

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Белгородский государственный технологический университет
им. В.Г. Шухова»

ИНСТИТУТ ЭНЕРГЕТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
И УПРАВЛЯЮЩИХ СИСТЕМ

Кафедра программного обеспечения вычислительной
техники и автоматизированных систем

Лабораторная работа
по дисциплине: Безопасность программно-информационных
систем
тема: «Работа с пакетом (R)Selenium и Docker Toolbox для
получения данных с веб-ресурсов»

Работу выполнил:
студент группы ПВ-31
Притчин Иван Сергеевич

Проверил:

Макаров Антон Михайлович

Дата сдачи:
« ____ » _____ 2018 г.

Задача: создать приложение, которое скачает произвольный учебник, недоступный для загрузки из сайта библиотеки БГТУ им. Шухова.

Основные подзадачи:

1. Запуск сервера
2. Авторизация на сайте библиотеки
 - a. Выбор способа авторизации
 - b. Авторизация на выбранном ресурсе
3. Получение адреса скачиваемой книги
4. Получение файлов книги
 - a. Подбор масштаба
 - b. Получение изображения
 - c. Скачивание изображения
 - d. Возврат на прошлую страницу
 - e. Переход в новой странице
 - f. Повтор до тех пор, пока не будет скачана вся книга
5. Конвертирование полученных файлов в pdf

Этапы выполнения работы

Запуск сервера

Установка Docker Toolbox. Подключение библиотеки Selenium:

```
Данил@Maker-Book MINGW64 /c/Program Files/Docker Toolbox
$ docker pull selenium/standalone-firefox:2.53.0
2.53.0: Pulling from selenium/standalone-firefox
cad964aed91d: Pull complete
3a80a22fea63: Pull complete
50de990d7957: Pull complete
61e032b8f2cb: Pull complete
9f03ce1741bf: Pull complete
fb6ea679a99b: Pull complete
fb570799429a: Pull complete
eae20523acb1: Pull complete
a67b6012f3e0: Pull complete
8e8cd675bac1: Pull complete
41ee78787d3b: Pull complete
c90228671973: Pull complete
83a2dba40982: Pull complete
f6a754bc2675: Pull complete
f13f961db352: Pull complete
d025228ca460: Pull complete
ebd6b790185d: Pull complete
Digest: sha256:097f3e0ce15e1a8313c5284bb11ae7391e7d48428c907e154bd040da5581c414
Status: Downloaded newer image for selenium/standalone-firefox:2.53.0

Данил@Maker-Book MINGW64 /c/Program Files/Docker Toolbox
$ docker run -d -p 4445:4444 selenium/standalone-firefox:2.53.0
6f5c0db22945edc1d2c6a08f142e5463e92d35f461b8582b5bf6a7eda38755f8

Данил@Maker-Book MINGW64 /c/Program Files/Docker Toolbox
$ docker ps
CONTAINER ID        IMAGE                                     COMMAND                  NAMES
6f5c0db22945        selenium/standalone-firefox:2.53.0     "/opt/bin/entry_poin"   compassionate_
fermi
20 seconds ago      Up 17 seconds          0.0.0.0:4445->4444/tcp

Данил@Maker-Book MINGW64 /c/Program Files/Docker Toolbox
$ docker-machine ip
192.168.99.100
```

