Application for GSOC 2018

Firedata

Adding Web Functionalities for Shiny and OpenCPU

Paul Spende

MSc. Computer Science at TU Berlin, Exchange Student at RMIT

Mentors:
Robin Kohze, Samuel Schmidt, Laurent Gatto, Bert Jehoul



Google
Summer of Code

1 Project info

Project title:

Firedata - Adding Web Functionalities for Shiny and OpenCPU

Project short title:

Firedata

URL of project idea page:

https://github.com/rstats-gsoc/gsoc2018/wiki/Firedata—The-Google-Firebase- Package

2 Mentors

Mentor name: Robin Kohze

Mentor email: robin@kohze.com

I know Robin from his R-Codes platform where he shared many cloud re- lated R tutorials. In addition to his project proposal, we discussed the many chances and challenges Firedata offers for the larger R community.

Mentor name: Samuel Schmidt
Mentor email: samuelschmidt218@gmail.com

Samuel is overviewing the development of the Firedata package in terms of real-world usage. He contributes with his unique perspective and insight about the difficulties and challenges of the creation of scientific data applications.

Mentor name: Laurent Gatto **Mentor email:** lg390@cam.ac.uk

Laurent has worked on the initial creation of Firedata and now overviews the package development process.

Mentor name: Bert Jehoul
Mentor email: bert@openknowledge.be

Bert is involved in the open knowledge organization and is interested to enable a wider usage of the Firedata package for various platforms within the R community.

Integrate Firedata with Shiny and openCPU

3 Coding Plan and Methods

Shiny enables R developer to build interactive web apps. It is possible either to host a standalone app on a webpage, embed them in R Markdown documents or build dashboards. Furthermore, it is also possible to augment them with CSS themes, htmlwidgets, and JavaScript actions.

OpenCPU is a more backend based framework compared to Shiny, that works over a REST API, which the web service needs to send the data to for the calculations. It is more performant than Shiny but doesn't offer the integrated visualizations of Shiny.

To integrate Shiny and openCPU with Firedata, I implement the additional Authorization functions like the social login functions and the anonymous login. Afterwards, I implement a custom widget for shiny that offers a login before the actual visualization. Thereby you can customize the data of the visualization based on the current viewer. It would also be possible to restrict certain access to data depending on the user. Firebase could also host Shiny widgets and applications then. Once these functions are put in place I also want to implement these mechanisms for OpenCPU.

Firebase hosting simplifies web hosting with tools made specifically for modern web apps. When web assets are uploaded, they automatically push them out to their global CDN and give them a free SSL certificate so the user's get a secure, reliable, low-latency experience, no matter where they are.

4 Test

The first test was to follow the instructions on github using the mtcars dataset.

```
if (!require("devtools")) install.packages("devtools")
devtools::install_github("Kohze/fireData")

library(fireData)

projectUrl = "https://firedata-b0e54.firebaseio.com/" apiKey =
"AIzaSyAjZLO9-CRV3gObpwdFz-k8AiTOxHSBmdc" apiSecret =
"2bYA6k72wKna90MqPGa6yuMG7jAysoDJZwJqYXsm"

userEmail = "paul.spende@gmail.com"

createUser(projectAPI = apiKey, email = userEmail,
```

```
password = "test4321" )

authData = auth(projectAPI = apiKey, email = userEmail,

password = "test4321")
timeout = Sys.time() + as.numeric(authData$expiresIn)

if(Sys.time() > timeout) {
authData = auth(projectAPI = api_key,

email = userEmail, password = "test4321")

}

location = upload(x = mtcars,
projectURL = projectUrl,

directory = "main",
token = authData$idToken)

download(projectURL = projectUrl, fileName = location,

token = authData$idToken)

resetPassword(projectAPI = apiKey, email = userEmail)

dataBackup(projectURL = projectUrl, secretKey = apiSecret,

fileName = "test.json")
```

The second task was to setup my own Firebase cloud, create my own user account via Firedata and upload/download a sample dataset (mtcars).

```
if (!require("devtools")) install.packages("devtools")
devtools::install_github("Kohze/fireData")

library(fireData)

projectUrl = "https://fir-abf00.firebaseio.com" apiKey =
"AIzaSyAYX_RxenP08jWpHExzsZIaO3CXeZhUSW8" apiSecret =
"aseOchXEnISsg9fSmhpz1YtmU9A10g4G3oj1fYVR"

userEmail = "paul.spende@gmail.com"
```

```
createUser(projectAPI = apiKey, email = userEmail,

password = "test4321" )

authData = auth(projectAPI = apiKey, email = userEmail,

password = "test4321")
timeout = Sys.time() + as.numeric(authData$expiresIn)

if(Sys.time() > timeout) {
authData = auth(projectAPI = apiKey,

}

email = userEmail, password = "test4321")

upload(x = mtcars,
projectURL = projectUrl, directory = "main",
token = authData$idToken)

download(projectURL = projectUrl, fileName = "main/",

token = authData$idToken)

dataBackup(projectURL = projectUrl, secretKey = apiSecret,

fileName = "test.json")
```

The third task was to add examples of the package use to the Rd files in the following format.

```
% Generated by roxygen2: do not edit by hand % Please edit documentation
in R/index.R \name{auth}
\alias{auth}

\title{The user authentication function:} \usage{

auth(projectAPI, email = "prompt", password = "prompt") }

\arguments{
\item{projectAPI}{The Firebase Project API {string}}

\item{email}{The user email {string}} \item{password}{The user password
{string}}
```

```
} \value{

Returns the content of the firebase API request, such as the state of
registration, idToken, and validity of the user password. }
\description{

fireData::auth checks the validity of a login and returns the temporary
JWT user token. FireData_auth can be used to store individual user data
in specified directories that are only accessible to that specific user.

} \examples{

\dontrun{ auth(projectAPI =

"AIzaSyAjZLO9-CRV3gObpwdFz-k8AiTOxHSBmdc", email = "robin@kohze.com",

password = "12341234") }

}
```

I implemented the examples for the functions upload, download, createUser, auth, resetPassword and dataBackup.

11 Further Involvement in Open Source

There are still some possible features of Firebase to be integrated into Firedata, which I would like to support the package developers to implement. I hope to be a part of the project to push it closer to being an officially recognized API for Firebase.

