Package ‘CatMapR’

November 14, 2024

Type Package

Title This package interfaces with the CatMapper application API (catmapper.org)

Version 0.1.0

Author Robert J. Bischoff

Maintainer Robert J. Bischoff <bischrob@gmail.com>

Description CatMapper organizes dynamic and complex category systems commonly used by scientists and policymakers, including ethnicities, languages, religions, political districts, political parties, and technologies. Each of these systems includes thousands of categories encoded in diverse, dynamic and incompatible ways across a growing corpus of thousands of datasets.
CatMapper assists users in (1) exploring key contextual information about categories of interest (e.g., Aymara ethnicity, Balochi language, Rajshahi district), (2) identifying which of thousands of datasets contain information about specific categories, and (3) reconciling distinct and incompatible encodings for the same category across diverse datasets to enable novel analyses.
CatMapper currently houses two apps—SocioMap and ArchaMap.
SocioMap organizes the thousands of sociopolitical categories—e.g., ethnicities, languages, religions, districts, and political parties—frequently used by social scientists and policymakers. Users can search for basic contextual information on each category—geographical location, population size, alternative names, and language—as well as the datasets containing specific social, demographic, cultural and economic data for each category. In the future, SocioMap will also provide tools for facilitating and sharing merges of diverse and heterogenous external datasets by these category systems to enable novel analyses.
ArchaMap will organize artifact types—e.g., ceramics, lithics—frequently used by archaeologists in analyses of material culture. ArchaMap will share SocioMap’s functionalities for merging data from multiple sources by diverse category systems.

License GPL

Encoding UTF-8

LazyData true

Suggests roxygen2

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Depends httr,
jsonlite,
tictoc,
xml2

Contents

`allDatasets` *Retrieve All Datasets from a Specified Database*

Description

This function retrieves detailed information about all datasets from a specified database. It returns information such as dataset identifiers, names, applicable years, project details, and other relevant metadata.

Usage

```
allDatasets(database)
```

Arguments

`database`	A string specifying the database from which to retrieve datasets. Valid options are "SocioMap" or "ArchaMap".

Value

A list containing detailed information about each dataset, or an error message if the request fails. The list includes fields such as nodeID, CMName, CMID, shortName, project, Unit, parent, ApplicableYears, and more.

Examples

```
allDatasets(database = "SocioMap")
allDatasets(database = "ArchaMap")
```

callAPI *Call API*

Description

This is a helper function to call the API. It takes the endpoint, parameters, request type, and URL as input and returns the API response.

Usage

```
callAPI(
  endpoint,
  parameters,
  request = "GET",
  url = "https://catmapper.org/api",
  type = "default"
)
```

Arguments

`endpoint`	API endpoint
`parameters`	API parameters
`request`	GET or POST
`url`	API URL
`type`	default or stream

Value

API response

Examples

```
CatMapR:::callAPI(endpoint = "search", parameters = list(term = "Dan", database = "SocioMap", property = "Name"
```

CMIDinfo *Retrieve Details for a Specific CMID (CatMapperID)*

Description

This function retrieves detailed information about a specified CatMapperID (CMID) from a specified database. It fetches both node properties and their relationships associated with the given CMID.

Usage

```
CMIDinfo(database, cmid)
```

Arguments

`database`	A string specifying the database to search in. Valid options are "SocioMap" or "ArchaMap".
`cmid`	The CatMapperID for which details are to be retrieved (e.g., "SM1" or "AM1").

Value

A list containing node properties and relationships associated with the specified CMID, or an error message if the request fails.

Examples

```
CMIDinfo(database = "SocioMap", cmid = "SM1")
CMIDinfo(database = "ArchaMap", cmid = "AM1")
```

`createLinkfile` *Create Linkfile from Dataset CMIDs*

Description

This function propose a merge based on a category domain and selected datasets. It then processes the response and returns a list of dictionaries containing proposed merge information.

Usage

```
createLinkfile(
  categoryLabel,
  datasetChoices,
  database = "SocioMap",
  intersection = FALSE
)
```

Arguments

`categoryLabel`	Character vector specifying the category domain for the merge.
`datasetChoices`	Character vector of CMIDs representing the selected datasets.
`database`	Character string specifying the database to use. (default: "SocioMap" – "ArchaMap" is the other option)
`intersection`	Boolean value specifying whether to return the intersection of the datasets or all categories. (default: FALSE)

Value

Dataframe containing proposed merge information.

