Package: rneos (via r-universe)

August 26, 2024

Version 0.4-0

Date 2020-04-22

Title XML-RPC Interface to NEOS

Depends R (>= 2.10.0), methods, RCurl, XML, stats

LazyLoad yes

Description Within this package the XML-RPC API to NEOS <https://neos-server.org/neos/> is implemented. This enables the user to pass optimization problems to NEOS and retrieve results within R.

License GPL (>= 2)

Maintainer Bernhard Pfaff <bernhard@pfaffikus.de>

Repository/R-Forge/Project rneos

Repository/R-Forge/Revision 267

Repository/R-Forge/DateTimeStamp 2020-04-22 12:56:33

Date/Publication 2020-04-23 07:40:02 UTC

NeedsCompilation no

Author Bernhard Pfaff [aut, cre], Duncan Temple Lang [ctb] (included XMLRPC package, hosted at http://www.omegahat.net/XMLRPC/; BSD license.)

Repository https://bpfaff.r-universe.dev

RemoteUrl https://github.com/cran/rneos

RemoteRef HEAD

RemoteSha 7d6a1daf182f46f8efb3556b0b5c4f813b1b34ca

Contents

`CreateNeosComm` *Creating an object for communications with NEOS*

Description

This function creates an object of class `NeosComm` that will contain all necessary information for dealing with HTTP requests to NEOS. This object will be needed in all requests to NEOS and hence must be created in advance of XML-RPC requests.

Usage

```
CreateNeosComm(curlopts = list(httpheader = c(`Content-Type` =
"text/xml", `User-Agent` = "R"), port = 3333), curlhandle =
getCurlHandle())
```

Arguments

`curlopts`	A named list of elements that are passed as options to curl. By default, the `httpheader` and the `port` options are preset.
`curlhandle`	An object of class `CURLHandle`. By default the returned object of getCurlHandle() is employed.

Details

A list of valid `curl` options can be retrieved from `listCurlOptions()`. Please note, that the relevant HTTP-bodies within the requests will be created directly in the API-functions provided in this package and must not be provided as list elements in `curloptions`. However, if one accesses NEOS *via* a Proxy-Server, for instance, than the values for the relevant options must be set within the list-argument `curlopts`. The values of the returned object will be passed down to the function `xml.rpc()` which is utilised for all calls to the function `Nfoo` contained in this package. Hereby, `foo` signify the name of NEOS-API.

Value

An object of class `NeosComm`.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

`NeosComm`

Examples

```
## Not run:
nc <- CreateNeosComm()
nc

## End(Not run)
```

`CreateXmlString` *Inserting CDATA into XML-templates of NEOS*

Description

With this function the information for XML-templates can be inserted. Ordinarily, one creates an object with the function `NgetSolverTemplate()` first and then inserts the requested CDATA fields of this XML-form with this function.

Usage

```
CreateXmlString(neosxml, cdatalist)
```

Arguments

neosxml	An object of class NeosXml created with the function NgetSolverTemplate().
cdatalist	A named list object with the CDATA tags to be filled.

Value

A character string containing the specified optimization problem, which can then be used in a call to NsubmitJob().

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosXml and NgetSolverTemplate

Examples

```
## Not run:

tmp <-NgetSolverTemplate(category = "go", solvername = "ASA",
inputMethod = "AMPL")
## setting path to example model and data file
modf <- system.file("ExAMPL", "diet.mod", package = "rneos")
datf <- system.file("ExAMPL", "diet.dat", package = "rneos")
## import of file contents
modc <- paste(paste(readLines(modf), collapse = "\n"), "\n")
cat(modc)
datc <- paste(paste(readLines(datf), collapse = "\n"), "\n")
cat(datc)
## create list object
argslist <- list(model = modc, data = datc, commands = "",
comments = "")
## create XML string
xmls <- CreateXmlString(neosxml = tmp, cdatalist = argslist)
xmls

## End(Not run)
```

NemailHelp *XML-RPC method emailHelp of Neos*

Description

This functions calls the XML-RPC method "emailHelp()" of NEOS and returns general help message for email users.

