Package ‘nzilbb.labbcat’

January 16, 2020

Version 0.4-2
Date 2020-01-16
Title Accessing Data Stored in 'LaBB-CAT' Instances
Imports jsonlite, httr, stringr, utils, rstudioapi
Description 'LaBB-CAT' is a web-based language corpus management system developed by the New Zealand Institute of Language, Brain and Behaviour (NZILBB) - see <https://labbcat.canterbury.ac.nz>. This package defines functions for accessing corpus data in a 'LaBB-CAT' instance. You must have at least version 20200108.1025 of 'LaBB-CAT' to use this package.
For more information about 'LaBB-CAT', see
or
License GPL (>= 3)
Copyright New Zealand Institute of Language, Brain and Behaviour, University of Canterbury
URL https://github.com/nzilbb/labbcat-R,
https://labbcat.canterbury.ac.nz
RoxygenNote 7.0.2
Suggests testthat (>= 2.1.0)
NeedsCompilation no
Author Robert Fromont [aut, cre]
Maintainer Robert Fromont <robert.fromont@canterbury.ac.nz>
Repository CRAN
Date/Publication 2020-01-16 17:40:07 UTC

R topics documented:
countAnnotations .......................... 2
countAnnotations

*Gets the number of annotations on the given layer of the given graph.*

**Description**

Returns the number of annotations on the given layer of the given graph (transcript).

**Usage**

```python
countAnnotations(labbcat.url, id, layerId)
```

**Arguments**

- `labbcat.url`: URL to the LaBB-CAT instance
- `id`: A graph ID (i.e. transcript name)
- `layerId`: A layer name

**Value**

The number of annotations on that layer
getAnchors

See Also

getGraphIds getGraphIdsInCorpus getGraphIdsWithParticipant

Examples

```r
## Not run:
## define the LaBB-CAT URL
labbcat.url <- "https://labbcat.canterbury.ac.nz/demo/"

## Count the number of words in UC427_ViktoriaPapp_A_ENG.eaf
token.count <- countAnnotations(labbcat.url, "UC427_ViktoriaPapp_A_ENG.eaf", "orthography")

## End(Not run)
```

getAnchors

*Gets the given anchors in the given graph.*

**Description**

Lists the given anchors in the given graph (transcript).

**Usage**

getAnchors(labbcat.url, id, anchorId)

**Arguments**

- **labbcat.url**: URL to the LaBB-CAT instance
- **id**: A graph ID (i.e. transcript name)
- **anchorId**: A vector of anchor IDs (or a string representing one anchor ID)

**Value**

A named list of anchors, with members:

- **id** The annotation’s unique ID,
- **offset** The offset from the beginning (in seconds if it’s a transcript of a recording, or in characters if it’s a text document)
- **confidence** A rating from 0-100 of the confidence of the offset, e.g. 10: default value, 50: force-aligned, 100: manually aligned

**See Also**

getAnnotations
getAnnotationLabels

Gets labels of annotations on a given layer, identified by given annotation IDs.

Usage

getAnnotationLabels(labbcat.url, id, layerId, count = 1, no.progress = FALSE)

Arguments

- **labbcat.url**: URL to the LaBB-CAT instance
- **id**: A vector of annotation IDs.
- **layerId**: A layer name.
- **count**: The number of annotations on the given layer to retrieve.
- **no.progress**: Optionally suppress the progress bar when multiple fragments are specified - TRUE for no progress bar.

Value

A data frame of labels.

Examples

```r
## Not run:
## define the LaBB-CAT URL
labbcat.url <- "https://labbcat.canterbury.ac.nz/demo/"

## Get the first 20 orthography tokens in UC427_ViktoriaPapp_A_ENG.eaf
orthography <- getAnnotations(labbcat.url, "UC427_ViktoriaPapp_A_ENG.eaf", "orthography", 20, 0)

## Get the start anchors for the above tokens
word.starts <- getAnchors(labbcat.url, "UC427_ViktoriaPapp_A_ENG.eaf", orthography$startId)

## End(Not run)
```
## Get the topic annotations for the matches

```r
topics <- getAnnotationLabels(labbcat.url, results$MatchId, "topic")
```

## End(Not run)

---

### getAnnotations

**getAnnotations**  
*Gets the annotations on the given layer of the given graph.*

**Description**

Returns the annotations on the given layer of the given graph (transcript).

