

Package ‘hpds’

October 23, 2019

Type Package

Version 0.1.0

Date 2019-10-17

Title Simplified R library to communicate with HPDS-hosted services on
a PIC-SURE network

Author Nick Benik [aut, cre]

Maintainer Nick Benik <nicholas_benik@hms.harvard.edu>

Description Library that is used to connect to access
HPDS-hosted services/data on a PIC-SURE network.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Depends R (>= 3.1.0)

Imports R6, httr, jsonlite, stringr, hash

R topics documented:

BypassAdapter	2
extract.count	3
extract.dataframe	3
extract.entries	4
extract.keys	4
find.in.dictionary	5
get.resource	6
hpds	6
HpdsAttribList	7
HpdsAttribListKeys	7
HpdsAttribListKeyValues	8
new.query	9
PicSureHpdsBypassConnection	9
PicSureHpdsBypassConnectionAPI	10
PicSureHpdsDictionary	11
PicSureHpdsDictionaryResult	11
PicSureHpdsQuery	12
PicSureHpdsResourceConnection	13

query.anyof.add	14
query.anyof.delete	14
query.crosscounts.add	15
query.crosscounts.delete	15
query.filter.add	16
query.filter.delete	17
query.require.add	17
query.require.delete	18
query.run	18
query.select.add	19
query.select.delete	20
query.show	20
Index	22

BypassAdapter	<i>R6 class that selects a HPDS-hosted resources directly without the use of the PIC-SURE Connection library.</i>
---------------	---

Description

R6 class that selects a HPDS-hosted resources directly without the use of the PIC-SURE Connection library.

Usage

BypassAdapter

Format

PicSureHpdsBypassAdapter object.

Value

Object of [R6Class](#) used to access a HPDS-hosted resource without using a PIC-SURE network.

Methods

- Documentation** For full documentation of each method go to <https://github.com/hms-dbmi/pic-sure-r-adapter-hpds>
- `new(url, token)` This method is used to create new object of this class which connects directly to a HPDS-hosted resource.
- `list()` This method prints a list of UUIDs of all resources hosted by the currently connected PIC-SURE network.
- `useResource(resource_uuid)` This method returns a new `PicSureHpdsResourceConnection` object configured to connect the specified HPDS-hosted resource on the PIC-SURE Network identified by the given `resource_uuid`.

extract.count	<i>Extract the number of results in the given data dictionary lookup.</i>
---------------	---

Description

Extract the number of results in the given data dictionary lookup.

Usage

```
extract.count(dictionary.results, verbose = FALSE)
```

Arguments

dictionary.results	A data dictionary search results object.
verbose	Flag to display additional runtime information.

Value

An integer of how many data dictionary entries were found in

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# asthma.terms <- hpds::find.in.dictionary(resource=myres, term="asthma")
# extract.count(asthma.terms)
```

extract.dataframe	<i>Extract all the results of a given data dictionary lookup in a data frame format.</i>
-------------------	--

Description

Extract all the results of a given data dictionary lookup in a data frame format.

Usage

```
extract.dataframe(dictionary.results, verbose = FALSE)
```

Arguments

dictionary.results	A dictionary results object.
verbose	Flag to display additional runtime information.

Value

Results in a dataframe format.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# asthma.terms <- hpds::find.in.dictionary(resource=myres, term="asthma")
# extract.dataframe(asthma.terms)
```

extract.entries	<i>Extract the entries of all the results in a given data dictionary lookup.</i>
-----------------	--

Description

Extract the entries of all the results in a given data dictionary lookup.

Usage

```
extract.entries(dictionary.results, verbose = FALSE)
```

Arguments

dictionary.results	A dictionary results object.
verbose	Flag to display additional runtime information.

