PropertyIsGreaterThanOrEqualTo`

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

Examples

```
expr <- PropertyIsGreaterThanOrEqualTo$new(PropertyName = "property", Literal = "value")
expr_xml <- expr$encode() #see how it looks like in XML
```

PropertyIsLessThan *PropertyIsLessThan*

Description

PropertyIsLessThan

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an `PropertyIsLessThan`

Methods

`new(PropertyName, Literal, matchCase)` This method is used to instantiate an `PropertyIsLessThan`

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

Examples

```
expr <- PropertyIsLessThan$new(PropertyName = "property", Literal = "value")
expr_xml <- expr$encode() #see how it looks like in XML
```

PropertyIsLessThanOrEqualTo
PropertyIsLesserThanOrEqualTo

Description

PropertyIsLesserThanOrEqualTo

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an PropertyIsLesserThanOrEqualTo

Methods

`new(PropertyName, Literal, matchCase)` This method is used to instantiate an PropertyIsLesserThanOrEqualTo

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

Examples

```
expr <- PropertyIsLessThanOrEqualTo$new(PropertyName = "property", Literal = "value")
expr_xml <- expr$encode() #see how it looks like in XML
```

PropertyIsLike *PropertyIsLike*

Description

PropertyIsLike

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an PropertyIsLike

Methods

```
new(PropertyOperator, PropertyName, Literal, escapeChar, singleChar, wildCard, matchCase)
```

This method is used to instantiate an PropertyIsLike

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

Examples

```
expr <- PropertyIsLike$new(PropertyName = "property", Literal = "value")  
expr_xml <- expr$encode() #see how it looks like in XML
```

PropertyIsNotEqualTo *PropertyIsNotEqualTo*

Description

PropertyIsNotEqualTo

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an PropertyIsNotEqualTo

Methods

```
new(PropertyName, Literal, matchCase)
```

This method is used to instantiate an PropertyIsNotEqualTo

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

Examples

```
expr <- PropertyIsNotEqualTo$new(PropertyName = "property", Literal = "value")  
expr_xml <- expr$encode() #see how it looks like in XML
```

PropertyIsNull	<i>PropertyIsNull</i>
----------------	-----------------------

Description

PropertyIsNull

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an PropertyIsNull

Methods

`new(PropertyName)` This method is used to instantiate an PropertyIsNull

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

Examples

```
expr <- PropertyIsNull$new(PropertyName = "property")
expr_xml <- expr$encode() #see how it looks like in XML
```

registerOWSNamespace	<i>registerOWSNamespace</i>
----------------------	-----------------------------

Description

registerOWSNamespace allows to register a new namespace in **ows4R**

Usage

```
registerOWSNamespace(id, uri, force)
```

Arguments

<code>id</code>	prefix of the namespace
<code>uri</code>	URI of the namespace
<code>force</code>	logical parameter indicating if registration has be to be forced in case the identified namespace is already registered

Author(s)

Emmanuel Blondel, <emmanuel.blondel1@gmail.com>

Examples

```
registerOWSNamespace(id = "myprefix", uri = "http://someuri")
```

setOWSNamespace	<i>setOWSNamespace</i>
-----------------	------------------------

Description

setOWSNamespace

Usage

```
setOWSNamespace()
```

UnaryLogicOpType	<i>UnaryLogicOpType</i>
------------------	-------------------------

Description

UnaryLogicOpType

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an UnaryLogicOpType

Methods

`new(...)` This method is used to instantiate an UnaryLogicOpType

Note

abstract super class of all the unary logical operation classes

WFSCapabilities

WFSCapabilities

Description

WFSCapabilities

Format

[R6Class](#) object.

Value

Object of [R6Class](#) with methods for interfacing an OGC Web Feature Service Get Capabilities document.

Methods

`new(url, version, logger)` This method is used to instantiate a WFSCapabilities object

`getFeatureTypes(pretty)` List the feature types available. If pretty is TRUE, the output will be an object of class `data.frame`

`findFeatureTypeByName(name, exact)` Find feature type(s) by name.

Note

Class used to read a WFSCapabilities document. The use of WFSCClient is recommended instead to benefit from the full set of capabilities associated to a WFS server.

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

WFSCClient

WFSCClient

Description

WFSCClient

Format

[R6Class](#) object.

Value

Object of [R6Class](#) with methods for interfacing an OGC Web Feature Service.

Methods

`new(url, serviceVersion, user, pwd, logger)` This method is used to instantiate a `WFSCClient` with the `url` of the OGC service. Authentication is supported using basic auth (using `user/pwd` arguments), bearer token (using `token` argument), or custom (using `headers` argument). By default, the `logger` argument will be set to `NULL` (no logger). This argument accepts two possible values: `INFO`: to print only **ows4R** logs, `DEBUG`: to print more verbose logs

`getCapabilities()` Get service capabilities. Inherited from `OWS Client`

`reloadCapabilities()` Reload service capabilities

`describeFeatureType(typeName)` Get the description of a given `featureType`

`getFeatures(typeName, ...)` Retrieves the features for a given feature type.

