

Test

Table of contents

```
library(httr)
library(jsonlite)
library(dplyr)
```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

filter, lag

The following objects are masked from 'package:base':

intersect, setdiff, setequal, union

```
library(purrr)
```

Attaching package: 'purrr'

The following object is masked from 'package:jsonlite':

flatten

```
library(openxlsx)
```

```
extract_textquote_prefix <- function(annotation_data) {  
  # Initialize a vector to store the extracted prefixes  
  prefixes <- vector("list", length(annotation_data$target))  
  
  # Loop through each row of the annotation data  
  for (i in seq_along(annotation_data$target)) {  
    # Extract the selectors for the current target  
    selectors <- annotation_data$target[[i]]$selector  
  
    # Filter for "TextQuoteSelector" and extract the prefix  
    if (!is.null(selectors)) {  
      prefixes[[i]] <- selectors[[1]] |>  
        dplyr::filter(type == "TextQuoteSelector") |>  
        dplyr::select(prefix) |>  
        dplyr::pull()  
    } else {  
      prefixes[[i]] <- NA # Handle cases where target/selector is missing  
    }  
  }  
  
  # Return a flattened vector or a list depending on requirements  
  annotation_data |>  
    dplyr::bind_cols(as.data.frame(purrr::flatten_chr(prefixes)) |>  
      rename(quote = "purrr::flatten_chr(prefixes)"))  
}
```

```
# List of URLs to loop over
```

```
urls <- c(  
  "https://research-it-swiss-tph.github.io/quarto_training/index.html",  
  "https://research-it-swiss-tph.github.io/quarto_training/notes/quarto_intro.html",  
  "https://research-it-swiss-tph.github.io/quarto_training/notes/notebook_structure.html",  
  "https://research-it-swiss-tph.github.io/quarto_training/notes/python_r_short_demo.html",  
  "https://research-it-swiss-tph.github.io/quarto_training/practicals/exercise1_gettingsta",  
  "https://research-it-swiss-tph.github.io/quarto_training/practicals/exercise2_gettingsta",  
  "https://research-it-swiss-tph.github.io/quarto_training/practicals/exercise3.html",  
  "https://research-it-swiss-tph.github.io/quarto_training/practicals/exercise3_instructio",  
  "https://research-it-swiss-tph.github.io/quarto_training/practicals/exercise4_instructio",  
  "https://research-it-swiss-tph.github.io/quarto_training/practicals/exercise5_instructio"
```

```

"https://research-it-swiss-tph.github.io/quarto_training/correction/exercice3_test.html"
"https://research-it-swiss-tph.github.io/quarto_training/correction/exercise3_correction
"https://research-it-swiss-tph.github.io/quarto_training/correction/exercise3_step_by_st
"https://research-it-swiss-tph.github.io/quarto_training/correction/exercise4_step_by_st
)

```

```

all_annotations <- tibble()
for (url in urls) {
  api_url <- paste0("https://api.hypothes.is/api/search?uri=", url)
  response <- GET(api_url)

  if (status_code(response) == 200) {
    annotations <- fromJSON(content(response, "text"))
    if (!is.null(annotations$rows)) {
      annotation_data <- extract_textquote_prefix(annotations$rows)
      all_annotations <- bind_rows(all_annotations, annotation_data)
    }
  } else {
    message(paste("Failed to fetch data for URL:", url))
  }
}

```

No encoding supplied: defaulting to UTF-8.
 No encoding supplied: defaulting to UTF-8.
 No encoding supplied: defaulting to UTF-8.
 No encoding supplied: defaulting to UTF-8.
 No encoding supplied: defaulting to UTF-8.
 No encoding supplied: defaulting to UTF-8.
 No encoding supplied: defaulting to UTF-8.
 No encoding supplied: defaulting to UTF-8.
 No encoding supplied: defaulting to UTF-8.
 No encoding supplied: defaulting to UTF-8.
 No encoding supplied: defaulting to UTF-8.
 No encoding supplied: defaulting to UTF-8.
 No encoding supplied: defaulting to UTF-8.

```

all_annotations <- all_annotations |>
  as.data.frame() |>
  dplyr::select(created,

```

```

      text,
      quote,
      user,
      uri) |>
dplyr::rename(date = created,
              comment = text) |>
dplyr::mutate(date = as.Date(date),
              user = sub(".*:(.*)@.*", "\\1", user))
all_annotations

```

```

      date
1 2025-01-23
2 2025-02-10
3 2025-02-10
4 2025-02-10
5 2025-02-10
6 2025-02-14
7 2025-01-23

```

```

                                comment
1                                Done
2                                .q
3                                Is this sentence correct?
4                                Such as? Would be interested to get some suggestions
5                                Maybe introduce the native pipe quickly? : )
6 This part doesn't have corresponding one in step-by-step correction.
7                                Done, should work

```

```

                                quote    user
1                                <NA> hlanget
2                                Create to open up a new Quarto ( axelir
3                                own or Quarto documents to PDF.\n axelir
4 .5\n0.1\nsetosa\n\n\n\n\n\n\n\n\n\n\n\nTip\n\n\n\n\n axelir
5                                aset, one can use:\n\n``{r}\niris axelir
6                                ho are still alive"\n\n\n\nTable 3\n\n zhuzh
7                                <NA> zhuzh

```

```

1                                https://research-it-swiss-tph.github.io/quarto_training/notes/python_r_
2                                https://research-it-swiss-tph.github.io/quarto_training/practicals/exercise1_gett
3                                https://research-it-swiss-tph.github.io/quarto_training/practicals/exercise2_gett
4                                https://research-it-swiss-tph.github.io/quarto_training/practicals/exercise2_gett
5                                https://research-it-swiss-tph.github.io/quarto_training/practicals/exercise2_gett
6                                https://research-it-swiss-tph.github.io/quarto_training/practicals/exercise4_in
7 https://research-it-swiss-tph.github.io/quarto_training/correction/exercise3_step_by_step_

```

```
# Write the annotation data to a CSV file
openxlsx::write.xlsx(all_annotations, 'annotations.xlsx')

# Print a message to indicate completion
cat("Annotations have been saved to 'annotations.csv'.")
```

Annotations have been saved to 'annotations.csv'.