Spring Security:-

Spring Security is a framework which provides various security features like: authentication, authorization to create secure Java Enterprise Application.

It overcomes all the problems that come during creating non spring security applications and manage new server environment for the application.

Spring Security Core Concept:-

Authentication:-

In Spring Security, “authentication” is the process of confirming that a user is who they say they are and that they have the right credentials to log in to a protected resource or to perform a privileged action in an application.

Authorization:-

Authorization is a process by which a server determines if the client has permission to use a resource or access a file. Authorization is usually coupled with authentication so that the server has some concept of who the client is that is requesting access.

Principle:-

The principal *is* the currently logged in user. However, you retrieve it through the security context which is bound to the current thread and as such it's also bound to the current request and its session.

Authorities:-

Authorities are collection of permission. Authenticated users should have some Permission to do some set of work (or) can access some set of resource.

Role:

A role is a group of authorities. Sometimes most of them are using authority and role as same. But there are some differences in that.

MvcConfig.java

```
 package com.example.demo;
import org.springframework.context.annotation.Configuration;
import
org.springframework.web.servlet.config.annotation.ViewControllerRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
@Configuration
public class MvcConfig implements WebMvcConfigurer
{
        public void addViewControllers(ViewControllerRegistry registry) {
                registry.addViewController("/home").setViewName("home");
                registry.addViewController("/").setViewName("home");
                registry.addViewController("/hello").setViewName("hello");
                registry.addViewController("/login").setViewName("login");
        }
}
```

Security2Application.java

```
package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication


public class Security2Application

  {

     public static void main(String[] args)

       {

        SpringApplication.run(Security2Application.class, args);

       }

  }
```

webSecurity.java

```
package com.example.demo;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import
org.springframework.security.config.annotation.web.builders.HttpSecurity;

import
org.springframework.security.config.annotation.web.configuration.EnableWebS
ecurity;

import org.springframework.security.core.userdetails.User;

import org.springframework.security.core.userdetails.UserDetails;

import org.springframework.security.core.userdetails.UserDetailsService;

import
org.springframework.security.provisioning.InMemoryUserDetailsManager;

import org.springframework.security.web.SecurityFilterChain;

@Configuration

@EnableWebSecurity

public class webSecurity {

@Bean

public SecurityFilterChain securityFilterChain(HttpSecurity http) throws
Exception {

        http.authorizeHttpRequests((requests) -> requests

.requestMatchers("/", "/home").permitAll().anyRequest().authenticated())

.formLogin((form) ->

form.loginPage("/login")

.permitAll())

.logout((logout) -> logout.permitAll());

return http.build();

        }
        @Bean
        public UserDetailsService userDetailsService() {
                UserDetails user1 =User.withDefaultPasswordEncoder()
                                .username("user1")
                                .password("password1")
                                .roles("USER")
                                .build();

                UserDetails user2 = User.withDefaultPasswordEncoder()
                                        .username("user2")
                                        .password("password2")
                                        .roles("ADMIN")
                                        .build();

                return new InMemoryUserDetailsManager(user1,user2);
        }
}
```