`getFeatureTypes(pretty)` List the feature types available. If `pretty` is `TRUE`, the output will be an object of class `data.frame`

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

Examples

```
#example based on a WFS endpoint responding at http://localhost:8080/geoserver/wfs
wfs <- WFSCClient$new("http://localhost:8080/geoserver/wfs", serviceVersion = "1.1.1")

#get capabilities
caps <- wfs$getCapabilities()

#find feature type
ft <- caps$findFeatureTypeByName("mylayer")
if(length(ft)>0){
  data <- ft$getFeatures()
  data_with_filter <- ft$getFeatures(cql_filter = "somefilter")
}

#Advanced examples at https://github.com/eblondel/ows4R/wiki#wfs
```

WFSDescribeFeatureType

WFSDescribeFeatureType

Description

WFSDescribeFeatureType

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a WFS DescribeFeatureType request

Methods

`new(capabilities, op, url, version, typeName, logger, ...)` This method is used to instantiate a WFSDescribeFeatureType object

Note

Abstract class used by **ows4R** to trigger a WFS DescribeFeatureType request

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WFSFeatureType

WFSFeatureType

Description

WFSFeatureType

Format

[R6Class](#) object.

Value

Object of [R6Class](#) modelling a WFS feature type

Methods

`new(xmlObj, capabilities, version, logger)` This method is used to instantiate a WFSFeatureType object

`getName()` Get feature type name

`getTitle()` Get feature type title

`getAbstract()` Get feature type abstract

`getKeywords()` Get feature type keywords

`getDefaultCRS()` Get feature type default CRS

`getBoundingBox()` Get feature type bounding box

`getDescription(pretty)` Get feature type description. If pretty is TRUE, the output will be an object of class `data.frame`

`getFeatures()` Get features.

Note

Abstract class used by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WFSFeatureTypeElement *WFSFeatureTypeElement*

Description

WFSFeatureTypeElement

Format

[R6Class](#) object.

Value

Object of [R6Class](#) modelling a WFS feature type element

Methods

`new(xmlObj)` This method is used to instantiate a WFSFeatureTypeElement object

`getMinOccurs()` Get min occurs

`getMaxOccurs()` Get max occurs

`isNillable()` Get TRUE if nillable, FALSE otherwise

`getName()` Get element name

`getType()` Get element type

Note

Abstract class used by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WFSGetFeature

WFSGetFeature

Description

WFSGetFeature

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a WFS GetFeature request

Methods

`new(capabilities, op, url, version, typeName, outputFormat, logger, ...)` This method is used to instantiate a WFSGetFeature object

Note

Abstract class used by **ows4R** to trigger a WFS DescribeFeatureType request

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WMSCapabilities

WMSCapabilities

Description

WMSCapabilities

Format

[R6Class](#) object.

Value

Object of [R6Class](#) with methods for interfacing an OGC Web Map Service Get Capabilities document.

Methods

`new(url, version, logger, ...)` This method is used to instantiate a `WMSCapabilities` object

`getRequests(pretty)` List the requests available. If `pretty` is `TRUE`, the output will be an object of class `data.frame`

`getRequestNames()` List the request names available.

`getLayers(pretty)` List the layers available. If `pretty` is `TRUE`, the output will be an object of class `data.frame`

`findLayerByName(name, exact)` Find layer(s) by name.

Note

Class used to read a `WMSCapabilities` document. The use of `WMSCient` is recommended instead to benefit from the full set of capabilities associated to a WMS server.

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

WMSCient

WMSCient

Description

WMSCient

Format

[R6Class](#) object.

Value

Object of [R6Class](#) with methods for interfacing an OGC Web Map Service.

Methods

`new(url, serviceVersion, user, pwd, logger)` This method is used to instantiate a `WMSCient` with the `url` of the OGC service. Authentication is supported using basic auth (using `user/pwd` arguments), bearer token (using `token` argument), or custom (using `headers` argument). By default, the `logger` argument will be set to `NULL` (no logger). This argument accepts two possible values: `INFO`: to print only **ows4R** logs, `DEBUG`: to print more verbose logs

`getCapabilities()` Get service capabilities. Inherited from `OWS Client`

`reloadCapabilities()` Reload service capabilities

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

Examples

```
#example based on a WMS endpoint responding at http://localhost:8080/geoserver/wms
wms <- WMSClient$new("http://localhost:8080/geoserver/wms", serviceVersion = "1.1.1")

#get capabilities
caps <- wms$getCapabilities()

#get feature info

#Advanced examples at https://github.com/eblondel/ows4R/wiki#wms
```

WMSGetFeatureInfo

WMSGetFeatureInfo

Description

WMSGetFeatureInfo

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a WMS GetFeatureInfo request

Methods

`new(capabilities, op, url, version, layers, srs, styles, feature_count, x, y, width, height, bbox, info_fc)`
This method is used to instantiate a WMSGetFeatureInfo object

Note

Abstract class used by **ows4R** to trigger a WMS GetFeatureInfo request

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

WMSLayer

WMSLayer

Description

WMSLayer

Format

[R6Class](#) object.

