

2021-05-22 • SPRING ► WEB

Spring Boot Web - Thymeleaf

A Spring boot web application with thymeleaf template & basic spring security support, uses bootstrap for CSS and chart.js for rendering charts. Creates uber jar to deploy.

Github: <https://github.com/gitorko/project79>

- [Spring Web](#)
- [chart.js](#)
- [Thymeleaf](#)
- [Bootstrap 5](#)
- [Setup](#)
 - [Docker](#)
 - [Postgres DB](#)
 - [Dev](#)
 - [Prod](#)
- [Screenshots](#)
- [References](#)

Spring Web



A Spring Web MVC application that renders thymeleaf templates as HTML. Supports basic integration with spring security and provides login logout support.

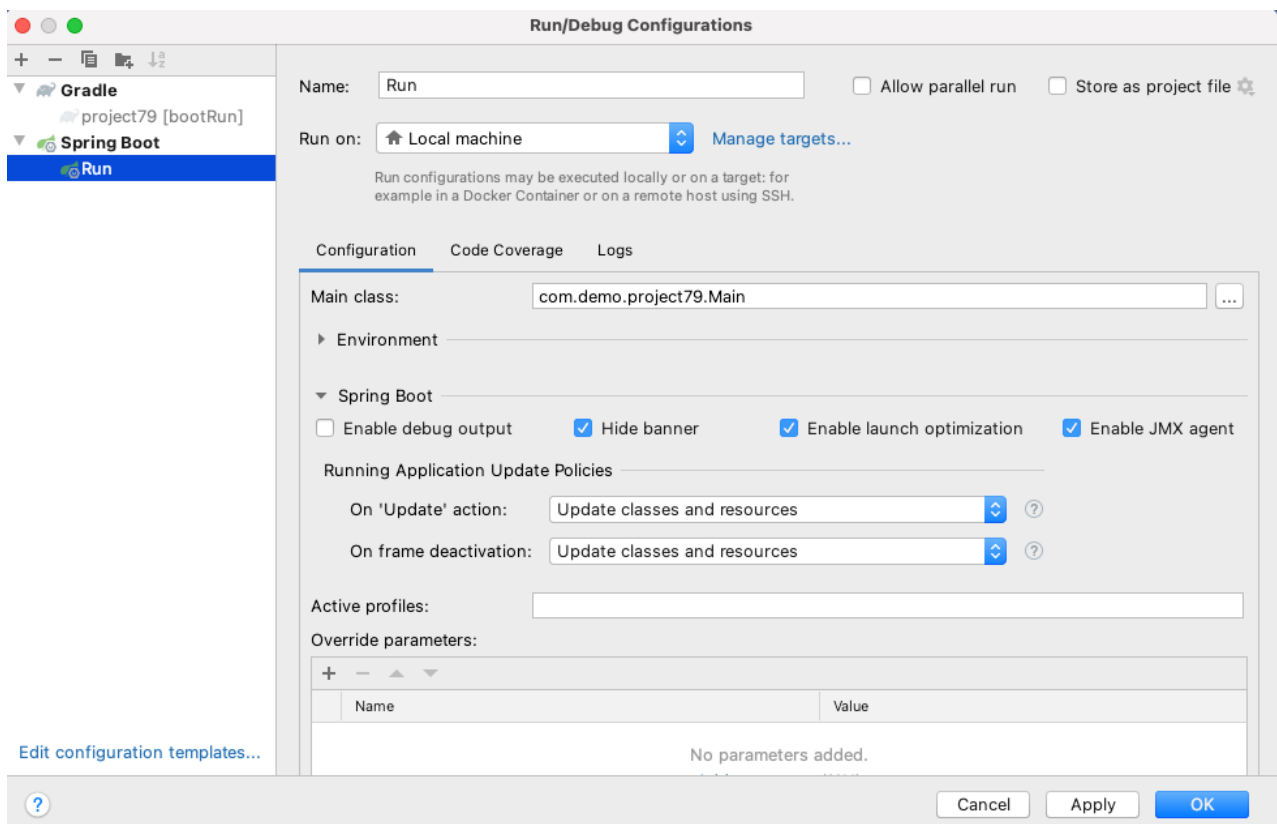
Uses Spring Data to persist data into the HSQL db. A file based HSQL server db is used so that data persists across restarts. This can easily be changed to in-memory HSQL db.

