Package 'atlas'

October 12, 2022

Type Package

Title Stanford 'ATLAS' Search Engine API

Version 1.0.0

Author Vladimir Polony

Maintainer Vladimir Polony <podalv@gmail.com>

Description

Stanford 'ATLAS' (Advanced Temporal Search Engine) is a powerful tool that allows constructing cohorts of patients extremely quickly and efficiently. This package is designed to interface directly with an instance of 'ATLAS' search engine and facilitates API queries and data dumps. Prerequisite is a good knowledge of the temporal language to be able to efficiently construct a query.
More information available at <https://shahlab.stanford.edu/start>.

License GPL-3

URL https://shahlab.stanford.edu/start

Imports httr, testthat

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1.9000

NeedsCompilation no

Repository CRAN

Date/Publication 2018-08-10 17:20:11 UTC

R topics documented:

atlas.connect *Connects to ATLAS instance*

Description

Attempts to connect to ATLAS instance using URL:PORT

Usage

```
atlas.connect(url)
```

Arguments

url	url address of a running ATLAS instance, usually containing port information

Value

data frame containing connection information used for all other accessory functions

Examples

```
atlas.connect("http://localhost:8080")
```

atlas.contains *Returns the statistics information*

Description

Returns the statistics information

Usage

```
atlas.contains(connection, patient_id)
```

Arguments

connection	connection object returned from connect(url) function
patient_id	numerical id of the patient

Value

TRUE or FALSE

Examples

```
atlas.contains(atlas.connect('http://localhost:8080'), 123)
```

atlas.csv — *Queries ATLAS with a CSV() command and imports the contents of the csv into a data frame*

Description

Queries ATLAS with a CSV() command and imports the contents of the csv into a data frame

Usage

```
atlas.csv(connection, query, file_name = NULL)
```

Arguments

connection	connection object returned from connect(url) function
query	ATLAS CSV query
file_name	if specified, stores the csv into the file_name, otherwise the temporary file used to download the data will be deleted after the data.frame is generated

Value

data frame containing CSV file

Examples

```
atlas.csv(atlas.connect('http://localhost:8080'), 'CSV(ICD9=250.50, CPT, LABS, ICD9)')
atlas.csv(atlas.connect('http://localhost:8080'), 'CSV(ICD9=250.50, CPT, LABS, ICD9)',
          '/output.csv')
```

atlas.dump — *Dumps patient from ATLAS to a file on disk*

Description

Dumps patient from ATLAS to a file on disk

Usage

```
atlas.dump(connection, patient_id, path, selection_query = NULL,
  contains_start = FALSE, contains_end = FALSE)
```

Arguments

connection	connection object returned from connect(url) function
patient_id	numerical id of the patient
path	path where to store the generated files
selection_query	returns only the part of patient's data that intersects with the result of the selection_query
contains_start	the dumped time interval's start has to be intersecting the selection_query
contains_end	the dumped time interval's end has to be intersecting the selection_query

Value

data frame containing patient IDs and time intervals (optional)

Examples

```
atlas.dump(atlas.connect('http://localhost:8080'), 123, '/path/to/dump/files/')
atlas.dump(atlas.connect('http://localhost:8080'), 123, '/path/', 'ICD9=250.50', TRUE, TRUE)
```

---

atlas.query *Queries ATLAS and returns a list of patient IDs*

Description

Queries ATLAS and returns a list of patient IDs

Usage

```
atlas.query(connection, query, output_time = FALSE)
```

Arguments

connection	connection object returned from connect(url) function
query	ATLAS query
output_time	equivalent to wrapping the query in OUTPUT() command. Togerther with patient IDs outputs each time interval in patient's timeline when the query was evaluated as true

Value

data frame containing patient IDs and time intervals (optional)

Examples

```
atlas.query(atlas.connect('http://localhost:8080'), 'ICD9=250.50')
atlas.query(atlas.connect('http://localhost:8080'), 'ICD9=250.50', TRUE)
```

`atlas.status` *Returns the status of the ATLAS*

Description

Returns the status of the ATLAS

Usage

```
atlas.status(connection)
```

Arguments

`connection`	connection object returned from connect(url) function

Value

data frame containing patient IDs and time intervals (optional)

Examples

```
atlas.status(atlas.connect('http://localhost:8080'))
```

Index