

SpringSecurity_BankingApp-Service [boot] [devtools]

src/main/java

in:neuron.nitin

in:neuron.nitin.controller

in:neuron.nitin.security.configuration

SecurityConfigApp.java

src/main/resources

static

templates

application.properties

src/test/java

in:neuron.nitin

JRE System Library [JavaSE-17]

Maven Dependencies

src

main

webapp

WEB-INF

pages

home.jsp

test

target

HELP.md

mmw

mmw.cmd

pom.xml

customizing the configuration for endpoints
"/" ->no authentication and authorization

SpringSecurity_BankingApp-Service [boot] [devtools]

src/main/java

in:neuron.nitin

SpringSecurityBankingAppServiceApplication.java

in:neuron.nitin.controller

BankOperationsController.java

src/main/resources

static

templates

application.properties

src/test/java

in:neuron.nitin

JRE System Library [JavaSE-17]

Maven Dependencies

src

main

webapp

WEB-INF

pages

access_denied.jsp

home.jsp

loan.jsp

offers.jsp

show_balance.jsp

test

target

HELP.md

mmw

mmw.cmd

pom.xml

```
@RequestMapping("/bankOperations")  
@PreAuthorize("permitAll()")  
public class BankOperationsController {  
    @GetMapping("/offers")  
    public String offers() {  
        return "offers";  
    }  
    @GetMapping("/balance")  
    public String balance() {  
        return "balance";  
    }  
    @GetMapping("/loan")  
    public String loan() {  
        return "loan";  
    }  
}
```

View Component

OAuth Integration

The diagram illustrates the OAuth integration process. A client sends a request to the HackerRankApp, which displays a 'No signup direct login (GitHub)' button. Clicking this button redirects the user to a 'Sign in to GitHub to continue to HackerRank' page. This page has input fields for 'Username or email address' and 'Password', a 'Forgot password?' link, and a 'Signin' button. The 'Signin' button sends 'credentials' to the 'GitHub Authentication Server'. The server returns a 'token (automatic)' to the client. The client then sends an 'Access the application' request to the 'GitHub Resource Server'. The server returns a 'user-profile' to the client. The 'GitHub Resource Server' is connected to a 'Database'.

SpringOAuthConfiguration [boot] [devtools]

src/main/java

SpringOAuthConfigurationApplication.java

in:neuron.nitin

in:neuron.nitin.rest.controller

WelcomeRestController.java

src/main/resources

static

templates

application.yml

src/test/java

in:neuron.nitin

JRE System Library [JavaSE-17]

Maven Dependencies

src

main

webapp

WEB-INF

pages

access_denied.jsp

home.jsp

loan.jsp

offers.jsp

show_balance.jsp

test

target

HELP.md

mmw

mmw.cmd

pom.xml

```
application.yml  
server:  
  port: 9999  
spring:  
  security:  
    oauth2:  
      client:  
        registration:  
          github:  
            clientId: b5057999f859ff279974  
            clientSecret: 1edbd8d501fc0b16886dd3e3b44c4f20700a0baaf
```