Usage

```
NemailHelp(convert = TRUE, nc = CreateNeosComm())
```

Arguments

convert	Logical, if convert = TRUE (the default) the value of the returned XML-RPC result is extracted and returned as character, otherwise the XML-RPC string is returned.
nc	Object of class NeosComm: By default, this argument is set by calling CreateNeosComm() and thereby using the default values of this function.

Value

An object of class NeosAns.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosAns and CreateNeosComm

Examples

```
## Not run:

NemailHelp()

## End(Not run)
```

NeosAns-class *Class "NeosAns"*

Description

Objects of this class contain the returned results from NEOS as well as information on which kind of query has been sent and how it was sent.

Objects from the Class

Objects can be created by calls of the form `new("NeosAns", ...)` or more conveniently by calling the relevant R API functions.

Slots

`ans`: Object of class `"character"`: The returned XML-RPC of NEOS as character string.

`method`: Object of class `"character"`: The name of the called API function.

`call`: Object of class `"call"`: The call to the generating function of the object.

`nc`: Object of class `"NeosComm"`: The NeosComm object that has been used in the request to NEOS.

Methods

show `signature(object = "NeosAns")`: Returns the converted slot `ans` from an object of class `NeosAns`.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosComm

Examples

```
showClass("NeosAns")
```

NeosComm-class *Class "NeosComm"*

Description

The purpose of this class is to gather the relevant information needed for HTTP requests that is passed to NEOS.

Objects from the Class

Objects can be created by calls of the form `new("NeosComm", ...)` or more conveniently by creating an object from `CreateNeosComm()`.

Slots

`url`: Object of class `"character"`: The URL to NEOS, *i.e.*, `http://www.neos-server.org`

`curlopts`: Object of class `"list"`: A named list of valid Curl options.

`curlhandle`: Object of class `"CURLHandle"`: Objects of this class can be created and altered with `getCurlHandle()`

Methods

No methods defined with class "NeosComm" in the signature.

Author(s)

Bernhard Pfaff

References

Omegahat web site for RCurl: `http://www.omegahat.net/RCurl`, and
libcurl web site: `http://curl.haxx.se`

See Also

`CreateNeosComm`, `getCurlHandle` and `CURLHandle-class`

Examples

```
showClass("NeosComm")
## Not run:

nc <- CreateNeosComm()
nc

## End(Not run)
```

NeosJob-class *Class "NeosJob"*

Description

Objects of this class contain among other slots the returned jobnumber and password for jobs submitted to NEOS. Objects of this class can then be used for retrieving the optimization results.

Objects from the Class

Objects can be created by calls of the form `new("NeosJob", ...)` or more conveniently by calling the relevant R API function `NsubmitJob`.

Slots

`jobnumber`: Object of class `"numeric"`: The returned job number.

`password`: Object of class `"character"`: The returned pass word.

`method`: Object of class `"character"`: The name of the called API function.

`call`: Object of class `"call"`: The call to the generating function of the object.

`nc`: Object of class `"NeosComm"`: The NeosComm object that has been used in the request to NEOS.

Methods

show `signature(object = "NeosJob")`: Displays the slots `jobnumber` and `password` from an object of class `NeosJob`.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

Examples

```
showClass("NeosJob")
## Not run:
tmp <-NgetSolverTemplate(category = "go", solvername = "ASA",
inputMethod = "AMPL")
## setting path to example model and data file
modf <- system.file("ExAMPL", "diet.mod", package = "rneos")
datf <- system.file("ExAMPL", "diet.dat", package = "rneos")
## import of file contents
modc <- paste(paste(readLines(modf), collapse = "\n"), "\n")
datc <- paste(paste(readLines(datf), collapse = "\n"), "\n")
## create list object
argslist <- list(model = modc, data = datc, commands = "",
```

```
comments = "")
## create XML string
xmls <- CreateXmlString(neosxml = tmp, cdatalist = argslist)
NsubmitJob(xmlstring = xmls, user = "rneos", interface = "", id = 0)

## End(Not run)
```

NeosOff-class *Class "NeosOff"*

Description

Objects of this class contain among other slots the partial result and the offset.