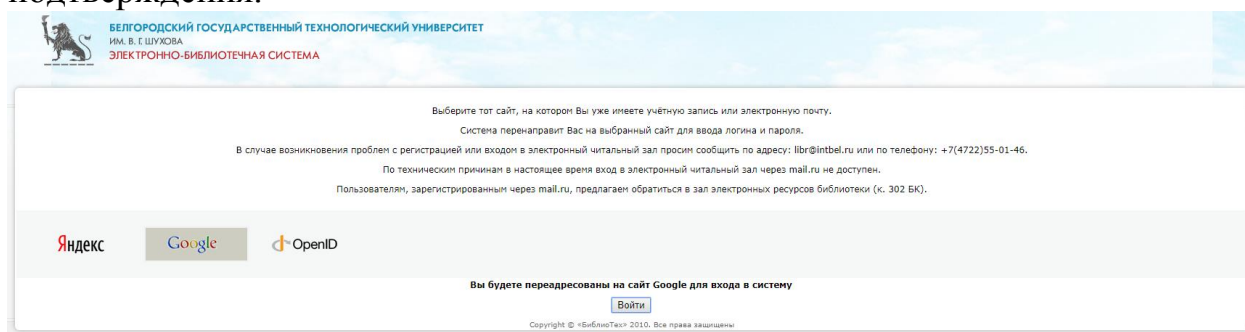
Подключение библиотек для работы с браузером из R:

```
library(RSelenium)
library(RCurl)
library(purrr)
library(stringr)

remDr <- remoteDriver(remoteServerAddr = "192.168.99.100", port = 4445L)
remDr$open()
```

Авторизация на сайте библиотеки

При попытке прогрузить страницу, принадлежащей библиотеке, встаёт вопрос авторизации. Была выбрана авторизация через Google. Выполнен поиск элемента на странице через класс. Произведен щелчок и переход на кнопку подтверждения.



```
authorizationInLibrary <- function() {
  remDr$navigate("https://elib.bstu.ru/Account/OpenID")
  remDr$screenshot(display = TRUE)

  webElem <- remDr$findElement(using = 'class', value = "google")
  webElem$clickElement()
  printPage()

  sleep(1)
  webElem <- remDr$findElement(using = 'id', value = "_sb")
  webElem$clickElement()
}
```

Аналогичный процесс проходим на сайте google. Выбираем поля по идентификатору для логина, и таб-индексу для пароля. Причина, по которой был выбран таб-индекс заключается в том, что класс для поля ввода пароля генерируется на ходу, а id отсутствует.

```
authorizationInGoogle <- function() {
  webElem <- remDr$findElement(using = 'id', value = "identifierId")
  webElem$clickElement()
  webElem$sendKeysToElement(list("Место для вашего логина!", key = "enter"))
  sleep(4)
  webElem <- remDr$findElement(using = 'css', value = "[tabindex='0']")
  webElem$sendKeysToElement(list("И пароля!", key = "enter"))
  printPage()
}
```

Войдите в аккаунт Google

Вход

Переход в приложение "bstu.ru"

Телефон или адрес эл. почты

[Забыли адрес эл. почты?](#)


[Создать аккаунт](#)

[ДАЛЕЕ](#)

Русский ▾ Справка Конфиденциальность Ус...

Войдите в аккаунт Google

Добро пожаловать!



Введите пароль

[Забыли пароль?](#)

[ДАЛЕЕ](#)

Русский ▾ Справка Конфиденциальность Условия

```
<input type="password" class="whsOnd zHQkBf" jsname="YPqjbf" autocomplete="current-password"
spellcheck="false" tabindex="0" aria-label="Введите пароль" name="password" autocapitalize="off"
autocorrect="off" data-initial-value>
```

После прохождения процедуры мы можем убедиться в корректности проведённых операций вызывав описанную в пользовательском модуле функцию printPage():

```
printPage <- function() {remDr$screenshot(display = TRUE)}
```

БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ
ИМ. В. Г. ШУХОВА
ЭЛЕКТРОННО-БИБЛИОТЕЧНАЯ СИСТЕМА

[ВЫХОД](#)

Личная

Ключевое слово или фраза

Дополнительные параметры

☐ Без картинок

Найдено изданий: 4664, выводить по 50 ▾

1 2 3 4 ... 94 [следующая](#)

База данных : методические указания к выполнению лабораторных работ для студентов очной формы обучения направления подготовки 38.03.05 – Бизнес-информатика

Авторы: [Лазарева А. Ю.](#)