**Usage**

```r
getAnnotations(labbcat.url, id, layerId, pageLength = NULL, pageNumber = NULL)
```

**Arguments**

- **labbcat.url**  
  URL to the LaBB-CAT instance
- **id**  
  A graph ID (i.e. transcript name)
- **layerId**  
  A layer name
- **pageLength**  
  The maximum number of annotations to return, or null to return all
- **pageNumber**  
  The zero-based page number to return, or null to return the first page

**Value**

A named list of annotations, with members:

- **id** The annotation’s unique ID
- **layerId** The name of the layer it comes from
- **label** The value of the annotation
- **startId** The ID of the start anchor,
- **endId** The ID of the end anchor,
- **parentId** The ID of the parent annotation,
- **ordinal** The ordinal of the annotation among its peers,
- **confidence** A rating from 0-100 of the confidence of the label e.g. 10: default value, 50: automatically generated, 100: manually annotated

**See Also**

- `getGraphIds`
- `getGraphIdsInCorpus`
- `getGraphIdsWithParticipant`
- `countAnnotations`
Examples

```r
## Not run:
## define the LaBB-CAT URL
labbcat.url <- "https://labbcat.canterbury.ac.nz/demo/"

## Get all the orthography tokens in UC427_ViktoriaPapp_A_ENG.eaf
orthography <- getAnnotations(labbcat.url, "UC427_ViktoriaPapp_A_ENG.eaf", "orthography")

## Get the first 20 orthography tokens in UC427_ViktoriaPapp_A_ENG.eaf
orthography <- getAnnotations(labbcat.url, "UC427_ViktoriaPapp_A_ENG.eaf", "orthography", 20, 0)

## End(Not run)
```

getAvailableMedia

List the media available for the given graph.

Description

List the media available for the given graph.

Usage

`getAvailableMedia(labbcat.url, id)`

Arguments

- `labbcat.url`: URL to the LaBB-CAT instance
- `id`: A graph ID (i.e. transcript name)

Value

A named list of media files available for the given graph, with members:

- `trackSuffix`: The track suffix of the media
- `mimeType`: The MIME type of the file
- `url`: URL to the content of the file
- `name`: Name of the file

See Also

`getGraphIds`
getCorpusIds

Examples

## Not run:
## define the LaBB-CAT URL
labbcat.url <- "https://labbcat.canterbury.ac.nz/demo/"

## List the media files available for BR2044_OllyOhlson.eaf
media <- getAvailableMedia(labbcat.url, "BR2044_OllyOhlson.eaf")

## End(Not run)

---

getCorpusIds  
*Gets a list of corpus IDs.*

Description

Returns a list of corpora in the given 'LaBB-CAT' instance.

Usage

gETCHelpCorpusIds(labbcat.url)

Arguments

labbcat.url  
URL to the LaBB-CAT instance

Value

A list of corpus IDs

Examples

## Not run:
## List corpora
corpora <- getCorpusIds("https://labbcat.canterbury.ac.nz/demo/"

## End(Not run)
getDictionaries

List the dictionaries available.

Description

List the dictionaries available.

Usage

getDictionaries(labbcat.url)

Arguments

labbcat.url URL to the LaBB-CAT instance

Value

A named list of layer manager IDs, each of which containing a list of dictionaries that the layer manager makes available.

See Also

getDictionaryEntries

Examples

## Not run:
## List the dictionaries available
dictionaries <- getDictionaries("https://labbcat.canterbury.ac.nz/demo/")

## End(Not run)

getDictionaryEntries

Lookup entries in a dictionary.

Description

Lookup entries in a dictionary.

Usage

getDictionaryEntries(labbcat.url, managerId, dictionaryId, keys)
Arguments

- `labbcat.url` URL to the LaBB-CAT instance
- `managerId` The layer manager ID of the dictionary, as returned by `getDictionaries`
- `dictionaryId` The ID of the dictionary, as returned by `getDictionaries`
- `keys` A list of entries to look up

Value

A data frame with the keys and their dictionary entries.

