Spring Boot + Thymeleaf + Redis + Heroku

This demo application has been created as an example of deploying Spring Boot + Thymeleaf + Redis on Heroku.

Technology Stack

- Spring Boot, no-xml Spring MVC 4 web application for Servlet 3.0 environment
- Spring Data Redis
- Database (Redis, Redis To Go)
- Thymeleaf templates with added Joda Time & Spring Security Dialects
- Heroku fully cloud deployable
- Testing (JUnit/Mockito/MockMVC/AssertJ/Hamcrest)
- Java 8, Spring Security 3.2, Maven 3, SLF4J, Logback, Bootstrap 3.3.4, jQuery 1.11.2, i18n, etc

Live Demo

Be aware that this application is currently running on a free Heroku account. If it hasn't been accessed in 30 minutes, then***the first request will take up to 120 seconds***. Note that the demo application might fail to load altogether if the Heroku servers are busy.

Here is the Redis Developer's Bookshelf running on Heroku.

Local Deployment

Load a local Redis database on port 6379. Flush the database with index equal to 0.

```
$ mvn clean install
$ mvn spring-boot:run
```

Navigate to http://localhost:8080.

The application can also be deployed by running the `Application.java` class.

Deploying to Heroku

The following steps require that the Heroku Toolbelt has been installed locally and that a Heroku account has been created.

Navigate to the project directory on the command line.

Before creating your Heroku application, make sure that there is a Git repository associated with the project.

```
$ git status
```

If a Git repository is not associated with the project, then create one before continuing.

Create a new application on Heroku.

```
$ heroku create
```

Rename your Heroku application if interested.

```
$ heroku apps:rename new-name
```

Add a Redis database to your Heroku application with the Redis To Go add-on. Note that your Heroku account must have a credit card attached in order to use free add-ons other than the PostgreSQL and MySQL add-ons.

```
$ heroku addons:create redistogo:nano
```

Deploy project to Heroku.

```
$ git push heroku master
```

Look at your application logs to see what is happening behind the scenes.

```
$ heroku logs
```

If your application deploys without timing out then open it as follows.

```
$ heroku open
```