Value

Object of [R6Class](#) modelling a WMS layer

Methods

`new(xmlObj, capabilities, version, logger)` This method is used to instantiate a WMSLayer object

`getName()` Get layer name

`getTitle()` Get layer title

`getAbstract()` Get layer abstract

`getKeywords()` Get layer keywords

`getDefaultCRS()` Get layer default CRS

`getBoundingBox()` Get layer bounding box

`getBoundingBoxSRS()` Get layer bounding box SRS

`getBoundingBoxCRS()` Get layer bounding box CRS

`getStyle()` Get layer style

`getDimensions()` Get layer dimensions

`getTimeDimension()` Get layer time dimension

`getElevationDimension()` Get layer elevation dimension

`getFeatureInfo(srs, styles, feature_count, x, y, width, height, bbox, info_format)` Get layer feature info

Note

Abstract class used by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSBoundingBoxData *WPSBoundingBoxData*

Description

WPSBoundingBoxData

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling an OGC WPS BoundingBox data

Methods

`new(xml, owsVersion, serviceVersion, logger)` This method is used to instantiate an WPSBoundingBoxData object

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSCapabilities *WPSCapabilities*

Description

WPSCapabilities

Format

[R6Class](#) object.

Value

Object of [R6Class](#) with methods for interfacing an OGC Web Processing Service (WPS) Get Capabilities document.

Methods

`new(url, version, logger, ...)` This method is used to instantiate a WPSCapabilities object

`getProcesses(pretty, full)` Return the list of processes offered by the service capabilities.
 pretty allows to control the type output. If TRUE, a data.frame will be returned. When prettified output, it is also possible to get a 'full' description of the process by setting full = TRUE in which case a the WPS client will request a process description (with more information about the process) for each process listed in the capabilities.

`describeProcess(identifier)` Get the description of a process, given its identifier, returning an object of class WPSProcessDescription

`execute(identifier, dataInputs, responseForm, storeExecuteResponse, lineage, status, update, updateInte)`
 Execute a process, given its identifier

Note

Class used to read a WPSCapabilities document. The use of WPSClient is recommended instead to benefit from the full set of capabilities associated to a WPS server.

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

 WPSClient

WPSClient

Description

WPSClient

Format

[R6Class](#) object.

Value

Object of [R6Class](#) with methods for interfacing an OGC Web Processing Service.

Methods

`new(url, serviceVersion, user, pwd, logger)` This method is used to instantiate a WPSClient with the url of the OGC service. Authentication is supported using basic auth (using user/pwd arguments), bearer token (using token argument), or custom (using headers argument). By default, the logger argument will be set to NULL (no logger). This argument accepts two possible values: INFO: to print only **ows4R** logs, DEBUG: to print more verbose logs

`getCapabilities()` Get service capabilities. Inherited from OWS Client

`reloadCapabilities()` Reload service capabilities

`getProcesses(pretty, full)` Return the list of processes offered by the service capabilities. `pretty` allows to control the type output. If `TRUE`, a `data.frame` will be returned. When prettified output, it is also possible to get a 'full' description of the process by setting `full = TRUE` in which case a the WPS client will request a process description (with more information about the process) for each process listed in the capabilities.

`describeProcess(identifier)` Get the description of a process, given its identifier, returning an object of class `WPSProcessDescription`

`execute(identifier, dataInputs, responseForm, storeExecuteResponse, lineage, status, update, updateInte`
Execute a process, given its identifier

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

Examples

```
#example based on a WPS endpoint responding at http://localhost:8080/geoserver/wps
wps <- WPSClient$new("http://localhost:8080/geoserver/wps", serviceVersion = "1.0.0")

#get capabilities
caps <- wps$getCapabilities()
```

WPSComplexData

WPSComplexData

Description

WPSComplexData

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a WPS Complex Data

Methods

`new(xml, value, schema, mimeType)` This method is used to instantiate a `WPSComplexData` object

`decode(xml)` Decodes from XML

`checkValidity(parameterDescription)` Check the object against a parameter description inherited from a WPS process description, object of class `WPSComplexInputDescription`. If not valid, the function will raise an error.

getFeatures() Returns features associates with output, if the output si made of a GML feature collection

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

WPSComplexInputDescription
WPSComplexInputDescription

Description

WPSComplexInputDescription

Format

[R6Class](#) object.

