1 The iuropa package

An `R` package to access data from the IUROPA API. This package provides an easy-to-use R interface for the IUROPA API. This API provides access to the IUROPA CJEU Database Platform.

The CJEU Database Platform is the largest and most comprehensive research-oriented database on the CJEU. It includes data on CJEU cases, decisions, and judges collected from the official register of the Court, InfoCuria (the Court's official database), and EUR-Lex (the EU's official database of legal documents). All of the data has been cleaned and cross-validated and is research-ready.

The database contains eight datasets that cover CJEU cases, the parties in each case, CJEU decisions (e.g., judgments, orders, Advocate General opinions, etc.), the judges that participated in each decision, the legal procedures associated with each decision (e.g., references for a preliminary ruling, actions for annulment, appeals, etc.), submissions of observations and interventions with respect to each decision, citations to other EU legal documents in each decision, and CJEU judges and Advocates General.

1.1 About the IUROPA Project

The IUROPA Project (ius + Europa) is a multidisciplinary platform for research on judicial politics in the European Union (EU). The project brings together a network of scholars with a background in law and political science. IUROPA studies the conditions for judicial independence and rule of law against the backdrop of ongoing developments with regards to the judicialization of politics (courts making decisions with political implications) and politicization of courts (efforts from political actors to influence courts).

A key feature of IUROPA is the development of a comprehensive database with detailed information on cases and actors of the Court of Justice of the European Union: The CJEU Database. The database includes information on the outcomes, processes and actors involved in the decision-making of the CJEU, compiled into a research-friendly format. The information has been collected from public sources and prepared and analyzed by means of automatic and manual coding.

You can learn more about the IUROPA Project on our website.

1.2 Installation

You can install the latest development version of the `iuropa` package from GitHub:

```
# install.packages("devtools")
devtools::install_github("jfjelstul/iuropa")
```

1.3 Citation

If you use data from the `iuropa` package in a project or paper, please cite the package.

The citation for the package is:

> Joshua Fjelstul (2021). iuropa: An R Interface to IUROPA API. R package version 0.1.0.

The `BibTeX` entry for the package is:

```
@Manual{,
  title = {iuropa: An R Interface to the IUROPA API},
  author = {Joshua Fjelstul},
  year = {2022},
  note = {R package version 0.1.0},
}
```

If you use data from the CJEU Database Platform, please also cite the database and the paper that introduces the database.

The citation for the database is:

> Brekke, Stein Arne, Joshua Fjelstul, Silje Synnøve Lyder Hermansen and Daniel Naurin. 2021. "The CJEU Database Platform: Decisions and Decision-Makers" (Stable Release 0.1), in Lindholm, Johan, Daniel Naurin, Urska Sadl, Anna Wallerman Ghavanini, Stein Arne Brekke, Joshua Fjelstul, Silje Synnøve Lyder Hermansen, Olof Larsson, Andreas Moberg, Moa Nässtrom, Michal Ovádek, Tommaso Pavone, and Philipp Schroeder, The Court of Justice of the European Union Database, IUROPA, URL: http://iuropa.pol.gu.se/.

The `BibTex` entry for the database is:

```
@incollection{CJEUDatabasePlatform,
  author = {Brekke, Stein Arne and Fjelstul, Joshua and Hermansen, Silje Synn{\o}ve
    Lyder and Naurin, Daniel},
  booktitle = {The Court of Justice of the European Union Database},
  editor = {Lindholm, Johan and Naurin, Daniel and {\v{S}}adl, Ur{\v{s}}ka and
    {Wallerman Ghavanini}, Anna and Brekke, Stein Arne and Fjelstul, Joshua and
    Hermansen, Silje Synn{\o}ve Lyder and Larsson, Olof and Moberg, Andreas and
    N{\"{a}}sstr{\"{o}}m, Moa and Ov{\'{a}}dek, Michal and Pavone, Tommaso and
    Schroeder, Philipp},
  publisher = {IUROPA},
  title = {{The CJEU Database Platform: Decisions and Decision-Makers}},
  url = {http://iuropa.pol.gu.se},
  year = {2021}
}
```

The citation for the paper is:

> Stein Arne Brekke, Joshua C. Fjelstul, Silje Synnøve Lyder Hermansen, and Daniel Naurin. 2021. The CJEU Database Platform: Decisions and Decision-Makers. Working paper.