Value

A collection of result entries.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# asthma.terms <- hpds::find.in.dictionary(resource=myres, term="asthma")
# extract.entries(asthma.terms)
```

extract.keys	<i>Extract the unique keys of all the results in a given data dictionary lookup.</i>
--------------	--

Description

Extract the unique keys of all the results in a given data dictionary lookup.

Usage

```
extract.keys(dictionary.results, verbose = FALSE)
```

Arguments

dictionary.results A data dictionary search results object.

verbose Flag to display additional runtime information.

Value

A list of unique keys for all the data dictionary search results.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# asthma.terms <- hpds::find.in.dictionary(resource=myres, term="asthma")
# extract.keys(asthma.terms)
```

find.in.dictionary	<i>Search the data dictionary of a PIC-SURE resource.</i>
--------------------	---

Description

Search the data dictionary of a PIC-SURE resource.

Usage

```
find.in.dictionary(resource, term = "", verbose = FALSE)
```

Arguments

resource A PIC-SURE resource object.

term A string to search for in the resource's data dictionary.

verbose Flag to display additional runtime information.

Value

An object representing the search results.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# asthma.terms <- hpds::find.in.dictionary(resource=myres, term="asthma")
```

<code>get.resource</code>	<i>Get a new reference to a HPDS-based PIC-SURE resource.</i>
---------------------------	---

Description

Get a new reference to a HPDS-based PIC-SURE resource.

Usage

```
get.resource(connection, resourceUUID, verbose = FALSE)
```

Arguments

<code>connection</code>	A PIC-SURE connection object.
<code>resourceUUID</code>	The UUID identity of a Resource hosted via the PIC-SURE connection.
<code>verbose</code>	Flag to display additional runtime information.

Value

An object which provides access to the requested Resource.

Examples

```
# myconn <- picsure::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
```

<code>hpds</code>	<i>hpds: An Adapter Library for HPDS Services on PIC-SURE Networks.</i>
-------------------	---

Description

This package is used to interact with HPDS Services and data hosted on a PIC-SURE network. This library requires the 'picsure' package to handle all connectivity and authentication with the PIC-SURE Network itself.

HpdsAttribList	<i>R6 class used as base class for all query parameter lists - DO NOT CREATE THIS OBJECT DIRECTLY!</i>
----------------	--

Description

R6 class used as base class for all query parameter lists - DO NOT CREATE THIS OBJECT DIRECTLY!

Usage

HpdsAttribList

Format

HpdsAttribList object.

Value

Object of R6Class used to access a HPDS-hosted resource's data dictionary.

Methods

Documentation For full documentation of each method go to <https://github.com/hms-dbmi/picture-r-adapter-hpds>

`new(connection)` This method is used to create new object of this class. DO NOT CREATE THIS OBJECT DIRECTLY!

`add()` This method adds one or more entries to the query parameter list.

`delete(keys)` This method deletes one or more entries from the query parameter list.

`clear()` This method clears all entries from the query parameter list.

`show()` This method displays the entries of the query parameter list.

`getQueryValues()` This is an internally used method that returns the entries for use by the parent query object.

HpdsAttribListKeys	<i>R6 class used to store/manipulate key-only query parameter lists - DO NOT CREATE THIS OBJECT DIRECTLY!</i>
--------------------	---

Description

R6 class used to store/manipulate key-only query parameter lists - DO NOT CREATE THIS OBJECT DIRECTLY!

Usage

HpdsAttribListKeys

Format

`HpdsAttribListKeys` object.

Value

Object of `R6Class` used to access a HPDS-hosted resource's data dictionary.

Methods

Documentation For full documentation of each method go to <https://github.com/hms-dbmi/picture-r-adapter-hpds>

`new(inst_list, help_text)` This method is used to create new object of this class. DO NOT CREATE THIS OBJECT DIRECTLY!

`add()` This method adds one or more entries to the query parameter list.

`delete(keys)` This method deletes one or more entries from the query parameter list.

`clear()` This method clears all entries from the query parameter list.

`show()` This method displays the entries of the query parameter list.

`getQueryValues()` This is an internally used method that returns the entries for use by the parent query object.

`HpdsAttribListKeyValues`

R6 class used to store/manipulate Key+Value query parameter lists - DO NOT CREATE THIS OBJECT DIRECTLY!

Description

R6 class used to store/manipulate Key+Value query parameter lists - DO NOT CREATE THIS OBJECT DIRECTLY!

Usage

`HpdsAttribListKeyValues`

Format

`HpdsAttribListKeyValues` object.

Value

Object of `R6Class` used to access a HPDS-hosted resource's data dictionary.

Methods

Documentation For full documentation of each method go to <https://github.com/hms-dbmi/picture-r-adapter-hpds>

`new(inst_list, help_text)` This method is used to create new object of this class. **DO NOT CREATE THIS OBJECT DIRECTLY!**

`add()` This method adds one or more entries to the query parameter list.

`delete(keys)` This method deletes one or more entries from the query parameter list.

`clear()` This method clears all entries from the query parameter list.

`show()` This method displays the entries of the query parameter list.

`getQueryValues()` This is an internally used method that returns the entries for use by the parent query object.

<code>new.query</code>	<i>Create a new query instance with no restrictions.</i>
------------------------	--

Description

Create a new query instance with no restrictions.

Usage