Spring dev tools allow seamless reload on any changes for html and java files so you can view the changes in the browser as soon as you edit them.

Features:

1. Supports basic login via spring security
2. Bootstrap 5
3. Login screen
4. CRUD UI for adding and removing customer
5. HSQL db
6. Spring JPA
7. Thymeleaf template
8. Chart.js charts for bar, pie, stack charts with data from rest api

On IntelliJ to allow spring dev tools to reload on change you need to enable 'Update classes and resources' as shown below



Spring MVC controller renders the HTML.

HomeController.java

[link](#)

```
29 @GetMapping(value = "/")
30 public String home(Model model) {
31     Iterable<Customer> customerLst = customerRepo.findAll();
32     model.addAttribute("customerLst", customerLst);
33     model.addAttribute("serverTime", new Date());
34     return "home";
35 }
```

Spring security is configured for BASIC authentication

WebSecurityConfig.java

[link](#)

```
19  @Override
20  protected void configure(HttpSecurity http) throws Exception {
21      http
22          .csrf().disable()
23          .authorizeRequests()
24          .antMatchers("/", "/home", "/rest/**").permitAll()
25          .antMatchers("/js/**", "/css/**", "/images/**").permitAll()
26          .anyRequest().authenticated()
27          .and()
28          .formLogin()
29          .loginPage("/login")
30          .permitAll()
31          .and()
32          .logout()
33          .permitAll();
34  }
```

chart.js



chart.js is a library that provides various charts, the project renders charts and the data is fetched from Rest API.

charts.html

[link](#)

```
16  <div class="row">
17      <div class="col">
18          <canvas id="piechartContainer" width="200" height="300"></canvas>
19      </div>
20      <div class="col">
21          <canvas id="barchartContainer" width="200" height="300"></canvas>
22      </div>
23  </div>
```

[link](#)

charts.js

```
23 $.getJSON("/rest/pie-data", function(json) {
24     new Chart(document.getElementById("piechartContainer"), {
25         type: 'pie',
26         data: {
27             labels: json[0],
28             datasets: [{
29                 backgroundColor: backgroundColors,
30                 borderColor: borderColors,
31                 borderWidth: 1,
32                 hoverOffset: 4,
33                 data: json[1]
34             }]
35     },
36     options: {
37         title: {
38             display: true,
39             text: 'Pie Chart'
40         },
41         responsive: true,
42         maintainAspectRatio: false,
43         scales: {
44             yAxes: [{
45                 ticks: {
46                     beginAtZero: true
47                 }
48             }]
49         }
50     }
51 });
52 });
```

Thymeleaf



We will use thymeleaf fragments to include menu headers.

home.html

[link](#)

3 <head>

```
4      <div th:replace="fragments/general :: include-frag"/>
5      <script src="js/home.js"></script>
6  </head>
7  <body>
8
9  <div th:replace="fragments/general :: menu-frag"/>
```

Bootstrap 5



We will use the bootstrap 5 library and use the many components it provides.

login.html

[link](#)

```
10  <div class="container">
11      <form method="post" th:action="@{/login}" role="form" class="form-hori
12          <br/>
13          <br/>
14          <h2>Login</h2>
15
16          <div class="mb-3">
17              <label for="exampleInputEmail1" class="form-label">Username</l
18              <input type="text" name="username" class="form-control" id="ex
19              <div id="usernameHelp" class="form-text">Enter ldap username.<
20          </div>
21
22          <div class="mb-3">
23              <label for="exampleInputPassword1" class="form-label">Password
24              <input type="password" name="password" class="form-control" id
25          </div>
26          <button type="submit" class="btn btn-primary" style="width:100%">S
27
28      <br/>
29      <br/>
30      <div th:if="${param.logout}" class="alert alert-success" role="ale
31          You have been logged out
32  </div>
```

```
33         <div th:if="${param.error}" class="alert alert-danger" role="alert">
34             Invalid username and password!
35         </div>
36     </form>
37
38 </div>
```