Objects from the Class

Objects can be created by calls of the form `new("NeosOff", ...)`.

Slots

`ans`: Object of class `"character"`: The partial result returned from NEOS.

`offset`: Object of class `"integer"`: The integer offset until the results have been returned.

`jobnumber`: Object of class `"numeric"`: The returned job number.

`password`: Object of class `"character"`: The returned pass word.

`method`: Object of class `"character"`: The name of the called API function.

`call`: Object of class `"call"`: The call to the generating function of the object.

`nc`: Object of class `"NeosComm"`: The NeosComm object that has been used in the request to NEOS.

Methods

update Updates an object of class `NeosOff`.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

Examples

```
showClass("NeosOff")
```

NeosXml-class *Class "NeosXml"*

Description

Objects of this class contain the returned results from NEOS as well as information on which kind of query has been sent and how it was sent.

Objects from the Class

Objects can be created by calls of the form `new("NeosXml", ...)` or more conveniently by calling the relevant R API functions.

Slots

`xml`: Object of class `"XMLNode"`: The returned and converted XML-template of NEOS.

`method`: Object of class `"character"`: The name of the called API function.

`call`: Object of class `"call"`: The call to the generating function of the object.

`nc`: Object of class `"NeosComm"`: The NeosComm object that has been used in the request to NEOS.

Methods

show `signature(object = "NeosXml")`: Returns the converted slot `xml` from an object of class `NeosXml`.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosComm

Examples

```
showClass("NeosXml")
## Not run:

tmp <-NgetSolverTemplate(category = "go", solvername = "ASA",
inputMethod = "AMPL")
## setting path to example model and data file
modf <- system.file("ExAMPL", "diet.mod", package = "rneos")
datf <- system.file("ExAMPL", "diet.dat", package = "rneos")
## import of file contents
```

```
modc <- paste(paste(readLines(modf), collapse = "\n"), "\n")
datc <- paste(paste(readLines(datf), collapse = "\n"), "\n")
## create list object
argslist <- list(model = modc, data = datc, commands = "",
comments = "")
## create XML string
xmls <- CreateXmlString(neosxml = tmp, cdatalist = argslist)
test <- NsubmitJob(xmlstring = xmls, user = "rneos", interface = "",
id = 0)
NgetJobStatus(obj = test, convert = TRUE)
NgetJobInfo(obj = test, convert = TRUE)
NgetFinalResults(obj = test, convert = TRUE)

## End(Not run)
```

`NgetFinalResults` *XML-RPC method getFinalResults of Neos*

Description

This functions calls the XML-RPC method "getFinalResults()" of NEOS, which gets results of a job submitted to NEOS. If the job is still running, then this function will hang until the job is finished.

Usage

```
NgetFinalResults(obj, convert = TRUE)
```

Arguments

`obj`	NeosJob, an object of class `NeosJob` as returned by the function `NsubmitJob`.
`convert`	Logical, if `convert = TRUE` (the default) the value of the returned XML-RPC result is extracted and returned as character, otherwise the XML-RPC string is returned (base-64 encoded).

Value

An object of class `NeosAns`.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

`NeosAns` and `NsubmitJob`

Examples

```
## Not run:
tmp <-NgetSolverTemplate(category = "lp", solvername = "MOSEK",
                         inputMethod = "AMPL")
## setting path to example model and data file
modf <- system.file("ExAMPL", "diet.mod", package = "rneos")
datf <- system.file("ExAMPL", "diet.dat", package = "rneos")
## import of file contents
modc <- paste(paste(readLines(modf), collapse = "\n"), "\n")
datc <- paste(paste(readLines(datf), collapse = "\n"), "\n")
## create list object
argslist <- list(model = modc, data = datc, commands = "",
comments = "")
## create XML string
xmls <- CreateXmlString(neosxml = tmp, cdatalist = argslist)
test <- NsubmitJob(xmlstring = xmls, user = "rneos", interface = "",
id = 0)
NgetFinalResults(obj = test, convert = TRUE)

## End(Not run)
```

NgetFinalResultsNonBlocking

XML-RPC method getFinalResultsNonBlocking of Neos

Description

This functions calls the XML-RPC method "getFinalResultsNonBlocking()" of NEOS, which gets results of a job submitted to NEOS. If the job is still running, then this function will return an empty string (base-64 encoded).