```
new.query(resource, verbose = FALSE)
```

Arguments

<code>resource</code>	A resource object that the returned query object will execute against.
<code>verbose</code>	Flag to display additional runtime information.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# myquery <- hpds::new.query(resource=myres)
```

<code>PicSureHpdsBypassConnection</code>	<i>R6 class that is a code-shim used by the PicSureBypassAdapter - DO NOT USE THIS OBJECT DIRECTLY!</i>
--	---

Description

R6 class that is a code-shim used by the PicSureBypassAdapter - **DO NOT USE THIS OBJECT DIRECTLY!**

Usage

PicSureHpdsBypassConnection

Format

PicSureHpdsBypassConnection object.

Value

Object of [R6Class](#) used by the PicSureBypassAdapter to access a HPDS-hosted resource without using a PIC-SURE network.

Methods

Documentation For full documentation of each method go to <https://github.com/hms-dbmi/pic-sure-r-adapter-hpds>

PicSureHpdsBypassConnectionAPI

*R6 class that is a code-shim used by the PicSureBypassConnection -
DO NOT USE THIS OBJECT DIRECTLY!*

Description

R6 class that is a code-shim used by the PicSureBypassConnection - DO NOT USE THIS OBJECT DIRECTLY!

Usage

PicSureHpdsBypassConnectionAPI

Format

PicSureHpdsBypassConnectionAPI object.

Value

Object of [R6Class](#) used by the PicSureHpdsBypassConnection to access a HPDS-hosted resource without using a PIC-SURE network.

Methods

Documentation For full documentation of each method go to <https://github.com/hms-dbmi/pic-sure-r-adapter-hpds>

PicSureHpdsDictionary *R6 class that runs searches against a HPDS resource's data dictionary - DO NOT CREATE THIS OBJECT DIRECTLY!*

Description

R6 class that runs searches against a HPDS resource's data dictionary - DO NOT CREATE THIS OBJECT DIRECTLY!

Usage

PicSureHpdsDictionary

Format

PicSureHpdsDictionary object.

Value

Object of [R6Class](#) used to access a HPDS-hosted resource's data dictionary.

Methods

Documentation For full documentation of each method go to <https://github.com/hms-dbmi/pic-sure-r-adapter-hpds>

`new(refHpdsResourceConnection)` This method is used to create new object of this class. DO NOT CREATE THIS OBJECT DIRECTLY!

`find(term=FALSE)` This method returns a `PicSureHpdsDictionaryResult` object containing the results of the search on the HPDS resource's data dictionary.

PicSureHpdsDictionaryResult

R6 class contain the results of a search against a HPDS resource's data dictionary - DO NOT CREATE THIS OBJECT DIRECTLY!

Description

R6 class contain the results of a search against a HPDS resource's data dictionary - DO NOT CREATE THIS OBJECT DIRECTLY!

Usage

PicSureHpdsDictionaryResult

Format

PicSureHpdsDictionaryResult object.

Value

Object of [R6Class](#) used to access a HPDS-hosted resource's data dictionary.