See Also

`getDictionaries`

Examples

```r
## Not run:
## define the LaBB-CAT URL
labbcat.url <- "https://labbcat.canterbury.ac.nz/demo/"
keys <- c("the", "quick", "brown", "fox")

## get the pronunciations according to CELEX
entries <- getDictionaryEntries(labbcat.url, "CELEX-EN", "Phonology (wordform)", keys)

## End(Not run)
```
getFragments

Arguments

- `labbcat.url` URL to the LaBB-CAT instance
- `id` The graph ID (transcript name) of the sound recording, or a vector of graph IDs.
- `start` The start time in seconds, or a vector of start times.
- `end` The end time in seconds, or a vector of end times.
- `layerId` A vector of layer IDs.
- `mimeType` Optional content-type - currently only "text/praat-textgrid" is supported.
- `no.progress` Optionally suppress the progress bar when multiple fragments are specified - TRUE for no progress bar.
- `path` Optional path to directory where the files should be saved.

Value

The name of the file, which is saved in the current directory, or a list of names of files, if multiple id's/start's/end's were specified.

If a list of files is returned, they are in the order that they were returned by the server, which *should* be the order that they were specified in the id/start/end lists.

Examples

```r
## Not run:
## define the LaBB-CAT URL
labbcat.url <- "https://labbcat.canterbury.ac.nz/demo/"

## Get the 5 seconds starting from 10s after the beginning of a recording
textgrid.file <- getFragments(labbcat.url, "AP2505_Nelson.eaf", 10.0, 15.0,
c("transcript", "phonemes"), path="samples")

## Load some search results previously exported from LaBB-CAT
results <- read.csv("results.csv", header=T)

## Get a list of fragment TextGrids, including the utterances, transcript, and phonemes layers
textgrid.files <- getFragments(
  labbcat.url, results$Transcript, results$Line, results$LineEnd,
c("utterances", "transcript", "phonemes"))

## Get a list of fragment TextGrids with no progress bar
textgrid.files <- getFragments(
  labbcat.url, results$Transcript, results$Line, results$LineEnd, no.progress=TRUE)

## End(Not run)
```
getGraphIds

Summary

getGraphIds

Description

Returns a list of graph IDs (i.e. transcript names).

Usage

getGraphIds(labbcat.url)

Arguments

labbcat.url URL to the LaBB-CAT instance

Value

A list of graph IDs

Examples

## Not run:
## List all transcripts
transcripts <- getGraphIds("https://labbcat.canterbury.ac.nz/demo/")

## End(Not run)

getGraphIdsInCorpus

Summary

getGraphIdsInCorpus

Description

Returns a list of corpora in the given 'LaBB-CAT' instance.

Usage

getGraphIdsInCorpus(labbcat.url, id)

Arguments

labbcat.url URL to the LaBB-CAT instance
id The ID (name) of the corpus

Value

A list of corpus IDs
getGraphIdsWithParticipant

Gets a list of IDs of graphs that include the given participant.

Description

Returns a list of IDs of graphs (i.e. transcript names) that include the given participant.

Usage

getGraphIdsWithParticipant(labbcat.url, id)

Arguments

labbcat.url URL to the LaBB-CAT instance
id A participant ID

Value

A list of graph IDs

See Also

getParticipantIds

Examples

## Not run:
## define the LaBB-CAT URL
labbcat.url <- "https://labbcat.canterbury.ac.nz/demo/"

## List transcripts in which UC427_ViktoriaPapp_A_ENG speaks
transcripts <- getGraphIdsWithParticipant(labbcat.url, "UC427_ViktoriaPapp_A_ENG")

## End(Not run)
getId

Gets the store’s ID.

Description
The store’s ID - i.e. the ID of the 'LaBB-CAT' instance.

Usage

getId(labbcat.url)

Arguments

labbcat.url     URL to the LaBB-CAT instance

Value
The annotation store’s ID

Examples

## Not run:
## Get ID of LaBB-CAT instance
instance.id <- getId("https://labbcat.canterbury.ac.nz/demo/")

## End(Not run)

getLayer

Gets a layer definition.

Description

Gets a layer definition.