Usage

```
NgetFinalResultsNonBlocking(obj, convert = TRUE)
```

Arguments

`obj`	NeosJob, an object of class `NeosJob` as returned by the function `NsubmitJob`.
`convert`	Logical, if `convert = TRUE` (the default) the value of the returned XML-RPC result is extracted and returned as character, otherwise the XML-RPC string is returned (base-64 encoded).

Value

An object of class `NeosAns`.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosAns and NsubmitJob

Examples

```
## Not run:

tmp <-NgetSolverTemplate(category = "go", solvername = "ASA",
inputMethod = "AMPL")
## setting path to example model and data file
modf <- system.file("ExAMPL", "diet.mod", package = "rneos")
datf <- system.file("ExAMPL", "diet.dat", package = "rneos")
## import of file contents
modc <- paste(paste(readLines(modf), collapse = "\n"), "\n")
datc <- paste(paste(readLines(datf), collapse = "\n"), "\n")
## create list object
argslist <- list(model = modc, data = datc, commands = "",
comments = "")
## create XML string
xmls <- CreateXmlString(neosxml = tmp, cdatalist = argslist)
test <- NsubmitJob(xmlstring = xmls, user = "rneos", interface = "",
id = 0)
NgetFinalResultsNonBlocking(obj = test, convert = TRUE)

## End(Not run)
```

NgetIntermediateResults

XML-RPC method getIntermediateResults of Neos

Description

This functions calls the XML-RPC method "getIntermediateResults()" of NEOS, which returns intermediate results of a job submitted to NEOS, starting at the character offset up to the last received data. Intermediate results are usually the standard output of the solver daemon. If the job is still running, then this function will hang until another packet of output is sent to NEOS or the job is finished. This function will return a tuple of thebase-64 encoded object and the new offset (object, newoffset). The offset refers to uncoded characters.

Usage

```
NgetIntermediateResults(obj, offset = NULL, convert = TRUE)
```

Arguments

obj	NeosJob, an object of class NeosJob as returned by the function NsubmitJob.
offset	Integer, the offset from which on the results are returned. In a first run this is set to integer(0).
convert	Logical, if convert = TRUE (the default) the value of the returned XML-RPC result is extracted and returned as character, otherwise the XML-RPC string is returned (base-64 encoded).

Value

An object of class NeosOff.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosOff and NsubmitJob

Examples

```
## Not run:

tmp <-NgetSolverTemplate(category = "go", solvername = "ASA",
inputMethod = "AMPL")
## setting path to example model and data file
modf <- system.file("ExAMPL", "diet.mod", package = "rneos")
datf <- system.file("ExAMPL", "diet.dat", package = "rneos")
## import of file contents
modc <- paste(paste(readLines(modf), collapse = "\n"), "\n")
datc <- paste(paste(readLines(datf), collapse = "\n"), "\n")
## create list object
argslist <- list(model = modc, data = datc, commands = "",
comments = "")
## create XML string
xmls <- CreateXmlString(neosxml = tmp, cdatalist = argslist)
test <- NsubmitJob(xmlstring = xmls, user = "rneos", interface = "",
id = 0)
NgetIntermediateResults(obj = test, convert = TRUE)

## End(Not run)
```

NgetIntermediateResultsNonBlocking

XML-RPC method getIntermediateResultsNonBlocking of Neos

Description

This functions calls the XML-RPC method "getIntermediateResultsNonBlocking()" of NEOS, which returns intermediate results of a job submitted to NEOS, starting at the character offset up to the last received data. Intermediate results are usually the standard output of the solver daemon. If the job is still running, then this function will hang until another packet of output is sent to NEOS or the job is finished. This function will return a tuple of thebase-64 encoded object and the new offset (object, newoffset). The offset refers to uncoded characters.