Value

Object of [R6Class](#) modelling a WPS process complex input description

Methods

new(xml, version, logger) This method is used to instantiate a WPSComplexInputDescription object

Note

Class used internally by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

WPSComplexOutputDescription
WPSComplexOutputDescription

Description

WPSComplexOutputDescription

Format

[R6Class](#) object.

Value

Object of [R6Class](#) modelling a WPS process complex input description

Methods

`new(xml, version, logger)` This method is used to instantiate a `WPSComplexOutputDescription` object

Note

Class used internally by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

WPSDescribeProcess *WPSDescribeProcess*

Description

WPSDescribeProcess

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a WPS DescribeProcess request

Methods

`new(capabilities, op, url, serviceVersion, identifier, logger, ...)` This method is used to instantiate a WPSDescribeProcess object

Note

Abstract class used by **ows4R** to trigger a WPS DescribeProcess request

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSDescriptionParameter

WPSDescriptionParameter

Description

WPSDescriptionParameter

Format

[R6Class](#) object.

Value

Object of [R6Class](#) modelling a WPS process input description parameter

Methods

`new(xml, version, logger)` This method is used to instantiate a WPSDescriptionParameter object

`getDataType()` Get data type

`getFormats()` Get formats

Note

Class used internally by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSException

WPSException

Description

WPSException

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a WPS Exception

Methods

`new(xml, serviceVersion)` This method is used to instantiate a WPSException object

`decode(xml)` Decodes WPS exception from XML

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSExecute

WPSExecute

Description

WPSExecute

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a WPS Execute request

Methods

`new(capabilities, op, url, serviceVersion, identifier, logger, ...)` This method is used to instantiate a WPSExecute object

`getProcessDescription()` Get process description

Note

Abstract class used by **ows4R** to trigger a WPS Execute request

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSEecuteResponse *WPSEecuteResponse*

Description

WPSEecuteResponse

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a WPS Execute response

Methods

`new(xml)` This method is used to instantiate a WPSEecuteResponse object

`getProcess()` Get the process

`getStatus()` Get the status

`getStatusLocation()` Get the status location

`getProcessOutputs()` Get the process output(s)

`decode()` Decodes WPS Execute response from XML

`update(verbose)` Updates the status based on the execute status location. Returns an object of class WPSEecuteResponse

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSFormat

WPSFormat

Description

WPSFormat

Format[R6Class](#) object.**Value**Object of [R6Class](#) modelling a WPS process input description**Methods**`new(xml, version, logger)` This method is used to instantiate a WPSFormat object`getMimeType()` Get mimetype`getEncoding()` Get encoding`getSchema()` Get schema`setIsDefault(default)` Set if default format or not`isDefault()` Is default format**Note**Class used internally by **ows4R****Author(s)**

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSInput*WPSInput*

Description

WPSInput

Format[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a WPS Input

Methods

`new(xml, identifier, data, serviceVersion)` This method is used to instantiate a WPSInput object

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSInputDescription *WPSInputDescription*

Description

WPSInputDescription

Format

[R6Class](#) object.

Value

Object of [R6Class](#) modelling a WPS process input description

Methods

`new(xml, capabilities, version, logger)` This method is used to instantiate a WPSInputDescription object

`getMinOccurs()` Get input min occurs

`getMaxOccurs()` Get input max occurs

Note

Class used internally by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSLiteralData

WPSLiteralData

Description

WPSLiteralData

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a WPS Literal Data

Methods

`new(xml, value, serviceVersion)` This method is used to instantiate a WPSLiteralData object

`decode(xml)` Decodes WPS input from XML

`checkValidity(parameterDescription)` Check the object against a parameter description inherited from a WPS process description, object of class WPSLiteralInputDescription. If not valid, the function will raise an error.

Author(s)

Emmanuel Blondel <emmanuel.blondell@gmail.com>

WPSLiteralInputDescription

WPSLiteralInputDescription

Description

WPSLiteralInputDescription

Format

[R6Class](#) object.

Value

Object of [R6Class](#) modelling a WPS process input description

Methods

`new(xml, version, logger)` This method is used to instantiate a `WPSLiteralInputDescription` object

`getDataType()` Get data type

`getDefaultValue()` Get default value

`getAllowedValues()` Get allowed values

Note

Class used internally by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSLiteralOutputDescription
WPSLiteralOutputDescription

Description

WPSLiteralOutputDescription

Format

[R6Class](#) object.