Methods

Documentation For full documentation of each method go to <https://github.com/hms-dbmi/pic-sure-r-adapter-hpds>

`new(results)` This method is used to create new object of this class. DO NOT CREATE THIS OBJECT DIRECTLY!

`count()` This method returns a integer of how many terms were returned by the data dictionary search.

`keys()` This method returns a vector of strings holding the unique record keys of the terms discovered by the data dictionary search.

`entries()` This method returns information about the terms discovered by the data dictionary search.

`DataFrame()` This method returns a dataframe containing the keys and record information that was returned by the data dictionary search.

PicSureHpdsQuery

R6 class used to build a multi-use query to search against a HPDS resource's data - DO NOT CREATE THIS OBJECT DIRECTLY!

Description

R6 class used to build a multi-use query to search against a HPDS resource's data - DO NOT CREATE THIS OBJECT DIRECTLY!

Usage

PicSureHpdsQuery

Format

[PicSureHpdsQuery](#) object.

Value

Object of [R6Class](#) used to access a HPDS-hosted resource's data dictionary.

Methods

Documentation For full documentation of each method go to <https://github.com/hms-dbmi/pic-sure-r-adapter-hpds>

`new(connection)` This method is used to create new object of this class. DO NOT CREATE THIS OBJECT DIRECTLY!

`show()` This method displays a list of all settings specified for the query.

`select()` This method accesses a reference to a query parameter list.

`crosscounts()` This method accesses a reference to a query parameter list.

`require()` This method accesses a reference to a query parameter list.
`anyof()` This method accesses a reference to a query parameter list.
`filter()` This method accesses a reference to a query parameter list.
`getCount()` This method returns a count of how many records are discovered by the query.
`getResults()` This method returns the records discovered by the query.
`getResultsDataFrame()` This method returns the discovered records in a dataframe format.
`getRunDetails()` This method returns information the performance of the query.

PicSureHpdsResourceConnection

R6 class that allows access to the data dictionary and query services of a selected HPDS-hosted resources on a PIC-SURE network.

Description

R6 class that allows access to the data dictionary and query services of a selected HPDS-hosted resources on a PIC-SURE network.

Usage

PicSureHpdsResourceConnection

Format

[PicSureHpdsResourceConnection](#) object.

Value

Object of [R6Class](#) used to access the dictionary and query services via the objects it returns.

Methods

Documentation For full documentation of each method go to <https://github.com/hms-dbmi/pic-sure-r-adapter-hpds>

`new(connection, resource_uuid)` This method is used to create new object of this class which uses the passed PicSureConnection object for communication with the PIC-SURE network along with a UUID to identify a HPDS-hosted resource.

`dictionary()` This method returns a PicSureHpdsDictionary object which is used to run lookups against a resources data dictionary.

`query()` This method returns a new PicSureHpdsQuery object configured to run all commands against the previously specified HPDS-hosted resource.

query.anyof.add	<i>Add an "AnyOf" restriction from a query instance.</i>
-----------------	--

Description

Add an "AnyOf" restriction from a query instance.

Usage

```
query.anyof.add(query, keys, verbose = FALSE)
```

Arguments

query	A query instance object.
keys	One or more keys to add to the given query object's AnyOf list.
verbose	Flag to display additional runtime information.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# myquery <- hpds::new.query(resource=myres)
# hpds::query.anyof.add(query=myquery, keys="\\demographics\\SEX\\")
```

query.anyof.delete	<i>Delete an "AnyOf" restriction from a query instance.</i>
--------------------	---

Description

Delete an "AnyOf" restriction from a query instance.

Usage

```
query.anyof.delete(query, keys, verbose = FALSE)
```

Arguments

query	A query instance object.
keys	One or more keys to delete from the given query object's AnyOf list.
verbose	Flag to display additional runtime information.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# myquery <- hpds::new.query(resource=myres)
# hpds::query.anyof.add(query=myquery, keys="\demographics\SEX")
## ...oops, added wrong term, lets delete it...
# hpds::query.anyof.delete(query=myquery, keys="\demographics\SEX")
```

query.crosscounts.add *Add a "cross-counts" restriction to a query instance.*

Description

Add a "cross-counts" restriction to a query instance.