Usage

```
NgetIntermediateResultsNonBlocking(obj, offset = NULL,
                                   convert = TRUE)
```

Arguments

`obj`	NeosJob, an object of class `NeosJob` as returned by the function `NsubmitJob`.
`offset`	Integer, the offset from which on the results are returned. In a first run this is set to `integer(0)`.
`convert`	Logical, if `convert = TRUE` (the default) the value of the returned XML-RPC result is extracted and returned as character, otherwise the XML-RPC string is returned (base-64 encoded).

Value

An object of class `NeosOff`.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

`NeosOff` and `NsubmitJob`

Examples

```
## Not run:

tmp <-NgetSolverTemplate(category = "go", solvername = "ASA",
inputMethod = "AMPL")
## setting path to example model and data file
modf <- system.file("ExAMPL", "diet.mod", package = "rneos")
datf <- system.file("ExAMPL", "diet.dat", package = "rneos")
## import of file contents
modc <- paste(paste(readLines(modf), collapse = "\n"), "\n")
datc <- paste(paste(readLines(datf), collapse = "\n"), "\n")
## create list object
argslist <- list(model = modc, data = datc, commands = "",
comments = "")
## create XML string
xmls <- CreateXmlString(neosxml = tmp, cdatalist = argslist)
test <- NsubmitJob(xmlstring = xmls, user = "rneos", interface = "",
id = 0)
NgetIntermediateResultsNonBlocking(obj = test, convert = TRUE)

## End(Not run)
```

NgetJobInfo *XML-RPC method getJobInfo of Neos*

Description

This functions calls the XML-RPC method "getJobInfo()" of NEOS and returns a four-tuple (category, solver name, input, status).

Usage

```
NgetJobInfo(obj, convert = TRUE)
```

Arguments

obj	Object of class NeosJob, as returned by the function NsubmitJob.
convert	Logical, if convert = TRUE (the default) the value of the returned XML-RPC result is extracted and returned as character, otherwise the XML-RPC string is returned.

Value

An object of class NeosAns.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosAns, NeosJob and NgetJobStatus

Examples

```
## Not run:

tmp <-NgetSolverTemplate(category = "go", solvername = "ASA",
inputMethod = "AMPL")
## setting path to example model and data file
modf <- system.file("ExAMPL", "diet.mod", package = "rneos")
datf <- system.file("ExAMPL", "diet.dat", package = "rneos")
## import of file contents
modc <- paste(paste(readLines(modf), collapse = "\n"), "\n")
datc <- paste(paste(readLines(datf), collapse = "\n"), "\n")
## create list object
argslist <- list(model = modc, data = datc, commands = "",
comments = "")
## create XML string
xmls <- CreateXmlString(neosxml = tmp, cdatalist = argslist)
test <- NsubmitJob(xmlstring = xmls, user = "rneos", interface = "",
id = 0)
NgetJobInfo(obj = test, convert = TRUE)

## End(Not run)
```

NgetJobStatus *XML-RPC method getJobStatus of Neos*

Description

This functions calls the XML-RPC method "getJobStatus()" of NEOS and returns an object of class NeosAns. The functions returns the current job status (either "Done", "Running", "Waiting", "Unknown Job" or "Bad Password").

Usage

```
NgetJobStatus(obj, convert = TRUE)
```

Arguments

obj	NeosJob, an object of class NeosJob as returned by the function NsubmitJob.
convert	Logical, if convert = TRUE (the default) the value of the returned XML-RPC result is extracted and returned as character, otherwise the XML-RPC string is returned.

Value

An object of class NeosAns.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosAns and NsubmitJob

Examples

```
## Not run:
tmp <-NgetSolverTemplate(category = "go", solvername = "ASA",
inputMethod = "AMPL")
## setting path to example model and data file
modf <- system.file("ExAMPL", "diet.mod", package = "rneos")
datf <- system.file("ExAMPL", "diet.dat", package = "rneos")
## import of file contents
modc <- paste(paste(readLines(modf), collapse = "\n"), "\n")
datc <- paste(paste(readLines(datf), collapse = "\n"), "\n")
## create list object
argslist <- list(model = modc, data = datc, commands = "",
comments = "")
## create XML string
xmls <- CreateXmlString(neosxml = tmp, cdatalist = argslist)
test <- NsubmitJob(xmlstring = xmls, user = "rneos", interface = "",
id = 0)
NgetJobStatus(obj = test, convert = TRUE)

## End(Not run)
```

NgetSolverTemplate *XML-RPC method getSolverTemplate of Neos*

Description

This functions calls the XML-RPC method "getSolverTemplate()" of NEOS. If the solver category:solvername:inputMethod exists on NEOS, then an XML template is returned as an object of class NeosXml.