Setup

Docker

To run the service by pulling the images from docker hub

```
1 docker-compose -f docker/docker-compose.yml up
```

Postgres DB

```
1 docker run -p 5432:5432 --name pg-container -e POSTGRES_PASSWORD=password -
2 docker ps
3 docker exec -it pg-container psql -U postgres -W postgres
4 CREATE USER test WITH PASSWORD 'test@123';
5 CREATE DATABASE "test-db" WITH OWNER "test" ENCODING UTF8 TEMPLATE template
6 grant all PRIVILEGES ON DATABASE "test-db" to test;
```

Dev

Run the project on dev

```
1 ./gradlew bootRun
```

Prod

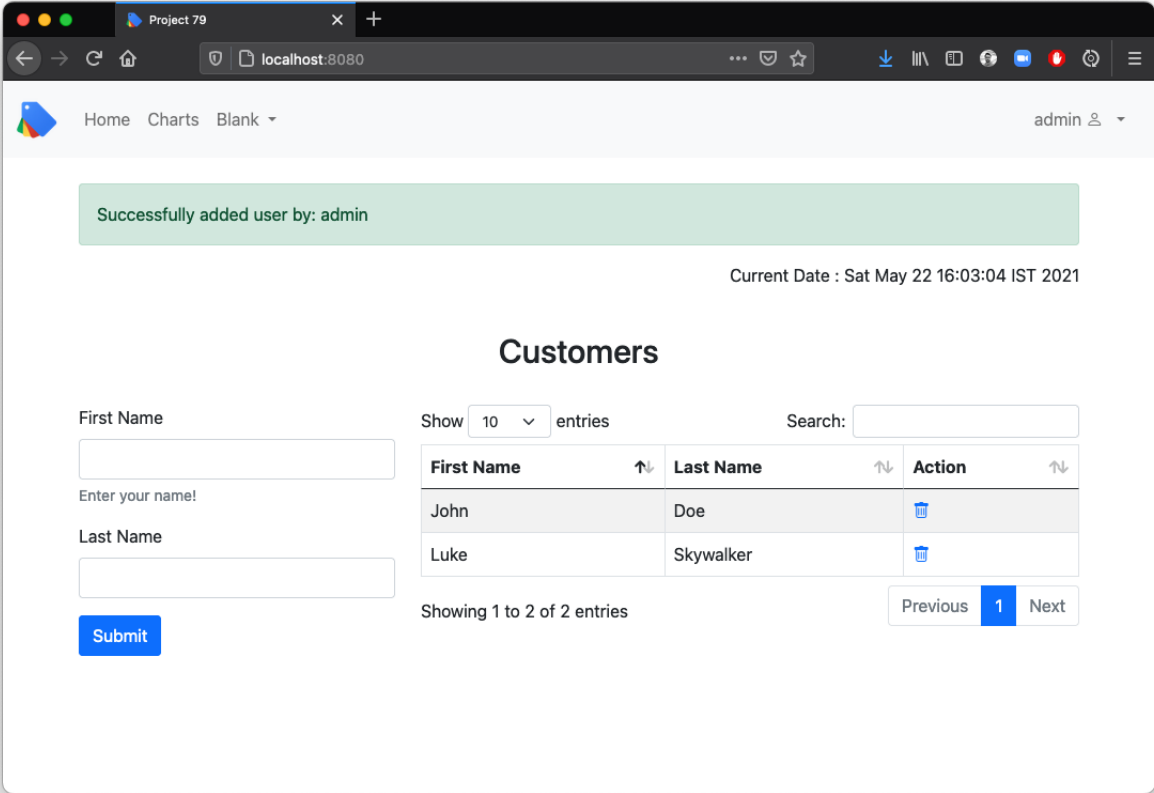
To build the fat jar & run the jar.