Usage

getLayer(labbcat.url, id)

Arguments

labbcat.url     URL to the LaBB-CAT instance
id              ID of the layer to get the definition for
Value

The definition of the given layer, with members:

- **id** The layer’s unique ID
- **parentId** The layer’s parent layer ID
- **description** The description of the layer
- **alignment** The layer’s alignment - 0 for none, 1 for point alignment, 2 for interval alignment
- **peers** Whether children have peers or not
- **peersOverlap** Whether child peers can overlap or not
- **parentIncludes** Whether the parent t-includes the child
- **saturated** Whether children must temporally fill the entire parent duration (true) or not (false)
- **parentIncludes** Whether the parent t-includes the child
- **type** The type for labels on this layer
- **validLabels** List of valid label values for this layer

See Also

getLayerIds, getLayers

Examples

```r
# Not run:
# Get the definition of the orthography layer
orthography.layer <- getLayer("https://labbccat.canterbury.ac.nz/demo/", "orthography")

# End(Not run)
```

getLayerIds

*Gets a list of layer IDs.*

Description

Layer IDs are annotation 'types'.

Usage

getLayerIds(labbcat.url)

Arguments

- labbcat.url URL to the LaBB-CAT instance

Value

A list of layer IDs
getLayers

Examples

```r
## Not run:
## Get names of all layers
layer.ids <- getLayerIds("https://labbcat.canterbury.ac.nz/demo/"
## End(Not run)
```

getLayers

*Gets a list of layer definitions.*

Description

Gets a list of layer definitions.

Usage

```r
getLayers(labbcat.url)
```

Arguments

- `labbcat.url` URL to the LaBB-CAT instance

Value

A list of layer definitions, with members:

- `id` The layer’s unique ID
- `parentId` The layer’s parent layer ID
- `description` The description of the layer
- `alignment` The layer’s alignment - 0 for none, 1 for point alignment, 2 for interval alignment
- `peers` Whether children have peers or not
- `peersOverlap` Whether child peers can overlap or not
- `parentIncludes` Whether the parent t-includes the child
- `saturated` Whether children must temporally fill the entire parent duration (true) or not (false)
- `parentIncludes` Whether the parent t-includes the child
- `type` The type for labels on this layer
- `validLabels` List of valid label values for this layer

See Also

- `getLayerIds`
getMatches

getMatches

Search for tokens.

Description
Searches through transcripts for tokens matching the given pattern.

Usage
getMatches(
  labbcat.url,
  pattern,
  participantId = NULL,
  mainparticipant = TRUE,
  words.context = 0,
  no.progress = FALSE
)

Arguments

labbcat.url URL to the LaBB-CAT instance

pattern An object representing the pattern to search for.

Strictly speaking, this should be a named list that replicates the structure of the
’search matrix’ in the LaBB-CAT browser interface, with one element called
“columns”, containing a named list for each column.

Each element in the “columns” named list contains an element named “layers”,
whose value is a named list for patterns to match on each layer, and optionally
an element named “adj”, whose value is a number representing the maximum
distance, in tokens, between this column and the next column - if “adj” is not
specified, the value defaults to 1, so tokens are contiguous.

Each element in the “layers” named list is named after the layer it matches, and
the value is a named list with the following possible elements:

• pattern A regular expression to match against the label

• min An inclusive minimum numeric value for the label

• max An exclusive maximum numeric value for the label

• not TRUE to negate the match
getMatches

- **anchorStart** TRUE to anchor to the start of the annotation on this layer (i.e. the matching word token will be the first at/after the start of the matching annotation on this layer)
- **anchorEnd** TRUE to anchor to the end of the annotation on this layer (i.e. the matching word token will be the last before/at the end of the matching annotation on this layer)
- **target** TRUE to make this layer the target of the search; the results will contain one row for each match on the target layer

Examples of valid pattern objects include:

```r
## words starting with 'ps...' pattern <- list(columns = list(
  list(layers = list(
    orthography = list(pattern = "ps.*"))))
```

```r
## the word 'the' followed immediately or with one intervening word by
## a hapax legomenon (word with a frequency of 1) that doesn't start with a vowel
pattern <- list(columns = list(
  list(layers = list(
    orthography = list(pattern = "the")),
    adj = 2),
  list(layers = list(
    phonemes = list(not = TRUE, pattern = "[cCEFHiIPqQuUV0123456789~#$@].*"),
    frequency = list(max = "2")))))
```

For ease of use, the function will also accept the following abbreviated forms:

```r
## a single list representing a 'one column' search,
## and string values, representing regular expression pattern matching
pattern <- list(orthography = "ps.*")
```

```r
## a list containing the columns (adj defaults to 1, so matching tokens are contiguous).
pattern <- list(
  list(orthography = "the"),
  list(phonemes = list(not = TRUE, pattern = "[cCEFHiIPqQuUV0123456789~#$@].*"),
    frequency = list(max = "2")))
```

**participantId** An optional list of participant IDs to search the utterances of. If not supplied, all utterances in the corpus will be searched.