Usage

```
NgetSolverTemplate(category, solvername, inputMethod,
                   nc = CreateNeosComm())
```

Arguments

category	Character, the abbreviation of a category
solvername	Character, the name of the solver to be used.
inputMethod	Character, the name of the solver for which the xml-template shall be returned.
nc	Object of class NeosComm: By default, this argument is set by calling CreateNeosComm() and thereby using the default values of this function.

Value

An object of class NeosXml.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosXml, NlistAllSolvers and CreateNeosComm

Examples

```
## Not run:
NgetSolverTemplate(category = "go", solvername = "ASA", inputMethod = "AMPL")

## End(Not run)
```

Nhelp *XML-RPC method help of Neos*

Description

This functions calls the XML-RPC method "help()" of NEOS and returns an object of class NeosAns.

Usage

```
Nhelp(convert = TRUE, nc = CreateNeosComm())
```

Arguments

convert	Logical, if convert = TRUE (the default) the value of the returned XML-RPC result is extracted and returned as character, otherwise the XML-RPC string is returned.
nc	Object of class NeosComm: By default, this argument is set by calling CreateNeosComm() and thereby using the default values of this function.

Value

An object of class NeosAns.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosAns and CreateNeosComm

Examples

```
## Not run:

Nhelp()

## End(Not run)
```

NkillJob *XML-RPC method killJob of Neos*

Description

This method is used to cancel a job submission running on NEOS (or waiting to run on NEOS). The job password is required to prevent abuse of this function (extracted from relevant slot of obj). This functions calls the XML-RPC method "killJob()" of NEOS and returns an object of class NeosAns.

Usage

```
NkillJob(obj, killmsg = NULL, convert = TRUE)
```

Arguments

obj	Object of class NeosJob, as returned by the function NsubmitJob.
killmsg	Character, optional description.
convert	Logical, if convert = TRUE (the default) the value of the returned XML-RPC result is extracted and returned as character, otherwise the XML-RPC string is returned.

Value

An object of class NeosAns containing the returned message from NEOS.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosAns and NeosJob

NlistAllSolvers *XML-RPC method listAllSolvers of Neos*

Description

This functions calls the XML-RPC method "listAllSolvers()" of NEOS and returns an object of class NeosAns; a list of category:solver:inputMethod.

Usage

```
NlistAllSolvers(convert = TRUE, nc = CreateNeosComm())
```

Arguments

convert	Logical, if convert = TRUE (the default) the value of the returned XML-RPC result is extracted and returned as character, otherwise the XML-RPC string is returned.
nc	Object of class NeosComm: By default, this argument is set by calling CreateNeosComm() and thereby using the default values of this function.

Value

An object of class NeosAns.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosAns and CreateNeosComm

Examples

```
## Not run:

NlistAllSolvers()

## End(Not run)
```

`NlistCategories` *XML-RPC method listCategories of Neos*

Description

This functions calls the XML-RPC method "listCategories()" of NEOS and returns an object of class `NeosAns`; a dictionary with entries ('abbreviated name':'full name',...).

Usage

```
NlistCategories(convert = TRUE, nc = CreateNeosComm())
```

Arguments

`convert`	Logical, if `convert = TRUE` (the default) the value of the returned XML-RPC result is extracted and returned as character, otherwise the XML-RPC string is returned.
`nc`	Object of class `NeosComm`: By default, this argument is set by calling `CreateNeosComm()` and thereby using the default values of this function.