The `BibTeX` entry for the paper is:

```
@unpublished{CJEUDatabasePlatformPaper,
  author = {Brekke, Stein Arne and Fjelstul, Joshua C. and Hermansen, Silje Synn{\o}ve
    Lyder and Naurin, Daniel},
  title = {{The CJEU Database Platform: Decisions and Decision-Makers}},
  year = {2021}
}
```

1.4 Problems

If you notice an error in the data or a bug in the `R` package, please report it here.

2 Vignette: Downloading data on chamber composition

Suppose we want to download data on which judges participated in each Court of Justice judgment since 2004. This data is available from the `assignments` dataset of the CJEU Database Platform.

We can easily get exactly the data we're looking for right from `R` in just a few easy steps using the `iuropa` package, which is an `R` interface for the `iuropa` API.

2.1 Authenticating with the IUROPA API

To download data from the IUROPA API, we need to use the `authenticate()` function to create a session.

We'll need to provide a username and password (placeholders are used in the example code). You can contact us to get a IUROPA account.

```
session <- authenticate(
  username = "USERNAME",
  password = "PASSWORD"
)
```

```
## Authenticating credentials with the IUROPA API...
## Authentication successful! You will stay connected for 30 minutes.
```

The `session` object has the class `iuropa_session` and stores an encrypted JSON web token (JWT) provided by the IUROPA API. To access data from the API, we need to provide this token. We can do this simply by passing the `session` object to any function in the `iuropa` package that has a `session` argument.

The token verifies to the API that you have provided a valid username and password. It also tells the API how long your session is valid for. A session automatically times out after 30 minutes. If you get an error from a `iuropa` function that says that an API query was not successful, it's most likely because your session has timed out.

You can check that your session is valid using `check_authentication()`.

```
check_authentication(session)
```

```
## Checking authentication with the IUROPA API...
## You are currently authenticated.
```

You can have multiple active sessions at the same time, even in the same `R` environment, as long as you use different names for the `iuropa_session` objects. The sessions can be for the same user or different users.

2.2 Looking up IUROPA components

Many functions in the `iuropa` package have a `component` argument that requires you to provide the name of a IUROPA component. As scholars add data to the IUROPA CJEU Database, more components will become available.

We can use the `list_components()` function to see a list of the components that are currently available via the IUROPA API. These are the values we can pass to the `component` argument. The function requests information from the IUROPA API and returns a string vector of component names. It has a `session` argument so we'll need to pass in the `iuropa_session` object we created using `authenticate()`.

```
list_components(session)
```

```
## Connecting to the IUROPA API...
## Response received.
```

```
## [1] "cjeu_database_platform" "cjeu_text_corpus"
```

In this example, we're interested in getting data from the CJEU Database Platform, so we'll use `cjeu_database_platform` whenever a function asks us to specify a `component`.

If we aren't sure which component we need, we can also use the function `describe_components()` to get more information about each component.

```
describe_components(session)
```

```
## Connecting to the IUROPA API...
## Response received.
```

```
## # A tibble: 2 x 5
##   key_id component_id component              label                  description
##    <int>        <int> <chr>                  <chr>                  <chr>
## 1      1            1 cjeu_database_platform CJEU Database Platform The CJEU Da~
## 2      2            2 cjeu_text_corpus       CJEU Text Corpus       The CJEU Te~
```

2.3 Looking up datasets

Next, we need to pick the dataset in the `cjeu_database_platform` that has data on the assignment of judges to proceedings. To see the datasets that are currently available in the `cjeu_database_platform`, we can use the `list_datasets()` function. We'll pass in our `iuropa_session` object to the `session` argument again The function takes one other argument, `component`, which needs to be one of the values returned by `list_component()`.

```
list_datasets(
  session = session,
  component = "cjeu_database_platform"
)
```

```
## Connecting to the IUROPA API...
## Response received.
```

```
## [1] "cases"       "parties"     "decisions"   "procedures"  "assignments"
## [6] "submissions" "citations"   "judges"
```

Like `list_components()`, the `list_datasets()` requests information via the IUROPA API and returns a string vector of component names. We're going to want the `assignments` dataset.