Hello.html

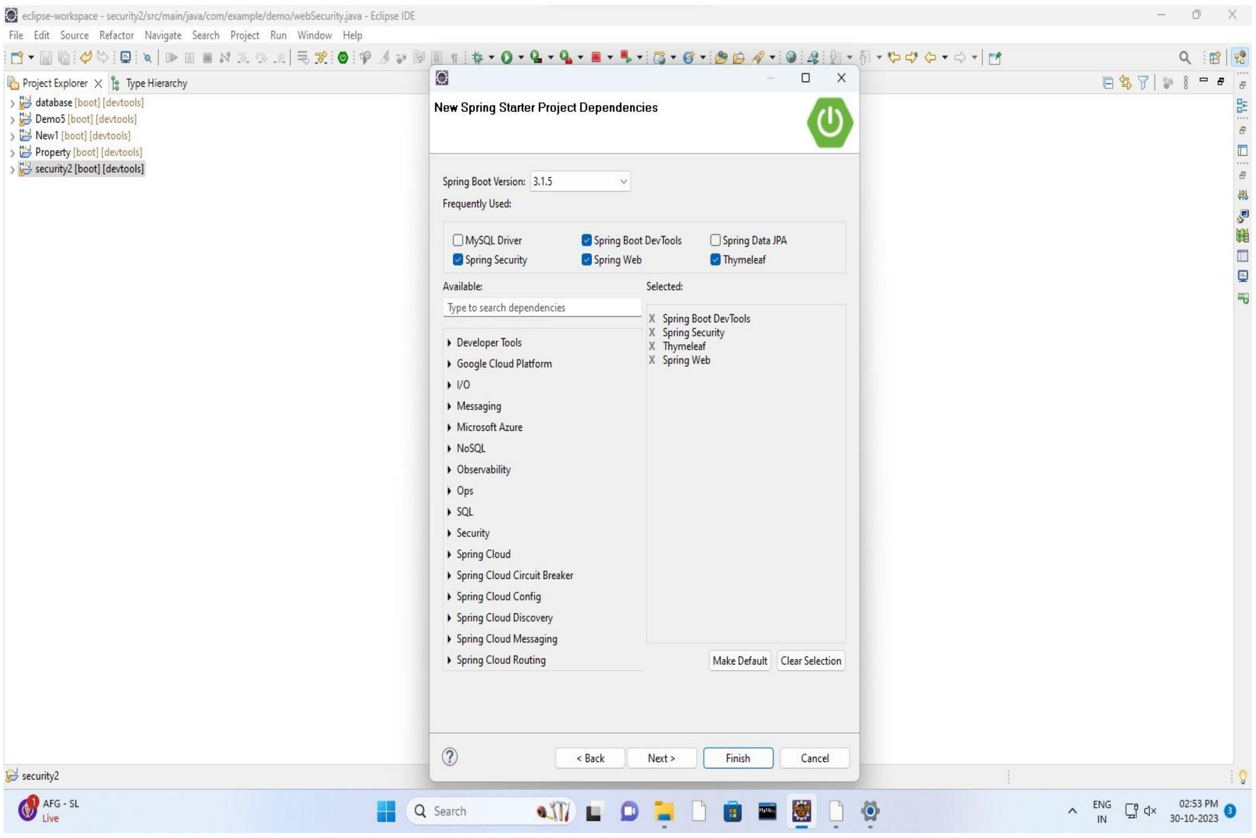
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:th="https://www.thymeleaf.org"
     xmlns:sec="https://www.thymeleaf.org/thymeleaf-extras-springsecurity6">
   <head> <title>Hello World!</title>
   </head>
   <body>
      <h1 th:inline="text">Hello <span th:remove="tag"
sec:authentication="name">thymeleaf</span>!</h1>
      <form th:action="@{/logout}" method="post">
          <input type="submit" value="Sign Out"/>
      </form>
   </body>
</html>
```

Home.html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:th="https://www.thymeleaf.org">
   <head>
      <title>Spring Security Example</title>
   </head>
   <body>
      <h1>Welcome!</h1>
      <p>Click <a th:href="@{/hello}">here</a> to see a greeting.</p>
   </body>
</html>
```

login.html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:th="https://www.thymeleaf.org">
    <head>
        <title>Spring Security Example </title>
    </head>
    <body>
        <div th:if="${param.error}">
            Invalid username and password.
        </div>
        <div th:if="${param.logout}">
            You have been logged out.
        </div>
        <form th:action="@{/login}" method="post">
            <div><label> User Name : <input type="text" name="username"/>
</label></div>
            <div><label> Password: <input type="password" name="password"/>
</label></div>
            <div><input type="submit" value="Sign In"/></div>
        </form>
    </body>
</html>
```


eclipse-workspace - security2/src/main/java/com/example/demo/webSecurity.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Type Hierarchy
security2
< Back
Next >
Finish
Cancel
Make Default
Clear Selection
Search
ENG IN
02:53 PM 30-10-2023
AFG - SL Live

