library('RCurl')

library('XML')

# This code will scrape metadata for a given GitHub repository.

# Example url: https://github.com/aem/website

# The data will include the number of commits to the repo, the number of active

# branches, the number of releases, the number of contributors, and the primary

# language in the repository

getUrl <- function(repo) {

# Turns a GitHub repository name into a GitHub URL

# Args:

# repo [Character]: the name of the repository

# Returns: [Character] the URL to the repository

return(paste('https://github.com/', repo, sep=""))

}

getNumberFromXPath <- function(pagetree, xpath) {

# Extracts a number from a tree at a path

# Args:

# pagetree [Character]: the XML tree to parse

# xpath [Character]: the path to search for the number

# Returns: [Number] the number at the given xpath

return(gsub("[^0-9]", "", unlist(xpathApply(pagetree, xpath, xmlValue))))

}

getTextFromXPath <- function(pagetree, xpath) {

# Extracts text from a tree at a path

# Args:

# pagetree [Character]: the XML tree to parse

# xpath [Character]: the path to search for the text

# Returns: [Character] the text at the given xpath

return(gsub("/n/r", "", unlist(xpathApply(pagetree, xpath, xmlValue))))

}

crawlAndStoreRepos <- function(names) {

# Extracts basic GitHub repository data from each of the given repositories.

# Args:

# names [Vector]: the list of github repository names

# Returns: [Data Frame] basic metadata about each of the repos

repoData <- data.frame(Name=character(),

Commits=integer(),

Branches=integer(),

Releases=integer(),

Contributors=integer(),

Language=character(),

stringsAsFactors=FALSE)

for(name in names) {

rawWebpage <- RCurl::getURL(getUrl(name))

tc <- textConnection(rawWebpage)

webpage <- readLines(tc)

close(tc)

pagetree <- htmlTreeParse(webpage, useInternalNodes = TRUE)

commits <- getNumberFromXPath(pagetree, "//\*[@class='numbers-summary']/li[1]/a/span")

branches <- getNumberFromXPath(pagetree, "//\*[@class='numbers-summary']/li[2]/a/span")

releases <- getNumberFromXPath(pagetree, "//\*[@class='numbers-summary']/li[3]/a/span")

contributors <- getNumberFromXPath(pagetree, "//\*[@class='numbers-summary']/li[4]/a/span")

language <- getTextFromXPath(pagetree, "//\*[@class='repository-lang-stats']/ol/li[1]/a/span[2]")

if (length(language) == 0) {

language <- "None"

}

repoData[nrow(repoData) + 1,] <- c(name,

commits,

branches,

releases,

contributors,

language)

}

return(repoData)

}

repos <- c('aem/docs-soap',

'github/gitignore',

'facebook/react',

'opengovernment/opengovernment',

'reactjs/redux',

'yavijava/yavijava')

result <- crawlAndStoreRepos(repos)

if (nrow(result) == length(repos)) {

print("SUCCESS")

print(result)

}