Издательство: Изд-во БГТУ им. В. Г. Шухова

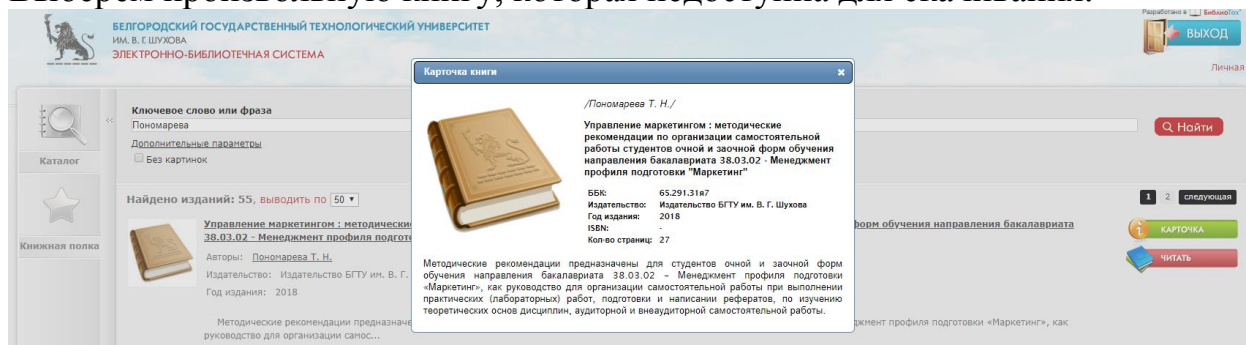
Год издания: 2017

[КАРТОЧКА](#)

[ЧИТАТЬ](#)

Получение адреса скачиваемой книги

Выберем произвольную книгу, которая недоступна для скачивания.



Перейдём по «Читать». Адрес в строке и будет считаться основным, от которого мы начнём боевые действия

<https://elib.bstu.ru/Reader/Book/2018041010203127600000659459>

Скачивание книги

Опишем ход мыслей по этапам, аргументируем принятые решения.

Изначально задача казалась тривиальной. Надо найти id элемента в изображении и скачать. Однако, движение в лоб показало, что при загрузке страницы на компьютер, её качество было хуже, чем нужно для нормального восприятия. Ещё был интересен и тот факт, что для разных машин при загрузке одной и той же книги, ссылки на изображения были разными. Вывод – ссылки полученные от одного браузера не могли использоваться для скачивания изображений «руками» или отдельным программным средством на другом. Также было замечено, что при изменении масштаба генерируемое изображение меняет свой адрес. Следовательно, перед загрузкой нам было необходимо настроить масштаб.

Настройка масштаба осуществлялась следующим образом: затирались старые значения масштаба (значение 100). И заменялось произвольное value. В нашем случае было выбрано значение 200. После ввода выполнено нажатие клавиши «Tab» и «Enter»

```
setScale <- function(value) {  
  webElem <- remDr$findElement(using = 'id', value = "scale")  
  webElem$sendKeysToElement(list("\u0003", "\u0003", "\u0003", as.character(value),  
  "\u0004", "\u0007"))  
}
```

Получение изображения. Для получения изображения был найден id содержащий необходимую информацию. Адрес возвращен как результат:

```
getPageLink <- function() {  
  webElem <- remDr$findElement(using = 'id', value = "thePage")  
  webElem$getAttribute("src")[[1]]  
}
```

Скачивание изображения. Выполняется посредством перехода по нужной ссылке и сохранении файла в необходимую директорию

```
remDr$navigate(link)
saveImage(oldNumPage + 1)
```

а функция **сохранения изображения** представляет собой:

```
saveImage <- function(i) {remDr$screenshot(
file = str_c(c("C:/Users/Данил/Desktop/1/", as.character(i), ".png"), collapse = "")
)}
```

