Package ‘repec’

August 31, 2018

Type Package
Title Access RePEc Data Through API
Version 0.1.0
Description Utilities for accessing RePEc (Research Papers in Economics) through a RESTful API. You can request a code and get detailed information at the following page: <https://ideas.repec.org/api.html>.
License MIT + file LICENSE
URL https://github.com/chrmongeau/repec
BugReports https://github.com/chrmongeau/repec/issues
Imports jsonlite, utils
Encoding UTF-8
LazyData true
RoxygenNote 6.1.0
NeedsCompilation no
Author Christian Alexander Mongeau Ospina [aut, cre]
Maintainer Christian Alexander Mongeau Ospina <christian+rpkg-repec@mongeau.net>
Repository CRAN
Date/Publication 2018-08-31 18:20:08 UTC

R topics documented:

repec-package .................................................. 2
areyouthere .................................................... 3
getauthorname .................................................. 3
getauthorrecordfull .......................................... 4
getauthorrecorddraw .......................................... 5
getauthorsforitem ............................................. 5
getauthorshortid .............................................. 6
getauthorstats .................................................. 7
Description

repec provides a way to interact with the RePEc API (Application Programming Interface) in order to get various information stored on RePEc about authors, items, institutions, etc.

Details

By using the functions of this package, you will be able to import RePEc information in R. You will need to ask for an access code by following the instructions at https://ideas.repec.org/api.html. The code will be needed when calling most of the functions by using the code parameter. One way to avoid passing explicitly the code is to create the REPEC_API_KEY environment variable. The code below can be put in your .RProfile so that it is always available when you start R:

```r
# YOURCODE is your personal access code
Sys.setenv(REPEC_API_KEY = 'YOURCODE')
```

Author(s)

Maintainer: Christian Alexander Mongeau Ospina <christian+rpkg-repec@mongeau.net> (0000-0002-4047-3924)

See Also

Useful links:

- [https://github.com/chrMongeau/repec](https://github.com/chrMongeau/repec)
- Report bugs at [https://github.com/chrMongeau/repec/issues](https://github.com/chrMongeau/repec/issues)
areyouthere  

| areyouthere | Verifies whether the server is alive |

### Description
Verifies whether the server is alive

### Usage
```r
areyouthere()
```

```r
are_you_there()
```

### Value
TRUE if the server is alive, FALSE otherwise.

### Examples
```r
## Not run:
areyouthere()
```

```r
## End(Not run)
```

---

getauthornep  

| getauthornep | Get the NEP fields for the papers of an author |

### Description
Get the NEP fields for the papers of an author

### Usage
```r
getauthornep(id, code = NA)
```

```r
get_author_nep(id, code = NA)
```

### Arguments
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>id</code></td>
<td>Short-ID.</td>
</tr>
<tr>
<td><code>code</code></td>
<td>User code required to use API. Included in every API call. If not supplied, the REPEC_API_KEY environment variable will be used.</td>
</tr>
</tbody>
</table>

### Value
```r
data.frame
```
getauthorrecordfull

Provides info about author, mostly in the form of handles

Description

Provides info about author, mostly in the form of handles

Usage

getauthorrecordfull(id, code = NA)
get_author_record_full(id, code = NA)

Arguments

id Short-ID.
code User code required to use API. Included in every API call. If not supplied, the REPEC_API_KEY environment variable will be used.

Value

data.frame

Examples

## Not run:
getauthorrecordfull('pmo915', code = YOURCODE)

## End(Not run)
getauthorrecordraw  

Provides extended info about author

Description

Provides extended info about author

Usage

getauthorrecordraw(id, code = NA)

get_author_record_raw(id, code = NA)

Arguments

id  
  Short-ID.

code  
  User code required to use API. Included in every API call. If not supplied, the REPEC_API_KEY environment variable will be used.

Value

data.frame. Note: columns article and paper are data.frames.

Examples

## Not run:
getauthorrecordraw('pmo915', code = YOURCODE)

## End(Not run)

getauthorsforitem  

Get Authors for an item

Description

Get Authors for an item

Usage

getauthorsforitem(id, code = NA)

get_authors_for_item(id, code = NA)
Arguments

- **id**: RePEc handle for an item.
- **code**: User code required to use API. Included in every API call. If not supplied, the `REPEC_API_KEY` environment variable will be used.

Value
data.frame.

Examples

```r
## Not run:

## End(Not run)
```

---

**getauthorshortid**  
*Get Short-ID and name from string in name or email*

Description

Get Short-ID and name from string in name or email

Usage

```r
getauthorshortid(x, code = NA)
```

```r
get_author_short_id(x, code = NA)
```

Arguments

- **x**: string.
- **code**: User code required to use API. Included in every API call. If not supplied, the `REPEC_API_KEY` environment variable will be used.

Value
data.frame

Examples

```r
## Not run:
getauthorshortid('christian', code = YOURCODE)

## End(Not run)
```
getauthorstats  

Provides Author's statistics

Description

Provides Author's statistics

Usage

getauthorstats(id, code = NA)

get_author_stats(id, code = NA)

Arguments

  id  Short-ID.
  code  User code required to use API. Included in every API call. If not supplied, the REPEC_API_KEY environment variable will be used.

Value

  Author's statistics.

