Package 'rrdf'

February 15, 2013

Type Package

Title rrdf - support for the Resource Description Framework

Version 1.9.2

Date 2012-11-30

Author Egon Willighagen <egon.willighagen@gmail.com>

Maintainer Egon Willighagen <egon.willighagen@gmail.com>

Depends R (>= 2.0.0), rJava, rrdflibs (>= 1.2.2)

Description Work with RDF data.

License AGPL-3

LazyLoad yes

Repository CRAN

Date/Publication 2012-11-30 07:43:38

NeedsCompilation no

R topics documented:

`rrdf-package` *Functionality to work with RDF data.*

Description

Methods to read and write RDF, and query RDF using SPARQL.

Details

Package based on the Jena RDF libraries to bring RDF and SPARQL functionality into the R environment.

Author(s)

Egon Willighagen

Maintainer: Egon Willighagen <egon.willighagen@gmail.com>

`add.data.triple` *Add an data property to the model.*

Description

Adds an add property to the model.

Usage

```
add.data.triple(store,
  subject="http://example.org/Subject",
  predicate="http://example.org/Predicate",
  data="Value",
  type=NULL)
```

Arguments

`store`	A triple store, for example create with new.rdf().
`subject`	URI of the subject.
`predicate`	URI of the predicate.
`data`	A data value.
`type`	Optional parameter for the data value type. Can be "string", "float", "double", or any other XML Schema Data Type.

Value

The update Java Model object containing the existing and new triple.

Author(s)

Egon Willighagen

Examples

```
store = new.rdf()
add.data.triple(store,
  subject="http://example.org/Subject",
  predicate="http://example.org/Predicate",
  data="Value")
add.data.triple(store,
  subject="http://example.org/Subject",
  predicate="http://example.org/Predicate",
  data="1", type="integer")
```

---

add.prefix *Define an prefix for a namespace.*

Description

Adds an prefix for a namespace to the model.

Usage

```
add.prefix(store, prefix, namespace)
```

Arguments

store	A triple store, for example create with new.rdf().
prefix	String to be used as prefix.
namespace	URI of the namespace.

Author(s)

Egon Willighagen

Examples

```
store = new.rdf()
add.prefix(store,
  prefix="dc",
  namespace="http://purl.org/dc/terms/"
)
```

`add.triple` *Add an object property to the model.*

Description

Adds an object property to the model.

Usage

```
add.triple(store,
  subject="http://example.org/Subject",
  predicate="http://example.org/Predicate",
  object="http://example.org/Object")
```

Arguments

`store`	A triple store, for example create with new.rdf().
`subject`	URI of the subject.
`predicate`	URI of the predicate.
`object`	URI of the object.

Value

The update Java Model object containing the existing and new triples.

Author(s)

Egon Willighagen

Examples

```
store = new.rdf()
add.triple(store,
  subject="http://example.org/Subject",
  predicate="http://example.org/Predicate",
  object="http://example.org/Object")
```

`combine.rdf` *Merge to Java Model objects.*

Description

Returns a new Java Model object containing all unique triples from both merged models.

Usage

```
combine.rdf(model1, model2)
```

Arguments

`model1`	The first Java Model to get triples from.
`model2`	The second Java Model to get triples from.

Value

A new Java Model object containing the triples from both models.

Author(s)

Egon Willighagen

Examples

```
model1 = new.rdf()
model2 = new.rdf()
## Not run: model3 = combine.rdf(model1, model2)
```

`construct.rdf` *Run a SPARQL query on a Java Model and construct a new model with the results.*

Description

Runs a query on the triples in a Java Model using the SPARQL language and construct a new model with the results.

Usage

```
construct.rdf(model, sparql)
```

Arguments

`model`	A Java Model object.
`sparql`	The SPARQL query.

Value

A Jena model object containing the results of the query.

Author(s)

Ryan Kohl

Examples

```
store = new.rdf()
add.triple(store,
  subject="http://example.org/Subject",
  predicate="http://example.org/Predicate",
  object="http://example.org/Object")
results = construct.rdf(store, paste(
  "CONSTRUCT { ?instance a <http://example.org/AnotherObject> }",
  "WHERE { ?instance a <http://example.org/Object> }"
))
```

---

construct.remote *Run a SPARQL CONSTRUCT query on a SPARQL end point and construct a new model with the results.*

Description

Runs a query against a SPARQL end point and construct a new model with the results.

Usage

```
construct.remote(endpoint, sparql)
```

Arguments

endpoint	The SPARQL end point.
sparql	The SPARQL query.