```
> database [boot] [devtools]
> Demo5 [boot] [devtools]
> New1 [boot] [devtools]
> Property [boot] [devtools]
> security2 [boot] [devtools]
```

New Spring Starter Project Dependencies

Spring Boot Version: 3.1.5

Frequently Used:

☐ MySQL Driver	☑ Spring Boot DevTools	☐ Spring Data JPA
☑ Spring Security	☑ Spring Web	☑ Thymeleaf

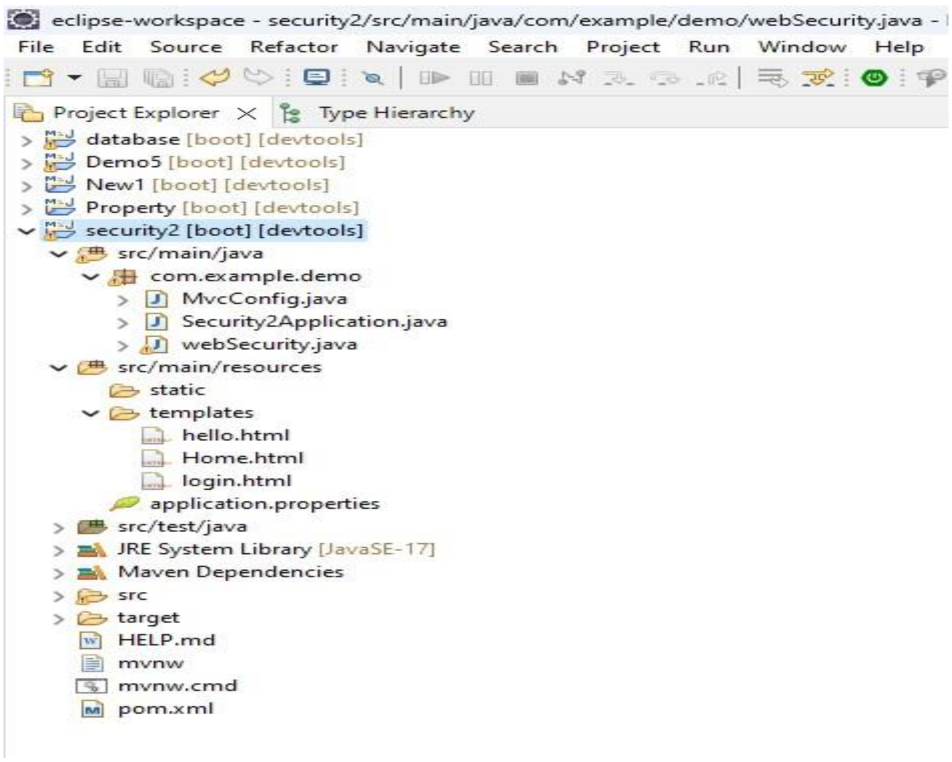
Available:

Type to search dependencies

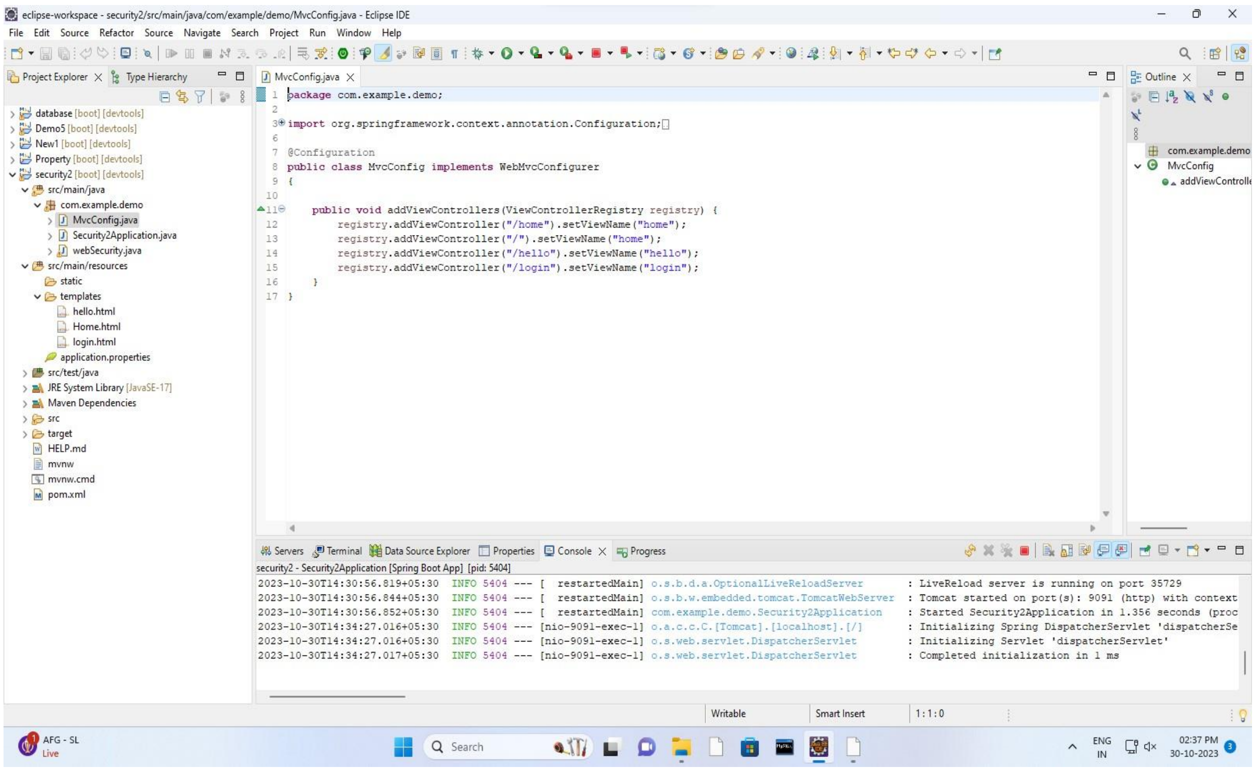
- Developer Tools
- Google Cloud Platform
- I/O
- Messaging
- Microsoft Azure
- NoSQL
- Observability
- Ops
- SQL
- Security
- Spring Cloud
- Spring Cloud Circuit Breaker
- Spring Cloud Config
- Spring Cloud Discovery
- Spring Cloud Messaging
- Spring Cloud Routing

Selected:

X Spring Boot DevTools
X Spring Security
X Thymeleaf
X Spring Web


eclipse-workspace - security2/src/main/java/com/example/demo/webSecurity.java -
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Type Hierarchy