Value

Object of [R6Class](#) modelling a WPS process literal output description

Methods

`new(xml, version, logger)` This method is used to instantiate a `WPSLiteralOutputDescription` object

Note

Class used internally by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSOutput

WPSOutput

Description

WPSOutput

Format[R6Class](#) object.**Value**Object of [R6Class](#) for modelling a WPS Input**Methods**`new(xml, identifier, title, data, dataType)` This method is used to instantiate a WPSOutput object`decode(xml)` Decodes WPS output from XML`getData()` Get data, as object of class `WPSLiteralData` or `WPSComplexData``getDataValue()` Get data value, as shortcut method to get value associated to `WPSLiteralData` or `WPSComplexData`**Author(s)**

Emmanuel Blondel <emmanuel.blondell@gmail.com>

WPSOutputDescription*WPSOutputDescription*

Description

WPSOutputDescription

Format[R6Class](#) object.**Value**Object of [R6Class](#) modelling a WPS process input description**Methods**`new(xml, version, logger)` This method is used to instantiate a WPSOutputDescription object

Note

Class used internally by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSParameter

WPSParameter

Description

WPSParameter

Format

[R6Class](#) object.

Value

Object of [R6Class](#) modelling a WPS parameter

Methods

`new(xml, version, logger)` This method is used to instantiate a WPSParameter object

`getIdentifier()` Get input identifier

`getTitle()` Get input title

`getAbstract()` Get input abstract

Note

Abstract class used internally by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSProcess

WPSProcess

Description

WPSProcess

Format[R6Class](#) object.**Value**Object of [R6Class](#) modelling a WPS process**Methods**`new(xml, capabilities, version, logger)` This method is used to instantiate a WPSProcess object`getIdentifier()` Get process identifier`getTitle()` Get process title`getVersion()` Get process version`getDescription()` Get process description as object of class WPSProcessDescription`execute(dataInputs, responseForm, storeExecuteResponse, lineage, status, update, updateInterval)`
Executes the process.**Note**Class used internally by **ows4R****Author(s)**

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSProcessDescription *WPSProcessDescription*

Description

WPSProcessDescription

Format[R6Class](#) object.

Value

Object of [R6Class](#) modelling a WPS process description

Methods

`new(xml, version, logger)` This method is used to instantiate a WPSProcessDescription object

`getIdentifier()` Get process identifier

`getTitle()` Get process title

`getAbstract()` Get process abstract

`getVersion()` Get process version

`isStatusSupported()` Get whether status is supported

`isStoreSupported()` Get whether store is supported

`getDataInputs()` Get data inputs

`getProcessOutputs()` Get process outputs

`asDataFrame()` Get process description as data.frame

Note

Class used internally by **ows4R**

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSResponseDocument *WPSResponseDocument*

Description

WPSResponseDocument

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a WPS ResponseDocument

Methods

`new(xml, storeExecuteResponse, lineage, status, output)` This method is used to instantiate a WPSResponseDocument object

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

WPSStatus

WPSStatus

Description

WPSStatus

Format

[R6Class](#) object.

Value

Object of [R6Class](#) for modelling a WPS Status

Methods

`new(xml, serviceVersion)` This method is used to instantiate a WPSStatus object

`decode(xml)` Decodes WPS status from XML

`getValue()` Get status value, among accepted WPS status values defined in the WPS standard:
ProcessAccepted, ProcessStarted, ProcessPaused, ProcessSucceeded, ProcessFailed.

`getPercentCompleted()` Get percent completed

`getCreationTime()` Get creation time

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

Index

- * **Abstract**
 - OGCAbstractObject, 16
- * **And**
 - And, 3
- * **BBOX**
 - BBOX, 4
- * **BinaryComparisonOpType**
 - BinaryComparisonOpType, 5
 - PropertyIsEqualTo, 31
 - PropertyIsGreaterThan, 32
 - PropertyIsGreaterThanOrEqualTo, 32
 - PropertyIsLessThan, 33
 - PropertyIsLessThanOrEqualTo, 34
 - PropertyIsNotEqualTo, 35
- * **BinaryLogicOpType**
 - And, 3
 - BinaryLogicOpType, 6
 - Or, 19
- * **Bounding**
 - WPSBoundingBoxData, 46
- * **Box**
 - WPSBoundingBoxData, 46
- * **CSW**
 - CSWCapabilities, 6
 - CSWClient, 7
 - CSWDescribeRecord, 9
 - CSWGetRecordById, 10
 - CSWGetRecords, 11
 - CSWHarvest, 11
 - CSWRecordProperty, 13
 - CSWTransaction, 14
- * **Capabilities**
 - CSWCapabilities, 6
 - WPSCapabilities, 46
- * **CodeType**
 - OWSCodeType, 23
- * **Common**
 - OWSClient, 22
- * **ComplexData**
 - WPSComplexData, 48
- * **DescribeFeatureType**
 - WFSDescribeFeatureType, 39
- * **DescribeProcess**
 - WPSDescribeProcess, 50
- * **DescribeRecord**
 - CSWDescribeRecord, 9
- * **Exception**
 - WPSException, 52
- * **ExecuteResponse**
 - WPSExecuteResponse, 53
- * **Execute**
 - WPSExecute, 52
- * **Expression**
 - And, 3
 - BBOX, 4
 - BinaryComparisonOpType, 5
 - BinaryLogicOpType, 6
 - Not, 16
 - OGCExpression, 17
 - Or, 19
 - PropertyIsBetween, 30
 - PropertyIsEqualTo, 31
 - PropertyIsGreaterThan, 32
 - PropertyIsGreaterThanOrEqualTo, 32
 - PropertyIsLessThan, 33
 - PropertyIsLessThanOrEqualTo, 34
 - PropertyIsLike, 34
 - PropertyIsNotEqualTo, 35
 - PropertyIsNull, 36
 - UnaryLogicOpType, 37
- * **FeatureType**
 - WFSFeatureType, 40
 - WFSFeatureTypeElement, 41
- * **Feature**
 - WFSClient, 38
- * **Filter**
 - CSWConstraint, 9
 - OGCFilter, 18