**main.participant** TRUE to search only main-participant utterances, FALSE to search all utterances.

**words.context** Number of words context to include in the ‘Before.Match’ and ‘After.Match’ columns in the results.

**no.progress** Optionally suppress the progress bar when multiple fragments are specified - TRUE for no progress bar.

**Value**

A data frame identifying matches, containing the following columns:
• **SearchName** A name based on the pattern – the same for all rows
• **Number** Row number
• **Transcript** Name of the transcript in which the match was found
• **Line** The start offset of the utterance/line
• **LineEnd** The end offset of the utterance/line
• **MatchId** A unique ID for the matching target token
• **Before.Match** Transcript text immediately before the match
• **Text** Transcript text of the match
• **Before.Match** Transcript text immediately after the match
• **Target.transcript** Text of the target word token
• **Target.transcript.start** Start offset of the target word token
• **Target.transcript.end** End offset of the target word token
• **Target.segments** Label of the target segment (only present if the segment layer is included in the pattern)
• **Target.segments.start** Start offset of the target segment (only present if the segment layer is included in the pattern)
• **Target.segments.end** End offset of the target segment (only present if the segment layer is included in the pattern)

**See Also**

getParticipantIds

**Examples**

```r
## Not run:
## define the LaBB-CAT URL
labbcat.url <- "https://labbcat.canterbury.ac.nz/demo/"

## create a pattern object to match against
pattern <- list(columns = list(
  list(layers = list(
    orthography = list(pattern = "the"),
    adj = 2),
  list(layers = list(
    phonemes = list(not=TRUE, pattern = "[cCEFHiIPqQuUV0123456789~#\$@].*"),
    frequency = list(max = "2")))))

## get the tokens matching the pattern
results <- getMatches(labbcat.url, pattern)

## results$MatchId can be used to access results

## End(Not run)
```
getMatchingGraphIds

Description

Gets a list of IDs of graphs (i.e. transcript names) that match a particular pattern.

Usage

getMatchingGraphIds(
    labbcat.url,
    expression,
    pageLength = NULL,
    pageNumber = NULL,
    order = NULL
)

Arguments

labbcat.url URL to the LaBB-CAT instance
expression An expression that determines which graphs match
pageLength The maximum number of IDs to return, or null to return all
pageNumber The zero-based page number to return, or null to return the first page
order An expression that determines the order the graphs are listed in - if specified, this must include the keyword 'ASC' for ascending or 'DESC' for descending order.

Details

The results can be exhaustive, by omitting pageLength and pageNumber, or they can be a subset (a 'page') of results, by given pageLength and pageNumber values.

The order of the list can be specified. If omitted, the graphs are listed in ID order.

The expression language is currently not well defined, but expressions such as those in the examples can be used.

Value

A list of graph IDs (i.e. transcript names)

Examples

## Not run:
## define the LaBB-CAT URL
labbcat.url <- "https://labbcat.canterbury.ac.nz/demo/"

## Get all transcripts whose names start with "BR"
transcripts <- getMatchingGraphIds(labbcat.url, "id MATCHES 'BR.+'")

## Get the first twenty transcripts in the "QB" corpus
transcripts <- getMatchingGraphIds(labbcat.url, "my('corpus').label = 'QB'", 20, 0)

## Get the second transcript that has "QB247_Jacqui" as a speaker
transcripts <- getMatchingGraphIds(labbcat.url, "'QB247_Jacqui' IN labels('who')", 1, 1)

## Get all transcripts whose names start with "BR" and have "QB247_Jacqui" as a speaker, # in word-count order
transcripts <- getMatchingGraphIds(labbcat.url, "my('corpus').label = 'QB' AND 'QB247_Jacqui' IN labels('who')", 1, 1, "my('transcript_word_count').label ASC")

## End(Not run)

getMedia

getMedia(labbcat.url, id, trackSuffix = "", mimeType = "audio/wav")