Examples

  ## Not run:
  getauthorstats('pmo915', code = YOURCODE)

  ## End(Not run)

getauthortwitter  

Provides Twitter handle

Description

Provides Twitter handle

Usage

getauthortwitter(id, code = NA)

get_author_twitter(id, code = NA)
**getfirstpubyear**

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Short-ID.</td>
</tr>
<tr>
<td>code</td>
<td>User code required to use API. Included in every API call. If not supplied, the REPEC_API_KEY environment variable will be used.</td>
</tr>
</tbody>
</table>

**Value**

String with the Twitter author linked to the Short-ID.

**Examples**

```r
## Not run:
getauthor twitter('pmo915', code = YOURCODE)

## End(Not run)
```

---

**getfirstpubyear**  
*Provides Author's first publication year*

**Description**

Provides Author's first publication year

**Usage**

getfirstpubyear(id, code = NA)

get_first_pub_year(id, code = NA)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Short-ID.</td>
</tr>
<tr>
<td>code</td>
<td>User code required to use API. Included in every API call. If not supplied, the REPEC_API_KEY environment variable will be used.</td>
</tr>
</tbody>
</table>

**Value**

Author's first publication year.

**Examples**

```r
## Not run:
getfirstpubyear('pmo915', code = YOURCODE)

## End(Not run)
```
### getgenealogy

*Provides Author's genealogy*

**Description**

Provides Author's genealogy

**Usage**

```r
getgenealogy(id, code = NA)
get_genealogy(id, code = NA)
```

**Arguments**

- `id` Short-ID.
- `code` User code required to use API. Included in every API call. If not supplied, the `REPEC_API_KEY` environment variable will be used.

**Value**

Author's genealogy.

**Examples**

```r
## Not run:
getgenealogy('pmo915', code = YOURCODE)

## End(Not run)
```

### gethindex

*Provides H-index*

**Description**

Provides H-index

**Usage**

```r
gethindex(id, code = NA)
get_h_index(id, code = NA)
```
getinstrecord

Arguments

<table>
<thead>
<tr>
<th>id</th>
<th>Short-ID.</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>User code required to use API. Included in every API call. If not supplied, the REPEC_API_KEY environment variable will be used.</td>
</tr>
</tbody>
</table>

Value

H-index.

Examples

```r
# Not run:
getinstrecord('repec:edi:dechiit', code = YOURCODE)

# End(Not run)
```

getinstrecord  
*Get record of Institution*

Description

Get record of Institution

Usage

```r
getinstrecord(id, code = NA)
get_inst_record(id, code = NA)
```

Arguments

<table>
<thead>
<tr>
<th>id</th>
<th>RePEc handle for an institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>User code required to use API. Included in every API call. If not supplied, the REPEC_API_KEY environment variable will be used.</td>
</tr>
</tbody>
</table>

Value

data.frame.

Examples

```r
# Not run:
getinstrecord('RePEc:edi:dechiit', code = YOURCODE)

# End(Not run)
```
getjelforitem

Get JEL codes for item

Description

Get JEL codes for item

Usage

getjelforitem(id, code = NA)
get_jel_foritem(id, code = NA)

Arguments

id
RePEc handle for an item.

code
User code required to use API. Included in every API call. If not supplied, the REPEC_API_KEY environment variable will be used.

Value

Vector with JEL codes.

Examples

## Not run:

## End(Not run)

repec_api_with_id

Generic API call, with an ID

Description

Generic API call, with an ID

Usage

repec_api_with_id(id = NA, method = NA, code = NA)
Arguments

id Short-ID.
method API method to call.
code User code required to use API. Included in every API call. If not supplied, the REPEC_API_KEY environment variable will be used.

Value

String with the Twitter author linked to the Short-ID.

Examples

## Not run:
repec_api_with_id(id = 'pmo915',
method = 'getauthortwitter', code = YOURCODE)

## End(Not run)

---

testip **Verifies whether the connection is performed from the registered IP**

Description

Verifies whether the connection is performed from the registered IP

Usage

testip(code = NA)

test_ip(code = NA)

Arguments

code User code required to use API. Included in every API call. If not supplied, the REPEC_API_KEY environment variable will be used.

Value

TRUE if the connection is from a registered IP, FALSE otherwise.

Examples

## Not run:
testip(YOURCODE)

## End(Not run)
whatismyip

whatismyip

Description

Returns the IP address you use to call the API

Usage

whatismyip()

what_is_myip()

Value

String with IP.

Examples

## Not run:
whatismyip()

## End(Not run)
Index

are_you_there (areyouthere), 3
areyouthere, 3
get_author_nep (getauthornep), 3
get_author_record_full
 (getauthorrecordfull), 4
get_author_record_raw
 (getauthorrecordraw), 5
get_author_short_id (getauthorshortid), 6
get_author_stats (getauthorstats), 7
get_author_twitter (getauthortwitter), 7
get_authors_for_item
 (getauthorsforitem), 5
get_first_pub_year (getfirstpubyear), 8
get_genealogy (getgenealogy), 9
get_h_index (gethindex), 9
get_inst_record (getinstrecord), 10
get_jel_foritem (getjelforitem), 11
getauthornep, 3
getauthorrecordfull, 4
getauthorrecordraw, 5
getauthorsforitem, 5
getauthorshortid, 6
getauthorstats, 7
getauthortwitter, 7
getfirstpubyear, 8
getgenealogy, 9
gethindex, 9
getinstrecord, 10
getjelforitem, 11
repec (repec-package), 2
repec-package, 2
repec_api_with_id, 11
test_ip (testip), 12
testip, 12
what_is_myip (whatismyip), 13
whatismyip, 13