- * **GetCapabilities**
 - OWSCapabilities, 21
 - OWSGetCapabilities, 23
 - WFSCapabilities, 38
 - WMSCapabilities, 42
- * **GetFeatureInfo**
 - WMSClient, 43
 - WMSGetFeatureInfo, 44
- * **GetFeature**
 - WFSGetFeature, 42
- * **GetRecordById**
 - CSWGetRecordById, 10
- * **GetRecords**
 - CSWGetRecords, 11
- * **HTTP**
 - OWSHttpRequest, 24
- * **Harvest**
 - CSWHarvest, 11
- * **Identification**
 - OWSServiceIdentification, 27
- * **Input**
 - WPSInput, 54
 - WPSOutput, 58
- * **Layer**
 - WMSLayer, 45
- * **LiteralData**
 - WPSLiteralData, 56
- * **Map**
 - WMSClient, 43
- * **Not**
 - Not, 16
- * **OGC**
 - And, 3
 - BBOX, 4
 - BinaryComparisonOpType, 5
 - BinaryLogicOpType, 6
 - CSWCapabilities, 6
 - CSWClient, 7
 - CSWConstraint, 9
 - CSWDescribeRecord, 9
 - CSWGetRecordById, 10
 - CSWGetRecords, 11
 - CSWHarvest, 11
 - CSWQuery, 12
 - CSWTransaction, 14
 - Not, 16
 - OGCAbstractObject, 16
 - OGCExpression, 17
 - OGCFilter, 18
 - Or, 19
 - OWSBoundingBox, 20
 - OWSCapabilities, 21
 - OWSClient, 22
 - OWSGetCapabilities, 23
 - OWSHttpRequest, 24
 - OWSOperation, 25
 - OWSOperationsMetadata, 26
 - OWSRequest, 27
 - OWSServiceIdentification, 27
 - OWSServiceProvider, 28
 - OWSUtils, 29
 - OWSWG84BoundingBox, 30
 - PropertyIsBetween, 30
 - PropertyIsEqualTo, 31
 - PropertyIsGreaterThan, 32
 - PropertyIsGreaterThanOrEqualTo, 32
 - PropertyIsLessThan, 33
 - PropertyIsLessThanOrEqualTo, 34
 - PropertyIsLike, 34
 - PropertyIsNotEqualTo, 35
 - PropertyIsNull, 36
 - UnaryLogicOpType, 37
 - WFSCapabilities, 38
 - WFSClient, 38
 - WFSDescribeFeatureType, 39
 - WFSFeatureType, 40
 - WFSFeatureTypeElement, 41
 - WFSGetFeature, 42
 - WMSCapabilities, 42
 - WMSClient, 43
 - WMSGetFeatureInfo, 44
 - WMSLayer, 45
 - WPSBoundingBoxData, 46
 - WPSCapabilities, 46
 - WPSClient, 47
 - WPSComplexData, 48
 - WPSComplexInputDescription, 49
 - WPSComplexOutputDescription, 50
 - WPSDescribeProcess, 50
 - WPSDescriptionParameter, 51
 - WPSException, 52
 - WPSExecute, 52
 - WPSExecuteResponse, 53
 - WPSFormat, 54
 - WPSInput, 54
 - WPSInputDescription, 55