Возврат на прошлую страницу осуществляется:

```
remDr$goBack()
```

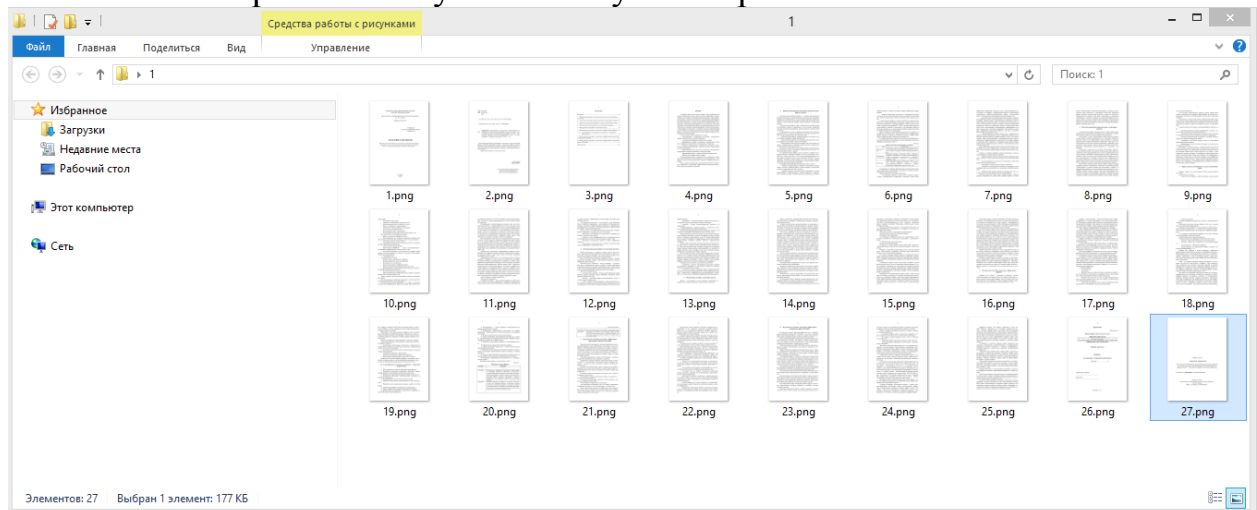
Переход в новой странице заключается в поиске элемента, которые отвечает за выполнение операции и нажатии на него:

```
nextPage <- function() {
  el <- remDr$findElements(using = 'class', value = "nextbutton")[[1]]
  el$clickElement()
}
```

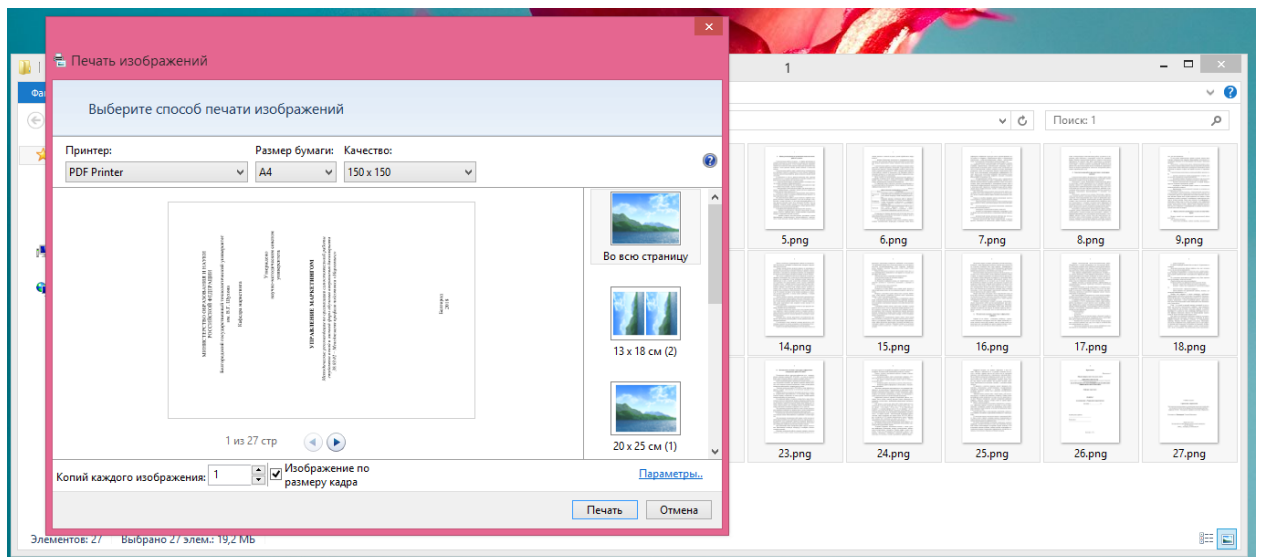
Все перечисленные операции будут выполняться до тех пор, пока не будет скачена вся книга. Это будет понятно, когда номер новой страницы после перехода будет совпадать с номером старой страницы. Для получения номера текущей страницы используется функция:

```
getCurrentNumPage <- function() {
  as.numeric(
    remDr$findElement(
      using = 'id',
      value = "curPageNum"
    )$getElementAttribute("value")[[1]]
  )
}
```