Usage

```
query.crosscounts.add(query, keys, verbose = FALSE)
```

Arguments

query	A query instance object.
keys	One or more keys to add to the given query object's cross-count list.
verbose	Flag to display additional runtime information.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# myquery <- hpds::new.query(resource=myres)
# hpds::query.crosscounts.add(query=myquery, keys="\demographics\SEX")
```

query.crosscounts.delete
Delete a "cross-counts" restriction from a query instance.

Description

Delete a "cross-counts" restriction from a query instance.

Usage

```
query.crosscounts.delete(query, keys, verbose = FALSE)
```

Arguments

query	A query instance object.
keys	One or more keys to delete from the given query object's cross-count list.
verbose	Flag to display additional runtime information.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# myquery <- hpds::new.query(resource=myres)
# hpds::query.crosscounts.add(query=myquery, keys="\\demographics\\SEX\\")
## ...oops, added wrong term, lets delete it...
# hpds::query.crosscounts.delete(query=myquery, keys="\\demographics\\SEX\\")
```

query.filter.add	<i>Add a "Filter" restriction to a query instance.</i>
------------------	--

Description

Add a "Filter" restriction to a query instance.

Usage

```
query.filter.add(query, keys, verbose = FALSE)
```

Arguments

query	A query instance object.
keys	One or more keys to add to the given query object's filter list.
verbose	Flag to display additional runtime information.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# myquery <- hpds::new.query(resource=myres)
# hpds::query.select.add(query=myquery, keys="\\demographics\\SEX\\", values="Male")
```

query.filter.delete	<i>Delete a "Filter" restriction from a query instance.</i>
---------------------	---

Description

Delete a "Filter" restriction from a query instance.

Usage

```
query.filter.delete(query, keys, verbose = FALSE)
```

Arguments

query	A query instance object.
keys	One or more keys to delete from the given query object's filter list.
verbose	Flag to display additional runtime information.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# myquery <- hpds::new.query(resource=myres)
# hpds::query.filter.add(query=myquery, keys="\\demographics\\SEX\\", values="Male")
## ...oops, added wrong term, lets delete it...
# hpds::query.filter.delete(query=myquery, keys="\\demographics\\SEX\\")
```

query.require.add	<i>Add a "require" restriction to a query instance.</i>
-------------------	---

Description

Add a "require" restriction to a query instance.

Usage

```
query.require.add(query, keys, verbose = FALSE)
```

Arguments

query	A query instance object.
keys	One or more keys to add to the given query object's require list.
verbose	Flag to display additional runtime information.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# myquery <- hpds::new.query(resource=myres)
# hpds::query.require.add(query=myquery, keys="\demographics\SEX\")
```

query.require.delete	<i>Delete a "require" restriction from a query instance.</i>
----------------------	--

Description

Delete a "require" restriction from a query instance.

Usage

```
query.require.delete(query, keys, verbose = FALSE)
```

Arguments

query	A query instance object.
keys	One or more keys to delete from the given query object's require list.
verbose	Flag to display additional runtime information.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# myquery <- hpds::new.query(resource=myres)
# hpds::query.require.add(query=myquery, keys="\demographics\SEX\")
## ...oops, added wrong term, lets delete it...
# hpds::query.require.delete(query=myquery, keys="\demographics\SEX\")
```

query.run	<i>Run a query instance using any restrictions that have been added to it.</i>
-----------	--

Description

Run a query instance using any restrictions that have been added to it.

Usage

```
query.run(query, result.type = "dataframe", verbose = FALSE)
```

Arguments

query	A query instance object.
result.type	A string specifying what type of results to return. Possible values: "count", "results", "dataframe" and "crosscount".
verbose	Flag to display additional runtime information.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# myquery <- hpds::new.query(resource=myres)

## ...modify the query by adding search criteria... ##

# results <- hpds::query.run(query=myquery)
```

query.select.add	<i>Add a "select" restriction to a query instance.</i>
------------------	--

Description

Add a "select" restriction to a query instance.

Usage

```
query.select.add(query, keys, verbose = FALSE)
```

Arguments

query	A query instance object.
keys	One or more keys to add to the given query object's select list.
verbose	Flag to display additional runtime information.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# myquery <- hpds::new.query(resource=myres)
# hpds::query.select.add(query=myquery, keys="\\demographics\\SEX\\")
```

query.select.delete	<i>Delete a "select" restriction from a query instance.</i>
---------------------	---

Description

Delete a "select" restriction from a query instance.