Arguments

labbcat.url URL to the LaBB-CAT instance
id A graph ID (i.e. transcript name)
trackSuffix The track suffix of the media
mimeType The MIME type of the media

Value

A URL to the given media for the given graph

See Also

getGraphIds
Examples

```r
## Not run:
## define the LaBB-CAT URL
labbcat.url <- "https://labbcat.canterbury.ac.nz/demo/"

## Get URL for the WAV file for BR2044_OllyOhlson.eaf
media <- getMedia(labbcat.url, "BR2044_OllyOhlson.eaf")

## Get URL for the 'QuakeFace' video file for BR2044_OllyOhlson.eaf
media <- getMedia(labbcat.url, "BR2044_OllyOhlson.eaf", "_face", "video/mp4")

## End(Not run)
```

---

### getMediaTracks

`getMediaTracks(labbcat.url)`

**Description**

List the predefined media tracks available for transcripts.

**Usage**

`getMediaTracks(labbcat.url)`

**Arguments**

- `labbcat.url` URL to the LaBB-CAT instance

**Value**

A list of media track definitions.

**Examples**

```r
## Not run:
## Get the media tracks configured in LaBB-CAT
tracks <- getMediaTracks("https://labbcat.canterbury.ac.nz/demo/")

## End(Not run)
```
getParticipantIds  

*Description*

Returns a list of participant IDs.

*Usage*

```r
getParticipantIds(labbcat.url)
```

*Arguments*

- `labbcat.url` URL to the LaBB-CAT instance

*Value*

A list of participant IDs

*Examples*

```r
## Not run:
## List all speakers
speakers <- getParticipantIds("https://labbcat.canterbury.ac.nz/demo/")

## End(Not run)
```

getSoundFragments  

*Description*

Gets sound fragments from 'LaBB-CAT'.

*Usage*

```r
getsoundfragments(
    labbcat.url, id, start, end, sampleRate = NULL, no.progress = FALSE, path = ""
)
```
**getSoundFragments**

**Arguments**

- **labbcat.url** URL to the LaBB-CAT instance
- **id** The graph ID (transcript name) of the sound recording, or a vector of graph IDs.
- **start** The start time in seconds, or a vector of start times.
- **end** The end time in seconds, or a vector of end times.
- **sampleRate** Optional sample rate in Hz - if a positive integer, then the result is a mono file with the given sample rate.
- **no.progress** Optionally suppress the progress bar when multiple fragments are specified - TRUE for no progress bar.
- **path** Optional path to directory where the files should be saved.

**Value**

The name of the file, which is saved in the current directory, or a list of names of files, if multiple id's/start's/end's were specified.

If a list of files is returned, they are in the order that they were returned by the server, which *should* be the order that they were specified in the id/start/end lists.

**Examples**

```r
## Not run:
## define the LaBB-CAT URL
labbcat.url <- "https://labcat.canterbury.ac.nz/demo/"

## Get the 5 seconds starting from 10s after the beginning of a recording
wav.file <- getSoundFragments(labbcat.url, "AP2505_Nelson.eaf", 10.0, 15.0, path="samples")

## Get the 5 seconds starting from 10s as a mono 22kHz file
wav.file <- getSoundFragments(labbcat.url, "AP2505_Nelson.eaf", 10.0, 15.0, 22050)

## Load some search results previously exported from LaBB-CAT
results <- read.csv("results.csv", header=T)

## Get a list of fragments
wav.files <- getSoundFragments(labbcat.url, results$Transcript, results$Line, results$LineEnd)

## Get a list of fragments with no progress bar
wav.file <- getSoundFragments(
    labbcat.url, results$Transcript, results$Line, results$LineEnd, no.progress=TRUE)

## End(Not run)
```
labbcatCredentials

Sets the username and password that the package should use for connecting to a given LaBB-CAT server in future function calls.

Description

This step is optional, as all functions will prompt the user for the username and password if required. If the script is running in RStudio, then the RStudio password input dialog is used, hiding the credentials from view. Otherwise, the console is used, and credentials are visible.

Usage

labbcatCredentials(labbcat.url, username, password)

Arguments

- labbcat.url: URL to the LaBB-CAT instance
- username: The LaBB-CAT username, if it is password-protected
- password: The LaBB-CAT password, if it is password-protected

Details

The recommended approach is to *not* use labbcatCredentials, to avoid saving user credentials in script files that may eventually become visible to other. Use labbcatCredentials *only* in cases where the script execution is unsupervised.

