

Package ‘classyfireR’

December 6, 2018

Type Package

Title R Interface to the ClassyFire RESTful API

Version 0.1.2

Date 2018-12-06

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

Depends magrittr

Imports httr, jsonlite, tibble, crayon, clisymbols

Suggests testthat, covr, dplyr, purrr, tidyr

License GPL (>= 3)

Encoding UTF-8

LazyData true

URL <https://github.com/wilsontom/classyfireR>

BugReports <https://github.com/wilsontom/classyfireR/issues>

RoxygenNote 6.1.0

Roxygen list(markdown = TRUE)

NeedsCompilation no

Author Tom Wilson [aut, cre]

Maintainer Tom Wilson <tpw2@aber.ac.uk>

R topics documented:

entity_classification	2
get_status_code	3
retrieve_classification	4
submit_classification	4

Index	6
-------	---

entity_classification *Entity Classification*

Description

Retrieve entity classification from 'http://classyfire.wishartlab.com/entities/'

Usage

```
entity_classification(inchi_key)
```

Arguments

inchi_key a character string of a valid InChIKey

Value

a tibble containing the following;

- **Level** Classification level (kingdom, superclass, class and subclass)
- **Classification** The compound classification
- **CHEMONT** Chemical Ontology Identification code

Examples

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

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

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

# Using `dplyr` a vector of InChI Keys can be submitted and easily parsed
library(dplyr)
library(purrr)
library(tidyr)

keys <- c(
  'BRMWTNUJHUMWMS-LURJTMIESA-N',
  'XFNJVJPLKCPIBV-UHFFFAOYSA-N',
  'TYEYBOSBBBHJIV-UHFFFAOYSA-N',
  'AFENDNXGAFYKQO-UHFFFAOYSA-N',
  'WHEUWNKSCXYKBU-QPWUGHHJSA-N',
  'WHBMMWSBFZVSSR-GSVOUTGSA-N')
```

```
classification_list <- map(keys, entity_classification)

classification_list <- map(classification_list, ~{select(.,-CHEMONT)})

spread_tibble <- purrr::map(classification_list, ~{
  spread(., Level, Classification)
}) %>% bind_rows() %>% data.frame()

rownames(spread_tibble) <- keys

classification_df <- data.frame(InChIKey = rownames(spread_tibble),
  Kingdom = spread_tibble$kingdom,
  SuperClass = spread_tibble$superclass,
  Class = spread_tibble$class,
  SubClass = spread_tibble$subclass)

print(classification_df)
```

get_status_code

Get Status Code

Description

Retrieve the status code of a new submission for entity classification

Usage

```
get_status_code(query_id)
```

Arguments

query_id a numeric value for the query id

Value

a list of query_id and classification status; either In progress or Done

Examples

```
get_status_code(2813259)
```

`retrieve_classification`*Retrieve classification results*

Description

Retrieve classification results from a new submission

Usage

```
retrieve_classification(query_id)
```

Arguments

`query_id` a numeric value for the query id

Value

a tibble containing the following;

- **Level** Classification level (kingdom, superclass, class and subclass)
- **Classification** The compound classification
- **CHEMONT** Chemical Ontology Identification code

Examples

```
retrieve_classification(2813259)
```

`submit_classification` *Submit InChI Code for Classification*

Description

Submit and new entity for classification using the ClassyFire webserver

Usage

```
submit_classification(query, label, type = "STRUCTURE")
```

Arguments

`query` a character string of InChI Code or SMILE
`label` a character string of the query name
`type` the label type (Default = STRUCTURE)

Value

if the classification has completed; a `tibble` containing the following;

- **Level** Classification level (kingdom, superclass, class and subclass)
- **Classification** The compound classification
- **CHEMONT** Chemical Ontology Identification code

Index

`entity_classification`, [2](#)

`get_status_code`, [3](#)

`retrieve_classification`, [4](#)

`submit_classification`, [4](#)