Package: classyfireR (via r-universe)

September 22, 2026

Type Package

Title R Interface to the 'ClassyFire' REST API

Version 0.3.8

Date 2022-02-20

Description Access to the 'ClassyFire' REST API <http://classyfire.wishartlab.com>. Retrieve existing entity classifications and submit new entities for classification.

Depends magrittr

Imports httr, jsonlite, tibble, crayon, clisymbols, cli, purrr, dplyr, stringr, methods, rjson, RSQLite, tidyjson, tidyr

Suggests testthat, covr, knitr, rmarkdown, stringi

License GPL (>= 3)

Encoding UTF-8

URL https://github.com/aberHRML/classyfireR

BugReports https://github.com/aberHRML/classyfireR/issues

RoxygenNote 7.1.2

Roxygen list(markdown = TRUE)

Collate 'allGenerics.R' 'allClasses.R' 'accessor-methods.R' 'get_classification.R' 'internals.R' 'query.R' 'show-method.R' 'zzz.R'

VignetteBuilder knitr

Config/pak/sysreqs libicu-dev libssl-dev

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

Date/Publication 2022-02-20 19:49:11 UTC

RemoteUrl https://github.com/wilsontom/classyfireR

RemoteRef HEAD

RemoteSha ac8c9204f40a0a74276f02994bc1ad14cd160496

Contents

alternative_parents . . . 2
chebi . . . 3
classification . . . 3
ClassyFire-class . . . 4
description . . . 4
descriptors . . . 5
direct_parent . . . 6
get_classification . . . 6
get_query . . . 7
meta . . . 8
open_cache . . . 8
Query-class . . . 9
show,ClassyFire-method . . . 9
submit_query . . . 10
unclassified . . . 10

Index **12**

`alternative_parents` *alternative_parents*

Description

Get the alternative_parents

Get the alternative_parents

Usage

```
alternative_parents(object)

## S4 method for signature 'ClassyFire'
alternative_parents(object)

## S4 method for signature 'Query'
alternative_parents(object)
```

Arguments

`object`	a `Query` S4 object

Value

a `tbl_df` of alternative parents

a `tbl_df` of alternative parents

chebi *chebi*

Description

Get the predicted ChEBI identifications

Get the predicted ChEBI identifications

Usage

```
chebi(object)

## S4 method for signature 'ClassyFire'
chebi(object)

## S4 method for signature 'Query'
chebi(object)
```

Arguments

object	a Query S4 object

Value

a character vector of ChEBI terms

a character vector of ChEBI terms

classification *Accessor Methods for the ClassyFire S4 class*

Description

Get the ClassyFire classification results

Get the ClassyFire classification results

Usage

```
classification(object)

## S4 method for signature 'ClassyFire'
classification(object)

## S4 method for signature 'Query'
classification(object)
```

Arguments

`object` a `Query` S4 object

Value

a `tbl_df` of classifications

a `tbl_df` of classifications

`ClassyFire-class` *ClassyFire Class*

Description

A S4 class to store the results from the ClassyFire RESTful API

Slots

`meta` a list of the input SMILE, InChI Key and classification version used

`classification` a `tibble` of the ClassyFire classification results

`direct_parent` a list of the direct parent

`alternative_parents` a `tibble` of alternative parents

`predicted_chebi` a character vector of the predicted ChEBI terms

`external_descriptors` a `tibble` of any external identificatons

`description` a character string of the compound description

`description` *description*

Description

Get the `ClassyFire` description

Get the `ClassyFire` description

Usage

```
description(object)

## S4 method for signature 'ClassyFire'
description(object)

## S4 method for signature 'Query'
description(object)
```

Arguments

object	a Query S4 object

Value

a character string of the classification description

a character string of the classification description

`descriptors` *descriptors*

Description

Get the external descriptors

Get the external descriptors

Usage

```
descriptors(object)

## S4 method for signature 'ClassyFire'
descriptors(object)

## S4 method for signature 'Query'
descriptors(object)
```

Arguments

object	a Query S4 object

Value

a tbl_df of available external descriptors

a tbl_df of available external descriptors

`direct_parent` *direct_parent*

Description

Get the direct_parent

Get the direct_parent

Usage

```
direct_parent(object)

## S4 method for signature 'ClassyFire'
direct_parent(object)

## S4 method for signature 'Query'
direct_parent(object)
```

Arguments

`object`	a `Query` S4 object

Value

a list of the direct parent

a list of the direct parent

`get_classification` *Get Entity Classification*

Description

Retrieve entity classification from 'http://classyfire.wishartlab.com/entities/'. The optional local cache function enables classification requests with less waiting time. Furthermore, there will be fewer traffic on the classyFire servers. For best high efficiency there is an option for creating a SQLight database to cache results.