Value

NULL if the username/password are correct, and a string describing the problem if a problem occurred, e.g. "Credentials rejected" if the username/password are incorrect, or a string starting "Version mismatch" if the server’s version of LaBB-CAT is lower than the minimum required.

Examples

```r
## Not run:
## define the LaBB-CAT URL
labbcat.url <- "https://labbcat.canterbury.ac.nz/demo/"

## specify the username/password in the script
## (only use labbcatCredentials for scripts that must execute unsupervised!)
labbcatCredentials(labbcat.url, "demo", "demo")
```

## End(Not run)
labbcatTimeout

Sets the timeout for request to the LaBB-CAT server in future function calls. The default timeout is 10 seconds.

Description

Sets the timeout for request to the LaBB-CAT server in future function calls. The default timeout is 10 seconds.

Usage

labbcatTimeout(seconds = NULL)

Arguments

seconds The number of seconds before requests return with a timeout error.

Value

The request timeout in seconds

Examples

## Not run:
## the request timeout
labbcatTimeout(30)
## End(Not run)

nzilbb.labbcat

Accessing Data Stored in 'LaBB-CAT' Instances

Description

'LaBB-CAT' is a web-based language corpus management system developed by the New Zealand Institute of Language, Brain and Behaviour (NZILBB) - see <https://labbcat.canterbury.ac.nz>. This package defines functions for accessing corpus data in a 'LaBB-CAT' instance. You must have at least version 20200108.1025 of 'LaBB-CAT' to use this package. For more information about 'LaBB-CAT', see Robert Fromont and Jennifer Hay (2008) <doi:10.3366/E1749503208000142> or Robert Fromont (2017) <doi:10.1016/j.csl.2017.01.004>.
Details

Package: nzilbb.labbcat
Version: 0.4-2
Date: 2020-01-16
Title: Accessing Data Stored in 'LaBB-CAT' Instances
Authors@R: c(person("Robert", "Fromont", role = c("aut", "cre"), email = "robert.fromont@canterbury.ac.nz"))
Imports: jsonlite, httr, stringr, utils, rstudioapi
Description: 'LaBB-CAT' is a web-based language corpus management system developed by the New Zealand Institute of Language, Brain and Behaviour at the University of Canterbury. The R package 'nzilbb.labbcat' provides an interface to 'LaBB-CAT' for R users. It allows access to data stored in 'LaBB-CAT' instances. The package requires 'LaBB-CAT' to be installed and configured properly. The developers of 'LaBB-CAT' are Robert Fromont and Jennifer Hay (2008) <doi:10.3366/E1749503208000142> or Robert Fromont (2017) <doi:10.1016/j.csl.2017.01.004>.
License: GPL (>= 3)
Copyright: New Zealand Institute of Language, Brain and Behaviour, University of Canterbury
RoxygenNote: 7.0.2
Suggests: testthat (>= 2.1.0)
Author: Robert Fromont [aut, cre]
Maintainer: Robert Fromont <robert.fromont@canterbury.ac.nz>

Index of help topics:

- countAnnotations: Gets the number of annotations on the given layer of the given graph.
- getAnchors: Gets the given anchors in the given graph.
- getAnnotationLabels: Gets labels of annotations on a given layer, identified by given annotation IDs.
- getAnnotations: Gets the annotations on the given layer of the given graph.
- getAvailableMedia: List the media available for the given graph.
- getCorpusIds: Gets a list of corpus IDs.
- getDictionaries: List the dictionaries available.
- getFragments: Gets fragments of annotation graphs (transcripts) from 'LaBB-CAT', converted to a given format (by default, Praat TextGrid).
- getGraphIds: Gets a list of graph IDs.
- getGraphIdsInCorpus: Gets a list of corpus IDs.
- getGraphIdsWithParticipant: Gets a list of IDs of graphs that include the given participant.
- getId: Gets the store's ID.
- getLayer: Gets a layer definition.
- getLayerIds: Gets a list of layer IDs.
- getLayers: Gets a list of layer definitions.
- getMatches: Search for tokens.
- getMatchingGraphIds: Gets a list of IDs of graphs that match a particular pattern.
- getMedia: Gets a given media track for a given graph.
- getMediaTracks: List the predefined media tracks available for
transcripts.