Value

An object of class `NeosAns`.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

`NeosAns` and `CreateNeosComm`

Examples

```
## Not run:

NlistCategories()

## End(Not run)
```

NlistSolversInCategory

XML-RPC method listSolversInCategory of Neos

Description

This functions calls the XML-RPC method "listSolversInCategory()" of NEOS and returns an object of class NeosAns. The returned content is a list of solver:input for every solver in the category (category can be abbreviation or full_name).

Usage

```
NlistSolversInCategory(category, convert = TRUE,
                       nc = CreateNeosComm())
```

Arguments

category	Character, the abbreviation of a category
convert	Logical, if convert = TRUE (the default) the value of the returned XML-RPC result is extracted and returned as character, otherwise the XML-RPC string is returned.
nc	Object of class NeosComm: By default, this argument is set by calling CreateNeosComm() and thereby using the default values of this function.

Value

An object of class NeosAns.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosAns, NlistCategories and CreateNeosComm

Examples

```
## Not run:
NlistSolversInCategory(category = "go")

## End(Not run)
```

Nping *XML-RPC method ping of Neos*

Description

This functions calls the XML-RPC method "ping()" of NEOS and returns an object of class NeosAns. It is verified that this NeosServer is running and a message 'NeosServer is alive' is returned.

Usage

```
Nping(convert = TRUE, nc = CreateNeosComm())
```

Arguments

convert	Logical, if convert = TRUE (the default) the value of the returned XML-RPC result is extracted and returned as character, otherwise the XML-RPC string is returned.
nc	Object of class NeosComm: By default, this argument is set by calling CreateNeosComm() and thereby using the default values of this function.

Value

An object of class NeosAns.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosAns, NlistCategories and CreateNeosComm

Examples

```
## Not run:
Nping()

## End(Not run)
```

NprintQueue *XML-RPC method printQueue of Neos*

Description

This functions calls the XML-RPC method "printQueue()" of NEOS and returns an object of class NeosAns, which is a string containing the current NEOS jobs.

Usage

```
NprintQueue(convert = TRUE, nc = CreateNeosComm())
```

Arguments

convert	Logical, if convert = TRUE (the default) the value of the returned XML-RPC result is extracted and returned as character, otherwise the XML-RPC string is returned.
nc	Object of class NeosComm: By default, this argument is set by calling CreateNeosComm() and thereby using the default values of this function.

Value

An object of class NeosAns.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosAns and CreateNeosComm

Examples

```
## Not run:

NprintQueue()

## End(Not run)
```

NsubmitJob *XML-RPC method submitJob of Neos*

Description

This functions calls the XML-RPC method "submitJob()" of NEOS and returns an object of class NeosJob.

Usage

```
NsubmitJob(xmlstring, user = "rneos", interface = "",
           id = 0, nc = CreateNeosComm())
```

Arguments

xmlstring	Character, the xml string according to the solver's template and filled withe the user's optimisation data.
user	Character, the name of the user; for certain optimizers an email address is required.
interface	Character, the name of the interface.
id	Integer, an identifier for the submitted job.
nc	Object of class NeosComm: By default, this argument is set by calling CreateNeosComm() and thereby using the default values of this function.

Value

An object of class NeosJob.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosJob, CreateXmlString and NgetSolverTemplate

Nversion *XML-RPC method version of Neos*

Description

This functions calls the XML-RPC method "version()" of NEOS and returns an object of class NeosAns, which is a string containing the version number of the NEOS server.

Usage

```
Nversion(convert = TRUE, nc = CreateNeosComm())
```

Arguments

convert	Logical, if convert = TRUE (the default) the value of the returned XML-RPC result is extracted and returned as character, otherwise the XML-RPC string is returned.
nc	Object of class NeosComm: By default, this argument is set by calling CreateNeosComm() and thereby using the default values of this function.

Value

An object of class NeosAns.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosAns and CreateNeosComm

Examples

```
## Not run:

Nversion()

## End(Not run)
```

Nwelcome *XML-RPC method welcome of Neos*

Description

This functions calls the XML-RPC method "welcome()" of NEOS and returns an object of class NeosAns, which is a welcome message (string).

