Test

Table of contents

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

```
Attache Paket: 'dplyr'

Die folgenden Objekte sind maskiert von 'package:stats':

    filter, lag

Die folgenden Objekte sind maskiert von 'package:base':

    intersect, setdiff, setequal, union
```

```r
library(purrr)
```

```
Attache Paket: 'purrr'

Das folgende Objekt ist maskiert 'package:jsonlite':

    flatten
```

```r
library(openxlsx)
```

```r
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)"))
}
```

```r
# 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.
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           comment quote    user
1 2025-01-23              Done  <NA> hlanget
2 2025-01-23 Done, should work  <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/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'.
```