getParticipantIds Gets a list of participant IDs.
getSoundFragments Gets sound fragments from 'LaBB-CAT'.
labbcatCredentials Sets the username and password that the package
should use for connecting to a given LaBB-CAT
server in future function calls.
labbcatTimeout Sets the timeout for request to the LaBB-CAT
server in future function calls. The default
timeout is 10 seconds.
nzilbb.labbcat Accessing Data Stored in 'LaBB-CAT' Instances

'LaBB-CAT' is a web-based language corpus management system and this package provides access
to data stored in a 'LaBB-CAT' instance. You must have at least version 20200108.1025 of 'LaBB-
CAT' to use this package.

Author(s)
NA

References
Robert Fromont and Jennifer Hay, "ONZE Miner: the development of a browser-based research
tool", 2008 Robert Fromont, "Toward a format-neutral annotation store", 2017

Examples
## Not run:
## define the LaBB-CAT URL
labbcat.url <- "https://labbcat.canterbury.ac.nz/demo/

## Load some search results previously exported from LaBB-CAT
results <- read.csv("results.csv", header=T)

## Get the phonemic transcriptions for the matches
phonemes <- getAnnotationLabels(labbcat.url, results$MatchId, "phonemes")

## Get sound fragments for the matches
wav.files <- getSoundFragments(labbcat.url, results$Transcript, results$Line, results$LineEnd)

## End(Not run)
Index

*Topic anchor
  getAnchors, 3

*Topic annotation
  getAnnotationLabels, 4

*Topic audio
  getAvailableMedia, 6
  getMedia, 20

*Topic connect
  labbcatCredentials, 24
  labbcatTimeout, 25

*Topic corpora
  getCorpusIds, 7
  getGraphIdsInCorpus, 11

*Topic corpus
  getGraphIdsInCorpus, 11

*Topic dictionary
  getDictionaries, 8
  getDictionaryEntries, 8

*Topic expression
  getMatchingGraphIds, 19

*Topic fragment
  getFragments, 9
  getSoundFragments, 22

*Topic graph
  countAnnotations, 2
  getAnnotations, 5
  getGraphIds, 11
  getGraphIdsWithParticipant, 12
  getMatchingGraphIds, 19

*Topic label
  getAnnotationLabels, 4

*Topic layer
  getAnnotationLabels, 4
  getLayer, 13
  getLayerIds, 14
  getLayers, 15

*Topic media
  getAvailableMedia, 6
  getMedia, 20
  getMediaTracks, 21

*Topic package
  nzilbb.labbcat, 25

*Topic participant
  getParticipantIds, 22

*Topic password
  labbcatCredentials, 24
  labbcatTimeout, 25

*Topic sample
  getFragments, 9
  getSoundFragments, 22

*Topic search
  getMatches, 16

*Topic sound
  getFragments, 9
  getMediaTracks, 21
  getSoundFragments, 22

*Topic speaker
  getParticipantIds, 22

*Topic timeout
  labbcatCredentials, 24
  labbcatTimeout, 25

*Topic transcript
  countAnnotations, 2
  getAnnotations, 5
  getGraphIds, 11
  getGraphIdsWithParticipant, 12
  getMatchingGraphIds, 19

*Topic username
  labbcatCredentials, 24
  labbcatTimeout, 25

*Topic wav
  getFragments, 9
  getSoundFragments, 22

  countAnnotations, 2, 5
  getAnchors, 3
  getAnnotationLabels, 4
  getAnnotations, 3, 5
INDEX

getAvailableMedia, 6
getCorpusIds, 7
getDictionaries, 8, 9
getDictionaryEntries, 8, 8
getFragments, 9
getGraphIds, 3, 5, 6, 11, 20
getGraphIdsInCorpus, 3, 5, 11
getGraphIdsWithParticipant, 3, 5, 12
getId, 13
getLayer, 13
getLayerIds, 14, 14, 15
getLayers, 14, 15
getMatches, 16
getMatchingGraphIds, 19
getMedia, 20
getMediaTracks, 21
getParticipantIds, 12, 18, 22
getSoundFragments, 22

labbcatCredentials, 24
labbcatTimeout, 25

nzilbb.labbcat, 25