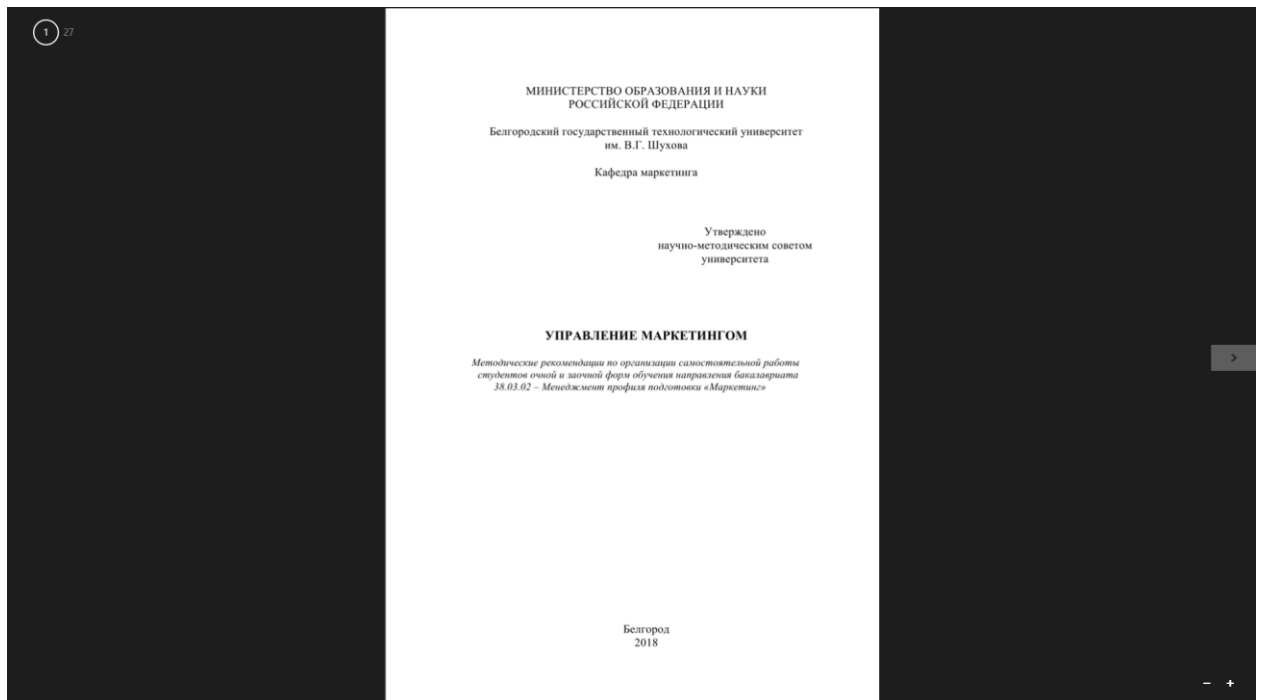
По окончанию работы получаем папку с изображениями:



Склейку полученных файлов выполним через виртуальный pdf-принтер



Откроем полученный файл посредством reader:



Заметим, что название книги и количество страниц совпадают с заявленным. На этом работу можно считать законченной.