Usage

```
query.select.delete(query, keys, verbose = FALSE)
```

Arguments

query	A query instance object.
keys	One or more keys to delete from the given query object's select list.
verbose	Flag to display additional runtime information.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# myquery <- hpds::new.query(resource=myres)
# hpds::query.select.add(query=myquery, keys="\\demographics\\SEX\\")
## ...oops, added wrong term, lets delete it...
# hpds::query.select.delete(query=myquery, keys="\\demographics\\SEX\\")
```

query.show	<i>Prints to screen the contents of a query instance's restriction parameters.</i>
------------	--

Description

Prints to screen the contents of a query instance's restriction parameters.

Usage

```
query.show(query, verbose = FALSE)
```

Arguments

query	A query instance object.
verbose	Flag to display additional runtime information.

Examples

```
# myconn <- picture::connect(url="http://your.pic-sure.server/PIC-SURE/", token="your-security-token")
# myres <- hpds::get.resource(connection=myconn, resourceUUID="612d00b7-YOUR-UUID-0000-1efe587f0bcd")
# myquery <- hpds::new.query(resource=myres)
# hpds::query.select.add(query=myquery, keys="\\demographics\\SEX\\")
# hpds::query.select.add(query=myquery, keys="\\demographics\\AGE\\")
# hpds::query.filter.add(query=myquery, keys="\\demographics\\SEX\\", values="Female")
# hpds::query.filter.add(query=myquery, keys="\\demographics\\AGE\\", min="10", max="65")
# hpds::query.show(query=myquery)
```

Index

*Topic **data**

- BypassAdapter, [2](#)
- HpdsAttribList, [7](#)
- HpdsAttribListKeys, [7](#)
- HpdsAttribListKeyValues, [8](#)
- PicSureHpdsBypassConnection, [9](#)
- PicSureHpdsBypassConnectionAPI, [10](#)
- PicSureHpdsDictionary, [11](#)
- PicSureHpdsDictionaryResult, [11](#)
- PicSureHpdsQuery, [12](#)
- PicSureHpdsResourceConnection, [13](#)

BypassAdapter, [2](#)

`extract.count`, [3](#)
`extract.dataframe`, [3](#)
`extract.entries`, [4](#)
`extract.keys`, [4](#)

`find.in.dictionary`, [5](#)

`get.resource`, [6](#)

`hpds`, [6](#)
`hpds-package (hpds)`, [6](#)
`HpdsAttribList`, [7](#), [7](#)
`HpdsAttribListKeys`, [7](#), [8](#)
`HpdsAttribListKeyValues`, [8](#), [8](#)

`new.query`, [9](#)

`PicSureHpdsBypassConnection`, [9](#), [10](#)
`PicSureHpdsBypassConnectionAPI`, [10](#), [10](#)
`PicSureHpdsDictionary`, [11](#), [11](#)
`PicSureHpdsDictionaryResult`, [11](#), [11](#)
`PicSureHpdsQuery`, [12](#), [12](#)
`PicSureHpdsResourceConnection`, [13](#), [13](#)

`query.anyof.add`, [14](#)
`query.anyof.delete`, [14](#)
`query.crosscounts.add`, [15](#)
`query.crosscounts.delete`, [15](#)
`query.filter.add`, [16](#)
`query.filter.delete`, [17](#)
`query.require.add`, [17](#)

`query.require.delete`, [18](#)
`query.run`, [18](#)
`query.select.add`, [19](#)
`query.select.delete`, [20](#)
`query.show`, [20](#)

`R6Class`, [2](#), [7](#), [8](#), [10–13](#)