- WPSLiteralData, [56](#)
- WPSLiteralInputDescription, [56](#)
- WPSLiteralOutputDescription, [57](#)
- WPSOutput, [58](#)
- WPSOutputDescription, [58](#)
- WPSParameter, [59](#)
- WPSProcess, [60](#)
- WPSProcessDescription, [60](#)
- WPSResponseDocument, [61](#)
- WPSStatus, [62](#)
- * **OWS**
 - OWSBoundingBox, [20](#)
 - OWSCapabilities, [21](#)
 - OWSClient, [22](#)
 - OWSCodeType, [23](#)
 - OWSHttpRequest, [24](#)
 - OWSNamespace, [25](#)
 - OWSOperation, [25](#)
 - OWSOperationsMetadata, [26](#)
 - OWSServiceIdentification, [27](#)
 - OWSServiceProvider, [28](#)
 - OWSUtils, [29](#)
 - OWSWG84BoundingBox, [30](#)
- * **Object**
 - OGCAbstractObject, [16](#)
- * **Or**
 - Or, [19](#)
- * **Parameter**
 - WPSParameter, [59](#)
- * **ProcessDescription**
 - WPSProcessDescription, [60](#)
- * **Processing**
 - WPSClient, [47](#)
- * **Process**
 - WPSClient, [47](#)
 - WPSComplexInputDescription, [49](#)
 - WPSComplexOutputDescription, [50](#)
 - WPSDescriptionParameter, [51](#)
 - WPSFormat, [54](#)
 - WPSInputDescription, [55](#)
 - WPSLiteralInputDescription, [56](#)
 - WPSLiteralOutputDescription, [57](#)
 - WPSOutputDescription, [58](#)
 - WPSProcess, [60](#)
 - WPSProcessDescription, [60](#)
- * **PropertyIsBetween**
 - PropertyIsBetween, [30](#)
- * **PropertyIsEqualTo**
 - PropertyIsEqualTo, [31](#)
- * **PropertyIsGreaterThanOrEqualTo**
 - PropertyIsGreaterThanOrEqualTo, [32](#)
- * **PropertyIsGreaterThan**
 - PropertyIsGreaterThan, [32](#)
- * **PropertyIsLessThan**
 - PropertyIsLessThan, [33](#)
- * **PropertyIsLesserThanOrEqualTo**
 - PropertyIsLesserThanOrEqualTo, [34](#)
- * **PropertyIsLike**
 - PropertyIsLike, [34](#)
- * **PropertyIsNotEqualTo**
 - PropertyIsNotEqualTo, [35](#)
- * **PropertyIsNull**
 - PropertyIsNull, [36](#)
- * **Provider**
 - OWSServiceProvider, [28](#)
- * **Query**
 - CSWQuery, [12](#)
- * **RecordProperty**
 - CSWRecordProperty, [13](#)
- * **Request**
 - OWSHttpRequest, [24](#)
 - OWSRequest, [27](#)
- * **ResponseDocument**
 - WPSResponseDocument, [61](#)
- * **Service**
 - OWSRequest, [27](#)
 - OWSServiceIdentification, [27](#)
 - OWSServiceProvider, [28](#)
- * **Status**
 - WPSStatus, [62](#)
- * **Transaction**
 - CSWTransaction, [14](#)
- * **UnaryLogicOpType**
 - Not, [16](#)
 - UnaryLogicOpType, [37](#)
- * **Utils**
 - OWSUtils, [29](#)
- * **WFS**
 - WFSCapabilities, [38](#)
 - WFSClient, [38](#)
 - WFSDescribeFeatureType, [39](#)
 - WFSFeatureType, [40](#)
 - WFSFeatureTypeElement, [41](#)
 - WFSGetFeature, [42](#)
- * **WMS**
 - WMSCapabilities, [42](#)