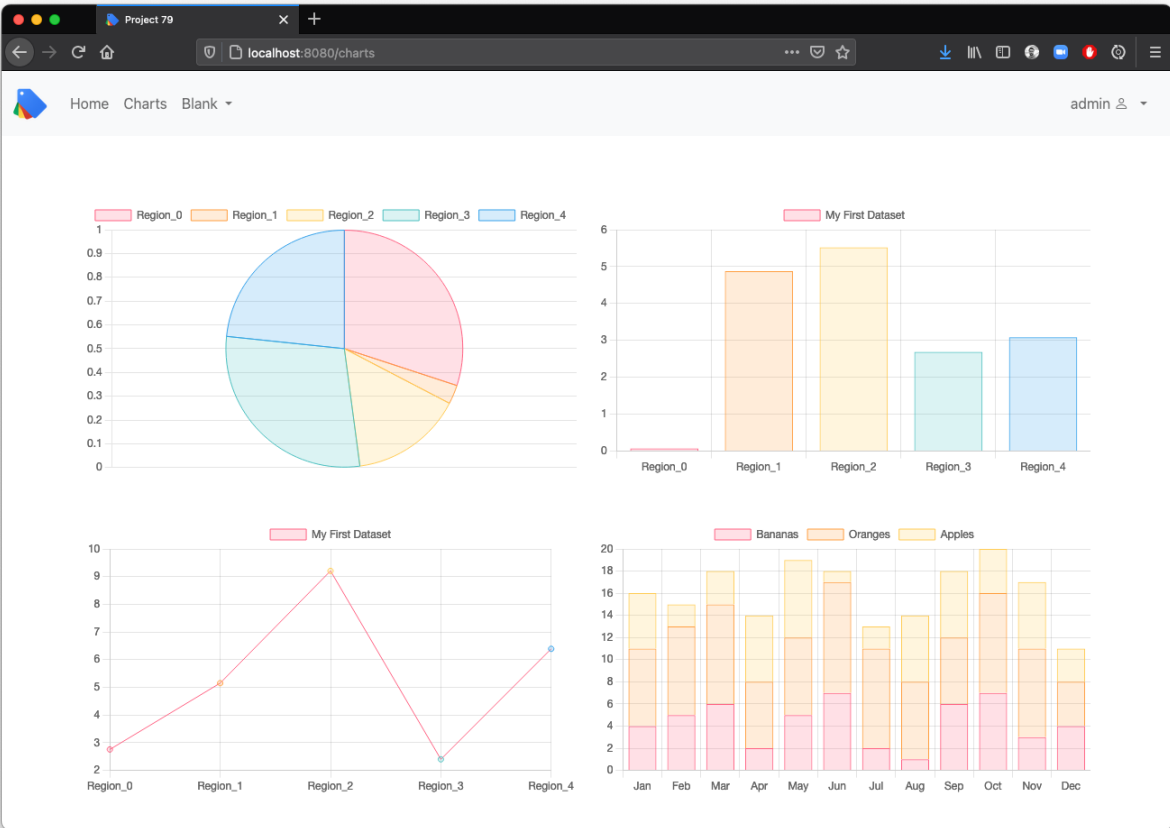
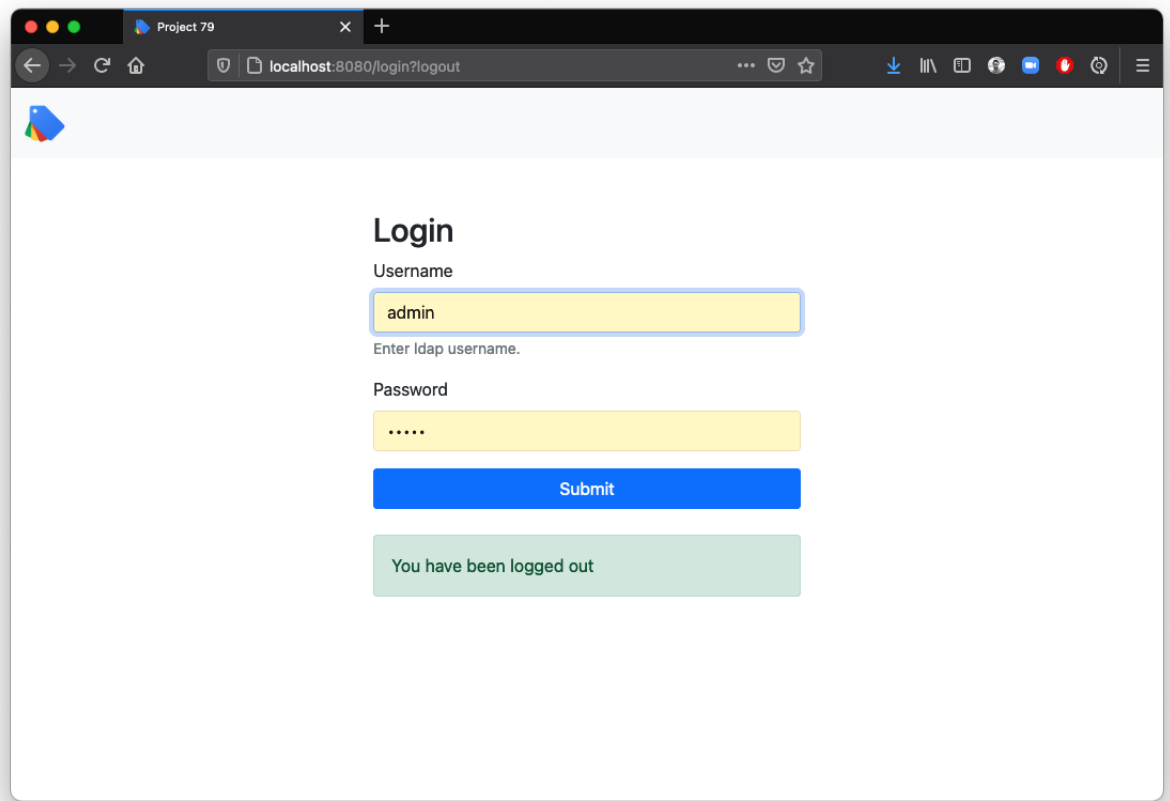
```
1 ./gradlew build
2 java -jar project79-1.0.0.jar
```