Usage

```
get_classification(inchi_key, conn = NULL)
```

Arguments

`inchi_key`	a character string of a valid InChIKey
`conn`	a DBIConnection object, as produced by dbConnect

Value

a ClassyFire S4 object.

See Also

ClassyFire-class

Examples

```
## Not run:

# Valid InChI key where all four classification levels are available
get_classification('BRMWTNUJHUMWMS-LURJTMIESA-N')

# Valid InChI key where only three classification levels are available
get_classification('MDHYEMXUFSJLGV-UHFFFAOYSA-N')

# Invalid InChI key
get_classification('MDHYEMXUFSJLGV-UHFFFAOYSA-B')

## End(Not run)
```

---

get_query *Retrieves the classification results for a given query.*

Description

Retrieves the classification results for a given query.

Usage

```
get_query(query_id, format = c("json", "sdf", "csv"))
```

Arguments

query_id	query_id a numeric value for the ID of the query.
format	a string of the format of the query (either JSON, CSV, or SDF)

Value

the parsed text output displaying the classification results for

meta *meta*

Description

Get the ClassyFire meta data

Get the ClassyFire meta data

Usage

```
meta(object)

## S4 method for signature 'ClassyFire'
meta(object)

## S4 method for signature 'Query'
meta(object)
```

Arguments

object	a Query S4 object

Value

a list of meta data

a list of meta data

---

open_cache *Open Cache*

Description

Creates a SQLight database for the local caching. This database includes already queried Inchikeys and the serialized classification object.

Usage

```
open_cache(dbname = ":memory:")
```

Arguments

dbname	The path to the database file. SQLite keeps each database instance in one single file. The name of the database is the file name, thus database names should be legal file names in the running platform. There are two exceptions: "" will create a temporary on-disk database. The file will be deleted when the connection is closed. ":memory:" or "file::memory:" will create a temporary in-memory database.

Value

conn a DBIConnection object, as produced by dbConnect

`Query-class` *ClassyFire Query Class*

Description

A S4 class to store the results from the ClassyFire RESTful API Queries

Slots

`meta` a 'tibble' of the input SMILE, InChI Key and classification version used

`classification` a `tibbles` of the ClassyFire classification results

`direct_parent` a `tibbles` the direct parent

`alternative_parents` a `tibbles` of alternative parents

`predicted_chebi` a list of character vectors of the predicted ChEBI terms

`external_descriptors` a list of `tibbles` of any external identifications

`description` a list of character vectors for the compound descriptions

`unclassified` a character string of any input structures that failed the ClassyFire classification

---

`show,ClassyFire-method` *show-ClassyFire*

Description

show-ClassyFire

show-Query

Usage

```
## S4 method for signature 'ClassyFire'
show(object)

## S4 method for signature 'Query'
show(object)
```

Arguments

`object` a `Query` S4 object

submit_query *Submits a ClassyFire query in a JSON format.*

Description

Submits a ClassyFire query in a JSON format.

Usage

```
submit_query(label, input, type = "STRUCTURE")
```

Arguments

label	a string the label of the query.
input	a named list or vector of SMILES strings.
type	a string the type of the query.

Value

A list of tibbles named by input name.

Examples

```
## Not run:
input <- c(MOL1 = 'CCCOCC', MOL2 = 'COCC=CCC')
submit_query(label = 'query_test', input = input, type = 'STRUCTURE')

## End(Not run)
```

unclassified *unclassified*

Description

Get the identifiers of any inputs that were not classified

Usage

```
unclassified(object)

## S4 method for signature 'Query'
unclassified(object)
```

Arguments

object	a Query S4 object

Value

a character string of `inputs` that were not successfully classified

Index

alternative_parents, 2
alternative_parents,ClassyFire-method
(alternative_parents), 2
alternative_parents,Query-method
(alternative_parents), 2

chebi, 3
chebi,ClassyFire-method (chebi), 3
chebi,Query-method (chebi), 3
classification, 3
classification,ClassyFire-method
(classification), 3
classification,Query-method
(classification), 3
ClassyFire-class, 4

description, 4
description,ClassyFire-method
(description), 4
description,Query-method (description),
4
descriptors, 5
descriptors,ClassyFire-method
(descriptors), 5
descriptors,Query-method (descriptors),
5
direct_parent, 6
direct_parent,ClassyFire-method
(direct_parent), 6
direct_parent,Query-method
(direct_parent), 6

get_classification, 6
get_query, 7

meta, 8
meta,ClassyFire-method (meta), 8
meta,Query-method (meta), 8

open_cache, 8

Query-class, 9

show,ClassyFire-method, 9
show,Query-method
(show,ClassyFire-method), 9
submit_query, 10

unclassified, 10
unclassified,Query-method
(unclassified), 10