- WMSClient, [43](#)
 - WMSGetFeatureInfo, [44](#)
 - WMSLayer, [45](#)
 - * **WPS**
 - WPSBoundingBoxData, [46](#)
 - WPSCapabilities, [46](#)
 - WPSClient, [47](#)
 - WPSComplexData, [48](#)
 - WPSComplexInputDescription, [49](#)
 - WPSComplexOutputDescription, [50](#)
 - WPSDescribeProcess, [50](#)
 - WPSDescriptionParameter, [51](#)
 - WPSException, [52](#)
 - WPSExecute, [52](#)
 - WPSExecuteResponse, [53](#)
 - WPSFormat, [54](#)
 - WPSInput, [54](#)
 - WPSInputDescription, [55](#)
 - WPSLiteralData, [56](#)
 - WPSLiteralInputDescription, [56](#)
 - WPSLiteralOutputDescription, [57](#)
 - WPSOutput, [58](#)
 - WPSOutputDescription, [58](#)
 - WPSParameter, [59](#)
 - WPSProcess, [60](#)
 - WPSProcessDescription, [60](#)
 - WPSResponseDocument, [61](#)
 - WPSStatus, [62](#)
 - * **boundingbox**
 - OWSBoundingBox, [20](#)
 - OWSWG84BoundingBox, [30](#)
 - * **catalogue**
 - CSWClient, [7](#)
 - * **complex**
 - WPSComplexInputDescription, [49](#)
 - WPSComplexOutputDescription, [50](#)
 - * **data**
 - WPSBoundingBoxData, [46](#)
 - * **description**
 - WPSComplexInputDescription, [49](#)
 - WPSComplexOutputDescription, [50](#)
 - WPSDescriptionParameter, [51](#)
 - WPSFormat, [54](#)
 - WPSInputDescription, [55](#)
 - WPSLiteralInputDescription, [56](#)
 - WPSLiteralOutputDescription, [57](#)
 - WPSOutputDescription, [58](#)
 - * **input**
 - WPSComplexInputDescription, [49](#)
 - WPSDescriptionParameter, [51](#)
 - WPSFormat, [54](#)
 - WPSInputDescription, [55](#)
 - WPSLiteralInputDescription, [56](#)
 - * **literal**
 - WPSLiteralOutputDescription, [57](#)
 - * **metadata**
 - OWSOperationsMetadata, [26](#)
 - * **namespace**
 - OWSNamespace, [25](#)
 - * **operation**
 - OWSOperation, [25](#)
 - OWSOperationsMetadata, [26](#)
 - * **output**
 - WPSComplexOutputDescription, [50](#)
 - WPSLiteralOutputDescription, [57](#)
 - WPSOutputDescription, [58](#)
 - * **parameter**
 - WPSDescriptionParameter, [51](#)
 - * **service**
 - CSWClient, [7](#)
 - * **web**
 - CSWClient, [7](#)
 - * **wgs84**
 - OWSWG84BoundingBox, [30](#)
- And, [3](#)
 - BBOX, [4](#)
 - BinaryComparisonOpType, [5](#)
 - BinaryLogicOpType, [6](#)
 - CSWCapabilities, [6](#)
 - CSWClient, [7](#)
 - CSWConstraint, [9](#)
 - CSWDescribeRecord, [9](#)
 - CSWGetRecordById, [10](#)
 - CSWGetRecords, [11](#)
 - CSWHarvest, [11](#)
 - CSWQuery, [12](#)
 - CSWRecordProperty, [13](#)
 - CSWTransaction, [14](#)
 - getOWSNamespace, [15](#)
 - getOWSNamespace, [15](#)
 - Not, [16](#)
 - OGCAbstractObject, [16](#)

- OGCExpression, [17](#), [18](#)
- OGCFilter, [18](#)
- Or, [19](#)
- ows4R, [19](#)
- ows4R-package (ows4R), [19](#)
- OWSBoundingBox, [20](#)
- OWSCapabilities, [21](#)
- OWSClient, [22](#)
- OWSCodeType, [23](#)
- OWSGetCapabilities, [23](#)
- OWSHttpRequest, [24](#)
- OWSNamespace, [25](#)
- OWSOperation, [25](#)
- OWSOperationsMetadata, [26](#)
- OWSRequest, [27](#)
- OWSServiceIdentification, [27](#)
- OWSServiceProvider, [28](#)
- OWSUtils, [29](#)
- OWSWGSS84BoundingBox, [30](#)

- PropertyIsBetween, [30](#)
- PropertyIsEqualTo, [31](#)
- PropertyIsGreaterThan, [32](#)
- PropertyIsGreaterThanOrEqualTo, [32](#)
- PropertyIsLessThan, [33](#)
- PropertyIsLessThanOrEqualTo, [34](#)
- PropertyIsLike, [34](#)
- PropertyIsNotEqualTo, [35](#)
- PropertyIsNull, [36](#)

- R6Class, [4–7](#), [9–14](#), [16–62](#)
- registerOWSNamespace, [36](#)

- setOWSNamespace, [37](#)

- UnaryLogicOpType, [37](#)

- WFSCapabilities, [38](#)
- WFSClient, [38](#)
- WFSDescribeFeatureType, [39](#)
- WFSFeatureType, [40](#)
- WFSFeatureTypeElement, [41](#)
- WFSGetFeature, [42](#)
- WMSCapabilities, [42](#)
- WMSClient, [43](#)
- WMSGetFeatureInfo, [44](#)
- WMSLayer, [45](#)
- WPSBoundingBoxData, [46](#)
- WPSCapabilities, [46](#)
- WPSClient, [47](#)
- WPSComplexData, [48](#)
- WPSComplexInputDescription, [49](#)
- WPSComplexOutputDescription, [50](#)
- WPSDescribeProcess, [50](#)
- WPSDescriptionParameter, [51](#)
- WPSException, [52](#)
- WPSExecute, [52](#)
- WPSExecuteResponse, [53](#)
- WPSFormat, [54](#)
- WPSInput, [54](#)
- WPSInputDescription, [55](#)
- WPSLiteralData, [56](#)
- WPSLiteralInputDescription, [56](#)
- WPSLiteralOutputDescription, [57](#)
- WPSOutput, [58](#)
- WPSOutputDescription, [58](#)
- WPSParameter, [59](#)
- WPSProcess, [60](#)
- WPSProcessDescription, [60](#)
- WPSResponseDocument, [61](#)
- WPSStatus, [62](#)