<http://localhost:8080/>
user: admin
pwd: admin@123

Screenshots

Here are some screenshots of the web application





References

Bootstarp 5 : <https://getbootstrap.com/>
Chart.js : <https://www.chartjs.org/>

#bootstrap 5 #chart.js #hsqldb #spring.jpa #spring.mvc #spring.security #thymeleaf

Comments Share

NEWER

Spring Boot Web - Angular

OLDER

Java Development in VSCode

What do you think?

0 Responses



Upvote



Love

Comments Community Privacy Policy

1 Login

Favorite Tweet f Share Sort by Best

Start the discussion...

LOG IN WITH

CATEGORIES

- Algorithms
- Angular
- Clarity
- CompletableFuture
- DS
- Design Pattern
- Docker
- Drools

[Dynamic Programming](#)
[ETL](#)
[Functional Programming](#)
[Grafana](#)
[Helm](#)
[Hexo](#)
[JMeter](#)
[JPA](#)
[JWT](#)
[Java](#)
[Kafka](#)
[Kubernetes](#)
[Messaging](#)
[MicroMeter](#)
[ModelMapper](#)
[MongoDB](#)
[Optimistic Locking](#)
[Pipeline](#)
[Prometheus](#)
[Puzzles](#)
[QueryDSL](#)
[RabbitMQ](#)
[RateLimit](#)
[ReactJS](#)
[Redis](#)
[Reports](#)
[Rest](#)
[SSE](#)
[Selenium](#)
[Sleuth](#)
[Spring](#)
[Spring Reactor](#)
[SpringBoot](#)
[Telepresence](#)
[Traefik](#)
[VSCode](#)
[Visualization](#)
[Web](#)
[WebSocket](#)
[Zipkin](#)
[leetcode](#)

TAGS

[Algorithms](#)
[Coding](#)
[CompletableFuture](#)
[Data Structures](#)
[Future](#)