If we don't already know that the data we're looking for is in the `assignments` database, we could also use the `describe_datasets()` function to get more information about each dataset. This function returns a `tibble` (see the `tidyverse`).

```
describe_datasets(
  session = session,
  component = "cjeu_database_platform"
)
```

```
## Connecting to the IUROPA API...
## Response received.
```

```
## # A tibble: 8 x 5
##   key_id dataset_id dataset     label              description
##    <int>      <int> <chr>       <chr>              <chr>
## 1      1          1 cases       CJEU cases         This dataset includes inform~
## 2      2          2 parties     Parties in CJEU c~ This dataset includes inform~
## 3      3          3 decisions   CJEU decisions     This dataset includes inform~
## 4      4          4 procedures  Legal procedures ~ This dataset includes inform~
## 5      5          5 assignments Chamber compositi~ This dataset includes inform~
## 6      6          6 submissions Observations and ~ This dataset includes inform~
## 7      7          7 citations   Citations in CJEU~ This dataset includes inform~
## 8      8          8 judges      CJEU judges and A~ This dataset includes inform~
```

To see the whole list, we can assign the output to an object and view in RStudio.

2.4 Checking the codebook

To double-check that the `assignments` dataset has the information we're looking for, we can get a list of variables using `list_variables()`. In addition to `session` and `component` arguments, this function has a `dataset` argument, which needs to be one of the values returned by `list_datasets()`.

```
list_variables(
  session = session,
  component = "cjeu_database_platform",
  dataset = "assignments"
)
```

```
## Connecting to the IUROPA API...
## Response received.
```

```
##  [1] "key_id"               "court_id"             "court"
##  [4] "iuropa_proceeding_id" "proceeding"           "proceeding_year"
##  [7] "proceeding_number"    "proceeding_date"      "iuropa_decision_id"
## [10] "ecli"                 "celex"                "decision_date"
## [13] "decision_type_id"     "decision_type"        "judge_id"
## [16] "last_name"            "president"            "rapporteur"
## [19] "acting"
```

Some of these variables are self-explanatory if we're familiar with CJEU data. If we aren't sure what a variable is, we can use `describe_variables()` to see the full codebook for the dataset as a `tibble`.

```
describe_variables(
  session = session,
  component = "cjeu_database_platform",
  dataset = "assignments"
)
```

```
## Connecting to the IUROPA API...
## Response received.
```

```
## # A tibble: 19 x 7
##    key_id dataset_id dataset     variable_id variable             type    description
##     <int>      <int> <chr>             <int> <chr>                <chr>   <chr>
##  1     97          5 assignments           1 key_id               numeric "An ID num~
##  2     98          5 assignments           2 court_id             numeric "An ID num~
##  3     99          5 assignments           3 court                string  "The name ~
##  4    100          5 assignments           4 iuropa_proceeding_id string  "An ID num~
##  5    101          5 assignments           5 proceeding           string  "The proce~
##  6    102          5 assignments           6 proceeding_year      numeric "The year ~
##  7    103          5 assignments           7 proceeding_number    numeric "The numbe~
##  8    104          5 assignments           8 proceeding_date      date    "The date ~
##  9    105          5 assignments           9 iuropa_decision_id   string  "An ID num~
## 10    106          5 assignments          10 ecli                 string  "The ECLI ~
## 11    107          5 assignments          11 celex                string  "The CELEX~
## 12    108          5 assignments          12 decision_date        date    "The date ~
## 13    109          5 assignments          13 decision_type_id     numeric "An ID num~
## 14    110          5 assignments          14 decision_type        string  "The type ~
## 15    111          5 assignments          15 judge_id             numeric "An ID num~
## 16    112          5 assignments          16 last_name            string  "The last ~
## 17    113          5 assignments          17 president            dummy   "A dummy v~
## 18    114          5 assignments          18 rapporteur           dummy   "A dummy v~
## 19    115          5 assignments          19 acting               dummy   "A dummy v~
```

2.5 Selecting data

Now we're ready to select the data we're interested in. We want the judges in the chamber for Court of Justice judgments since 2004, so we don't need to download the entire `assignments` dataset, which also includes data on decisions by the General Court and the Civil Service Tribunal. Instead of downloading the entire dataset, we can filter the data using the API and download just what we need.

We can use the `download_data()` function to download the data. This function has three required arguments, `session`, `component`, and `dataset`, and two optional arguments, `filters` and `variables`.