Usage

```
Nwelcome(convert = TRUE, nc = CreateNeosComm())
```

Arguments

convert	Logical, if convert = TRUE (the default) the value of the returned XML-RPC result is extracted and returned as character, otherwise the XML-RPC string is returned.
nc	Object of class NeosComm: By default, this argument is set by calling CreateNeosComm() and thereby using the default values of this function.

Value

An object of class NeosAns.

Author(s)

Bernhard Pfaff

References

NEOS API: https://neos-server.org/neos/xml-rpc.html

See Also

NeosAns and CreateNeosComm

Examples

```
## Not run:
Nwelcome()

## End(Not run)
```

rpc.serialize *Serialize R objects to XML-RPC format*

Description

This function and its methods convert R objects to XML for use in XML-RPC requests.

Usage

```
rpc.serialize(x, ...)
```

Arguments

x	the R object to be serialized in XML-RPC format
...	additional parameters understood by methods

Value

An XMLInternalNode object representing the XML content.

Author(s)

Duncan Temple Lang

References

The XML-RPC specification at http://www.xmlrpc.com/spec.md.

See Also

xml.rpc

xml.rpc *Invoke XML-RPC method from R*

Description

This function can be used to invoke a method provided by an XML-RPC (remote procedure call) server. It can pass R objects in the request by serializing them to XML format and also converts the result back to R.

Usage

```
xml.rpc(url, method, ..., .args = list(...), .opts = list(),
        .defaultOpts = list(httpheader = c("Content-Type" = "text/xml"),
                            followlocation = TRUE,
                            useragent = useragent),
         .convert = TRUE, .curl = getCurlHandle(),
          useragent = "R-XMLRPC")
```

Arguments

`url`	the URL of the XML-RPC server
`method`	a string giving the name of the XML-RPC method to invoke
`...`	a collection of argument valuesn
`.args`	an alternative way to specify the collection (list) of arguments
`.opts`	a list of options passed on to `postForm`. This is for the caller to specify server-specific curl options as opposed to general XML-RPC options which are set via `.defaultOpts`.
`.defaultOpts`	standard/default RCurl options used when making this call
`.convert`	either a logical value indicating whether to perform the defalt conversion (via `convertToR`) or not, or alternatively a function which is called with a string giving the body of the HTTP response of the XML-RPC call.
`.curl`	a CURLHandle object that the caller can specify to allow reusing existing handles and connections. This can greatly improve efficiency.
`useragent`	the string identifying the application that is reported to the Web server as making the request.

Value

If `.convert` is a logical value and `TRUE`, an R object giving the result of the XML-RPC method invocation. If `.convert` is `FALSE`, a string giving the body of the response.

If `.convert` is a function, it is called with the body of the XML-RPC response as a string.

Author(s)

Duncan Temple Lang

References

http://www.xmlrpc.com http://www.cafeconleche.org/books/xmljava/chapters/ch02s05.html for a DTD for XML-RPC and examples and discussion.

See Also

`postForm` `getURL` and REST Web services `SSOAP` package.

XMLRPCServer *Create an instance of an XMLRPCServer object*

Description

The XMLRPCServer class is a means to identify a string as the URL of an XML-RPC server. We can then use this to invoke a method provided by the server either via a call to xml.rpc or via an expression of the form server$methodName(arg1, arg2, ...).

The XMLRPCServerConnection class allows us to associate a CURLHandle object with an XML-RPC server. This connection is then used in each of the calls to that server. This allows us to reuse a single curl connection to the server and also slightly simplifies passing it to each call.

Usage

```
XMLRPCServer(url, curl = NULL,
class = if (!is.null(curl)) "XMLRPCServerConnection" else
  "XMLRPCServer", ..., .opts = list(...))
```

Arguments

url	the URL for the XML-RPC server.
curl	either a logical value indicating whether to create a new CURLHandle object, or an instance of a CURLHandle or alternatively NULL. If CURL options are specified via the ... or .opts parameters, then a CURL handle is automatically created using these.
class	the name of the class to create.
...	name=value pairs of CURL options that are used to create a new CURLHandle object.
.opts	an alternative way to specify the CURL options for the handle to be created.

Value

An object of class given by the value of class.

Author(s)

Duncan Temple Lang

See Also

xml.rpc

Index