Value

A Jena model object containing the results of the query.

Author(s)

Egon Willighagen

Examples

```
## Not run: store = construct.remote("http://rdf.farmbio.uu.se/chembl/sparql",
  paste(
    "CONSTRUCT { ?instance a <http://example.org/Article> } ",
    "WHERE { ?instance a <http://purl.org/ontology/bibo/Article> }",
    "LIMIT 5"
  ))
## End(Not run)
```

`load.rdf` *Load RDF content*

Description

Loads triples from a RDF serialization.

Usage

```
load.rdf(filename, format = "RDF/XML", appendTo=NULL)
```

Arguments

`filename`	Name of the file to read the triples from.
`format`	Format of the RDF file, e.g. RDF/XML.
`appendTo`	Optional Java Model object to which read statements are added.

Value

A Java Model object containing the triples loaded from the file.

Author(s)

Egon Willighagen

Examples

```
## Not run: model = load.rdf("someFile.xml", "RDF/XML")
## Not run: model = new.rdf(ontology=FALSE)
## Not run: load.rdf("someFile.xml", "RDF/XML", model)
```

new.rdf *Create a new RDF triple store object*

Description

Create a new RDF triple store object.

Usage

```
new.rdf(ontology=TRUE)
```

Arguments

ontology	Indicates if the model should be an ontological model (the default).

Value

A Java Model object containing the triples loaded from the file.

Author(s)

Egon Willighagen

Examples

```
store = new.rdf()
store = new.rdf(ontology=FALSE)
```

save.rdf *Save RDF content*

Description

Saves triples to a RDF serialization.

Usage

```
save.rdf(store, filename, format = "RDF/XML")
```

Arguments

store	A triple store, for example create with new.rdf().
filename	Name of the file to read the triples from.
format	Format of the RDF file, e.g. RDF/XML.

Value

A Java Model object containing the triples loaded from the file.

Author(s)

Egon Willighagen

Examples

```
store = new.rdf()
## Not run: save.rdf(store, "someFile.xml", "N3")
```

---

sparql.rdf *Run a SPARQL query on a Java Model.*

Description

Runs a query on the triples in a Java Model using the SPARQL language.

Usage

```
sparql.rdf(model, sparql, rowvarname)
```

Arguments

model	A Java Model object.
sparql	The SPARQL query.
rowvarname	Optional SPARQL variable name (without the ?) for which the values will be used as row names of the result matrix.

Value

A matrix object containing the results of the query.

Author(s)

Egon Willighagen

Examples

```
store = new.rdf()
sparql.rdf(store, paste(
  "SELECT ?subject ?predicate ?object {",
  "  ?subject ?predicate ?object",
  "} LIMIT 5"
))
sparql.rdf(store, paste(
  "SELECT ?subject ?predicate ?object {",
```

```
  "  ?subject ?predicate ?object",
  "} LIMIT 5"),
  rowvarname="subject"
)
```

`sparql.remote` *Run a SPARQL query on a remote SPARQL end point.*

Description

Runs a query against a SPARQL end point.

Usage

```
sparql.remote(endpoint, sparql, rowvarname, user, password, jena)
```

Arguments

`endpoint`	The SPARQL end point.
`sparql`	The SPARQL query.
`rowvarname`	Optional SPARQL variable name (without the ?) for which the values will be used as row names of the result matrix.
`user`	Optional user name for HTTP authentication.
`password`	Optional password for HTTP authentication.
`jena`	Boolean to indicate if Jena should be used for the remote SPARQL querying, or Apache's HttpClient.

Value

A matrix object containing the results of the query.

Author(s)

Egon Willighagen

Examples

```
## Not run: sparql.remote("http://rdf.farmbio.uu.se/chembl/sparql",
  paste(
    "SELECT DISTINCT ?instance",
    "WHERE { ?instance a <http://purl.org/ontology/bibo/Article> }",
    "LIMIT 5"
  )
)
## End(Not run)

# with authentication
```

```
## Not run: sparql.remote("http://rdf.farmbio.uu.se/chembl/sparql",
  paste(
    "SELECT DISTINCT ?predicate",
    "WHERE { [] ?predicate [] }"
  ),
  user="user", password="password"
)
## End(Not run)
```

---

summarize.rdf *Summarized the content of a Java Model.*

Description

Summarized the content of a Java Model, including the number of triples.

Usage

```
summarize.rdf(model)
```

Arguments

model	A Java Model object.

Value

Results are returned to the console.

Author(s)

Egon Willighagen

Examples

```
store = new.rdf()
summarize.rdf(store)
```

Index