We can use the `filters` argument to select a subset of observations. This argument takes a `list`. Each element in the list specifies a filter to apply to the data. The name of each element should be the name of a

variable in the specified dataset and the corresponding value should be a value or vector of values that the variable can take. The results will only include observations where the variable equals one of the provided values.

If we aren't sure what values a variable can take, we can look up the variable in the codebook using `describe_variables()`. The codebook indicates the type of the variable (`numeric`, `string`, etc.). For categorical variables with a limited number of categories, it also indicates the values that the variable can take. For numeric variables, we can add `_min` or `_max` to the end of the variable name to specify a minimum or maximum value.

If we specify multiple filters, the results will only include observations that match all of the filters.

To get a list of judges that participated in Court of Justice judgments since 2004, we need to specify three filters, one for `court`, one for `decision_type`, and one for the `decision_date`. Since we want data since the 2004 enlargement, we need to add `_min` to the end of the `decision_date` variable to set a lower bound on the date.

```
list(
  court = "Court of Justice",
  decision_type = "Judgment",
  decision_date_min = "2004-05-01"
)
```

We can use the `variables` argument to select a subset of variables. This argument takes a string vector where each element of the vector is a valid variable name. We can use `list_variables()` to double-check the variable names. In this example, we'll select three variables: `ecli`, `judge_id`, and `last_name`. Looking at the codebook, `ecli` is a unique identifier for each judgment, `judge_id` is a unique identifier for each judge, and `last_name` is the last name of the judge.

2.6 Downloading data

Now that we know how to select observations and variables, we can use `download_data()` to download the data we're interested in from the CJEU Database Platform via the IUROPA API.

```
out <- download_data(
  session = session,
  component = "cjeu_database_platform",
  dataset = "assignments",
  filters = list(
    court = "Court of Justice",
    decision_type = "Judgment",
    decision_date_min = "2004-05-01"
  ),
  variables = c("ecli", "judge_id", "last_name")
)
```

```
## Requesting data via the IUROPA API...
## Response received.
## Observations requested: 37728.
## Downloading 10000 observations every 5 seconds.
## Total estimated time: 0.25 minutes (15 seconds).
## Batch 1 of 4 complete (observations 1 to 10000 of 37728).
## Batch 2 of 4 complete (observations 10001 to 20000 of 37728).
## Batch 3 of 4 complete (observations 20001 to 30000 of 37728).
## Batch 4 of 4 complete (observations 30001 to 37728 of 37728).
## Your download is complete!
##
## If you use data from the CJEU Database Platform in a paper or project, please cite
```

```
the database:
##
## Brekke, Stein Arne, Joshua Fjelstul, Silje Synnøve Lyder Hermansen and Daniel
Naurin. 2021 "The CJEU Database Platform: Decisions and Decision-Makers (Stable
Release 0.1)", in Lindholm, Johan, Daniel Naurin, Urska Sadl, Anna Wallerman Ghavanini,
Stein Arne Brekke, Joshua Fjelstul, Silje Synnøve Lyder Hermansen, Olof Larsson, Andreas
Moberg, Moa Näsström, Michal Ovádek, Tommaso Pavone, and Philipp Schroeder, The Court of
Justice of the European Union Database, IUROPA, URL: http://iuropa.pol.gu.se/.
##
## Please also cite the paper that introduces the CJEU Database Platform:
##
## Stein Arne Brekke, Joshua C. Fjelstul, Silje Synnøve Lyder Hermansen, and Daniel
Naurin. The CJEU Database Platform: Decisions and Decision-Makers. Working paper.
##
## Please also cite the iuropa R package:
##
## Joshua C. Fjelstul (2021). iuropa: An R Interface to the IUROPA CJEU Database. R
package version 0.1.0. https://github.com/jfjelstul/iuropa.
```

The `download_data()` function downloads the data in batches of `10000` observations. The IUROPA API has a rate limit, but this function automatically manages the rate limit for us. It will download `1` batch approximately every `5` seconds.

The function prints useful information to the `console` while the data downloads. It tells us how many observations we have requested, how many batches it will take to download the data, and approximately how long it will take. It provides an update every time a batch is downloaded and counts down to the next batch. The function returns a `tibble` that we can manipulate with `dplyr` and `tidyr`.