[Interview](#)
[Java](#)
[Puzzles](#)
[QR code](#)
[angular](#)
[angular 11](#)
[apache-superset](#)
[bootstrap 5](#)
[chart.js](#)
[charts](#)
[checkstyle](#)
[clarity](#)
[code coverage](#)
[correto](#)
[data processing](#)
[data-visualization](#)
[datagrid](#)
[design pattern](#)
[docker](#)
[drools](#)
[events](#)
[flash sale](#)
[freemarker](#)
[github](#)
[google-chart](#)
[grafana](#)
[helm](#)
[hexo](#)
[hot swap](#)
[hsql db](#)
[jasper report](#)
[java](#)
[java8](#)
[jenkins](#)
[jib](#)
[jmeter](#)
[jmx](#)
[jpa](#)
[jwt](#)
[kafka](#)
[kafka stream](#)
[kubernetes](#)
[lambda](#)
[micrometer](#)
[modelmapper](#)
[mongodb](#)
[n+1 problem](#)
[open api](#)
[openjdk](#)
[optimistic locking](#)
[pessimistic locking](#)

[pipeline](#)
[postgres](#)
[postgres db](#)
[prometheus](#)
[properties](#)
[querydsl](#)
[r2dbc](#)
[rabbitmq](#)
[rate limit](#)
[reactive mongo](#)
[reactive-jdbc](#)
[reactjs](#)
[reactor](#)
[redis](#)
[rpc](#)
[scatter gather](#)
[selenium](#)
[server sent events](#)
[server-driven](#)
[server-sent-event](#)
[sleuth](#)
[spring](#)
[spring amqp](#)
[spring boot](#)
[spring cloud vault](#)
[spring jpa](#)
[spring mvc](#)
[spring query dsl](#)
[spring security](#)
[spring web](#)
[springboot](#)
[state machine](#)
[superset](#)
[telepresence](#)
[thymeleaf](#)
[ticket booking](#)
[voting system](#)
[vscode](#)
[webflux](#)
[websocket](#)
[xenon](#)
[zipkin](#)

TAG CLOUD

[Algorithms](#) [Coding](#) [CompletableFuture](#) [Data Structures](#) [Future](#) [Interview](#) [Java](#) [Puzzles](#) [QR code](#) [angular](#) [angular 11](#)
[apache-superset](#) [bootstrap 5](#) [chart.js](#) [charts](#) [checkstyle](#) [clarity](#) [code coverage](#) [correto](#) [data processing](#) [data-visualization](#)
[datagrid](#) [design pattern](#) [docker](#) [drools](#) [events](#) [flash sale](#) [freemarker](#) [github](#) [google-chart](#) [grafana](#) [helm](#) [hexo](#) [hot swap](#)

[hsqldb](#) [jasperreport](#) [java](#) [java8](#) [jenkins](#) [jib](#) [jmeter](#) [jmx](#) [jpa](#) [jwt](#) [kafka](#) [kafka-stream](#) [kubernetes](#) [lambda](#) [micrometer](#)
[modelmapper](#) [mongodb](#) [n+1 problem](#) [openapi](#) [openjdk](#) [optimistic locking](#) [pessimistic locking](#) [pipeline](#) [postgres](#)
[postgresdb](#) [prometheus](#) [properties](#) [querydsl](#) [r2dbc](#) [rabbitmq](#) [rate limit](#) [reactive mongo](#) [reactive-jdbc](#) [reactjs](#) [reactor](#) [redis](#)
[rpc](#) [scattergather](#) [selenium](#) [server sent events](#) [server-driven](#) [server-sent-event](#) [sleuth](#) [spring](#) [spring amqp](#) [spring boot](#)
[spring cloud vault](#) [spring jpa](#) [spring mvc](#) [spring query dsl](#) [spring security](#) [spring web](#) [springboot](#) [state machine](#)
[superset](#) [telepresence](#) [thymeleaf](#) [ticket booking](#) [voting system](#) [vscode](#) [webflux](#) [websocket](#) [xenon](#) [zipkin](#)

ARCHIVES

[April 2022](#)
[February 2022](#)
[January 2022](#)
[November 2021](#)
[October 2021](#)
[September 2021](#)
[July 2021](#)
[June 2021](#)
[May 2021](#)
[August 2020](#)
[March 2020](#)
[February 2020](#)
[December 2019](#)
[September 2019](#)
[June 2019](#)
[April 2019](#)
[March 2019](#)
[July 2018](#)

RECENT POSTS

[Spring Boot & Traefik - Rate Limit](#)
[Voting System](#)
[Java Puzzles](#)
[Spring Boot & Apache Kafka](#)
[Grokking the Coding Interview](#)