- datasetID: Character string representing the dataset ID.
- Key: Character string representing a unique identifier for the merge.
- CMName: Character string representing the category name.
- CMID: Character string representing the category ID.
- Name: Character string representing a semicolon-separated list of dataset names.

Examples

```
categoryLabel <- c("ETHNICITY")
datasetChoices <- c("SD5", "SD6")
merged_data <- createLinkfile(categoryLabel, datasetChoices)
```

`datasetInfo` *Retrieve Dataset Information by CMID*

Description

This function retrieves detailed information about a dataset based on a given CMID (CatMapperID) from a specified database, with an optional domain filter. It fetches relationships and properties of datasets associated with the specified CMID.

Usage

```
datasetInfo(database, CMID, domain = "CATEGORY")
```

Arguments

`database` A string specifying the database to search in. Valid options are "SocioMap" or "ArchaMap".

`CMID` The CMID of the dataset to retrieve information for (e.g., "SD1" or "AD1").

`domain` (Optional) A category to filter dataset relationships. Defaults to "CATEGORY" if not specified.

Value

A list containing detailed information about the dataset, or an error message if the request fails.

Examples

```
datasetInfo(database = "SocioMap", CMID = "SD1", domain = "CATEGORY")
datasetInfo(database = "ArchaMap", CMID = "AD1")
```

`joinDatasets` *Join Datasets by Key*

Description

This function calls the CatMapper API to join two datasets based on specified parameters. It retrieves translated keys from the database and returns the joined data with the CMID and CMName. The function requires a column named ’datasetID’ in both datasets that has the CMID of the translated dataset. It also requires the columns used to create the ’Key’ used in translation. For example, if the Key for the GADM dataset ’SD1’ is ’GID: AFG’, then the ’Key’ column should be ’GID’ in the dataset.

Usage

```
joinDatasets(database, joinLeft, joinRight)
```

Arguments

database	A string specifying the database to use, either "SocioMap" or "ArchaMap".
joinLeft	A data frame representing the left dataset with a "datasetID" column and other relevant columns.
joinRight	A data frame representing the right dataset with a "datasetID" column and other relevant columns.

Value

A data frame containing the joined datasets or an error message if the request fails.

Examples

```
joinLeft = data.frame(datasetID = "SD1", country = "Afghanistan", GID = "AFG", val0 = 1)
joinRight = data.frame(datasetID = "SD2", country = "Afghanistan", geonameid = "1149361", val1 = 2)
joinDatasets("SocioMap", joinLeft, joinRight)
```

---

searchDatabase *Search*

Description

Search for a term in a database and retrieve the CMID (CatMapperID), CMName (CatMapper Name), and other information about the matches.

Usage

```
searchDatabase(
  database,
  domain = NULL,
  term = NULL,
  property = "Name",
  yearStart = NULL,
  yearEnd = NULL,
  country = NULL,
  context = NULL,
  limit = 1000
)
```

Arguments

database	name of database (SocioMap or ArchaMap)
domain	name of category domain to search for (DISTRICT, ETHNICITY, etc.)
term	search term ("Afghanistan")
property	property to search by (Name, CMID, or Key)
yearStart	year to search by start

yearEnd	year to search by end
country	country to search by (must be a CMID)
context	context to search by (e.g., hierarchical category as in the state for a county–must be a CMID)
limit	limit number of results

Value

dataframe

Examples

```
searchDatabase(database = "SocioMap",domain = "ETHNICITY", term = "Dan",property = "Name", yearStart = NULL, ye
```

---

translate *Translate*

Description

This function translate a dataframe by matching the specified term with the property in the database.

Usage

```
translate(
  rows,
  database,
  term,
  property = "Name",
  domain = "CATEGORY",
  context = NULL,
  country = NULL,
  dataset = NULL,
  yearStart = NULL,
  yearEnd = NULL,
  key = "false",
  query = "false"
)
```

Arguments

database	database to use (SocioMap or ArchaMap)
term	column name to translate
property	property to search by (Name, CMID, or Key)
domain	name of category domain to search for (DISTRICT, ETHNICITY, etc.)
context	column name of context to search by (CMID)
country	column name of country to search by (CMID)
yearStart	year to search by start
yearEnd	year to search by end
query	return query instead of results

Value

dataframe

Examples

```
df = data.frame(country = "Afghanistan")
translate(rows = df, database = "SocioMap",domain = "ADM0", term = "country", property = "Name", yearStart = NUL
```

Index