

Package ‘RonFHIR’

October 18, 2017

Type Package

Title Read and Search Interface to the 'HL7 FHIR' REST API

Version 0.1.0

Description R on FHIR is an easy to use wrapper around the 'HL7 FHIR' REST API (STU 3). It provides tools to easily read and search resources on a FHIR server and brings the results into the R environment. R on FHIR is based on the FhirClient of the official 'HL7 FHIR .NET API', also made by Furore Health Informatics.

License BSD 3-clause License + file LICENSE

Imports R6, httr, jsonlite

Suggests testthat

Encoding UTF-8

LazyData true

Depends R (>= 3.1.0)

NeedsCompilation no

RoxygenNote 6.0.1

Author Sander Laverman [aut, cre],
Furore Health Informatics B.V. [aut, cph]

Maintainer Sander Laverman <s.laverman@furore.com>

Repository CRAN

Date/Publication 2017-10-18 17:17:52 UTC

R topics documented:

fhirClient	2
searchParams	3
Index	6

 fhirClient

fhirClient

Description

Read and search only client in R for FHIR STU 3. Based on [the official HL7 FHIR .NET API](#).

Usage

```
client <- fhirClient$new(endpoint)

client$read(location, summaryType = NULL)
client$search(resourceType, criteria = NULL, includes = NULL, pageSize = NULL, summaryType = NULL)
client$searchById(resourceType, id, includes = NULL, summaryType = NULL)
client$wholeSystemSearch(criteria = NULL, includes = NULL, pageSize = NULL, summaryType = NULL)
client$searchParams(params, resourceType = NULL)
client$continue(bundle)

print(client)
```

Arguments

client A fhirClient object.

endpoint The URL of the server to connect to.

resourceType The type of resource to search for.

id The id of the Resource to search for.

summaryType Whether to include only return a summary of the Resource(s).

location The url of the Resource to fetch. This can be a Resource id url or a version-specific.

criteria The search parameters to filter the Resources on. Each given string is a combined key/value pair (separated by '=').

includes Paths to include in the search.

pageSize Asks server to limit the number of entries per page returned.

query A searchParams object containing the search parameters.

bundle The bundle as received from the last response.

Details

`$new()` Creates a new fhirClient using a given endpoint. If the endpoint does not end with a slash (/), it will be added.

`$read()` Fetches a typed Resource from a FHIR resource endpoint.

`$search()` Search for Resources of a certain type that match the given criteria.

`$searchById()` Search for Resources based on a Resource's id.

`$wholeSystemSearch()` Search for Resources across the whole server that match the given criteria.

`$searchByQuery()` Search for Resources based on a `searchParams` object.

`$continue()` Uses the FHIR paging mechanism to go navigate around a series of paged result Bundles.

`print(p)` or `p$print()` Shows which endpoint is configured.

Examples

```
## Not run:
# Setting up a fhirClient
client <- fhirClient$new("http://vonk.furore.com")
# Read
client$read("Patient/example")

# Search
bundle <- client$search("Patient", c("name=Peter", "address-postalcode=3999"))

while(!is.null(bundle)){
  # Do something useful
  bundle <- client$continue(bundle)
}

## End(Not run)
```

searchParams

searchParams

Description

An alternative way to specify a query is by creating a `searchParams` object and pass this to the `fhirClient's` `searchByQuery`. The `searchParams` class has a set of fluent calls to allow you to easily construct more complex queries. Based on [the official HL7 FHIR .NET API](#).

Usage

```
query <- searchParams$new()

query$select(elements)
query$where(criteria)
query$include(path)
query$orderBy(paramName, sortOrder = "asc")
query$limitTo(count)
query$countOnly()
query$summaryOnly()
query$textOnly()
query$dataOnly()
```

Arguments

- query** A searchParams object that contains all specified search criteria.
- elements** Elements defined at the root level in the Resource.
- criteria** The search parameters to filter the Resources on. Each given string is a combined key/value pair (separated by '=').
- path** Paths to include in the search.
- paramName** Name of the parameter to order by.
- sortOrder** Direction of the order. Can be asc or desc (ascending and descending).
- count** The number of returned Resources per page.

Details

- `$new()` Creates a new searchParams object.
- `$select()` Specify the elements to be returned as part of a Resource.
- `$where()` Specify on which parameters to filter.
- `$include()` Specify the paths to include.
- `$orderBy()` Specify the order to return the results.
- `$limitTo()` Specify how many Resources should be returned in a single page of a Bundle.
- `$countOnly()` Specify to just return a count of the matching Resources, without returning the actual matches.
- `$summaryOnly()` Specify to return only those elements marked as "summary" in the base definition of the Resource(s).
- `$textOnly()` Specify to return only the "text" element, the 'id' element, the 'meta' element, and only top-level mandatory elements.
- `$dataOnly()` Specify to remove the text element.

Examples

```
## Not run:
# Setting up a fhirClient
client <- fhirClient$new("http://vonk.furore.com")

# Creating a new searchParams object
query <- searchParams$new()
query$select(c("name", "birthDate"))$where("given:exact=Peter")$orderBy("family")

peters <- client$searchByQuery(query, "Patient")
# equivalent:
# client$search("Patient", c("_elements=name,birthDate","given:exact=Peter", "_sort=family"))

while(!is.null(bundle)){
  # Do something useful
  peters <- client$continue(peters)
}
```

searchParams

5

End(Not run)

Index

fhirClient, [2](#)
fhirClient's, [3](#)
searchParams, [3](#)