```
> database [boot] [devtools]
> Demo5 [boot] [devtools]
> New1 [boot] [devtools]
> Property [boot] [devtools]
v security2 [boot] [devtools]
    v src/main/java
        v com.example.demo
            > MvcConfig.java
            > Security2Application.java
            > webSecurity.java
    v src/main/resources
            static
        v templates
                hello.html
                Home.html
                login.html
            application.properties
    > src/test/java
    > JRE System Library [JavaSE-17]
    > Maven Dependencies
    > src
    > target
      HELP.md
      mvnw
      mvnw.cmd
      pom.xml
```

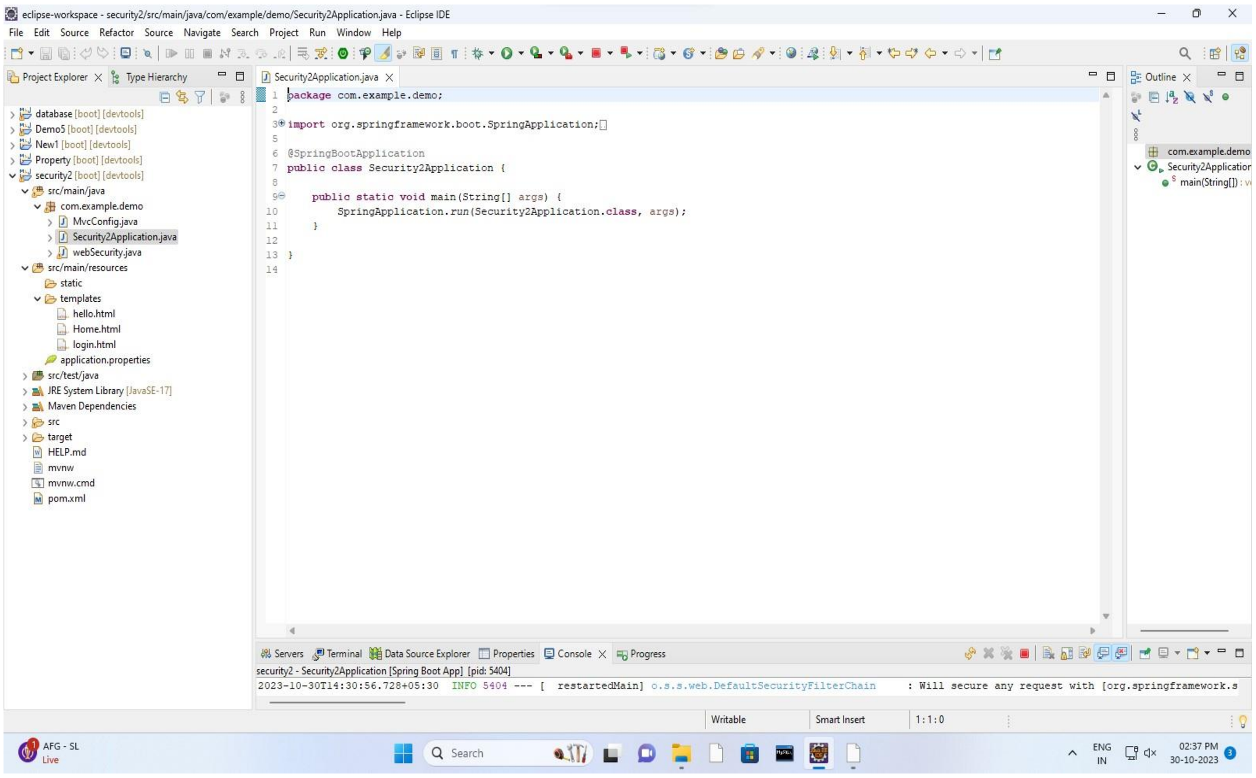


eclipse-workspace - security2/src/main/java/com/example/demo/MvcConfig.java - Eclipse IDE

```java
package com.example.demo;

import org.springframework.context.annotation.Configuration;

@Configuration
public class MvcConfig implements WebMvcConfigurer
{

    public void addViewControllers(ViewControllerRegistry registry) {
        registry.addViewController("/home").setViewName("home");
        registry.addViewController("/").setViewName("home");
        registry.addViewController("/hello").setViewName("hello");
        registry.addViewController("/login").setViewName("login");
    }
}
```

security2 - Security2Application [Spring Boot App] [pid: 5404]

```
2023-10-30T14:30:56.819+05:30  INFO 5404 --- [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on port 35729
2023-10-30T14:30:56.844+05:30  INFO 5404 --- [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 9091 (http) with context
2023-10-30T14:30:56.852+05:30  INFO 5404 --- [  restartedMain] com.example.demo.Security2Application    : Started Security2Application in 1.356 seconds (proc
2023-10-30T14:34:27.016+05:30  INFO 5404 --- [nio-9091-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherSe
2023-10-30T14:34:27.016+05:30  INFO 5404 --- [nio-9091-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2023-10-30T14:34:27.017+05:30  INFO 5404 --- [nio-9091-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 1 ms
```

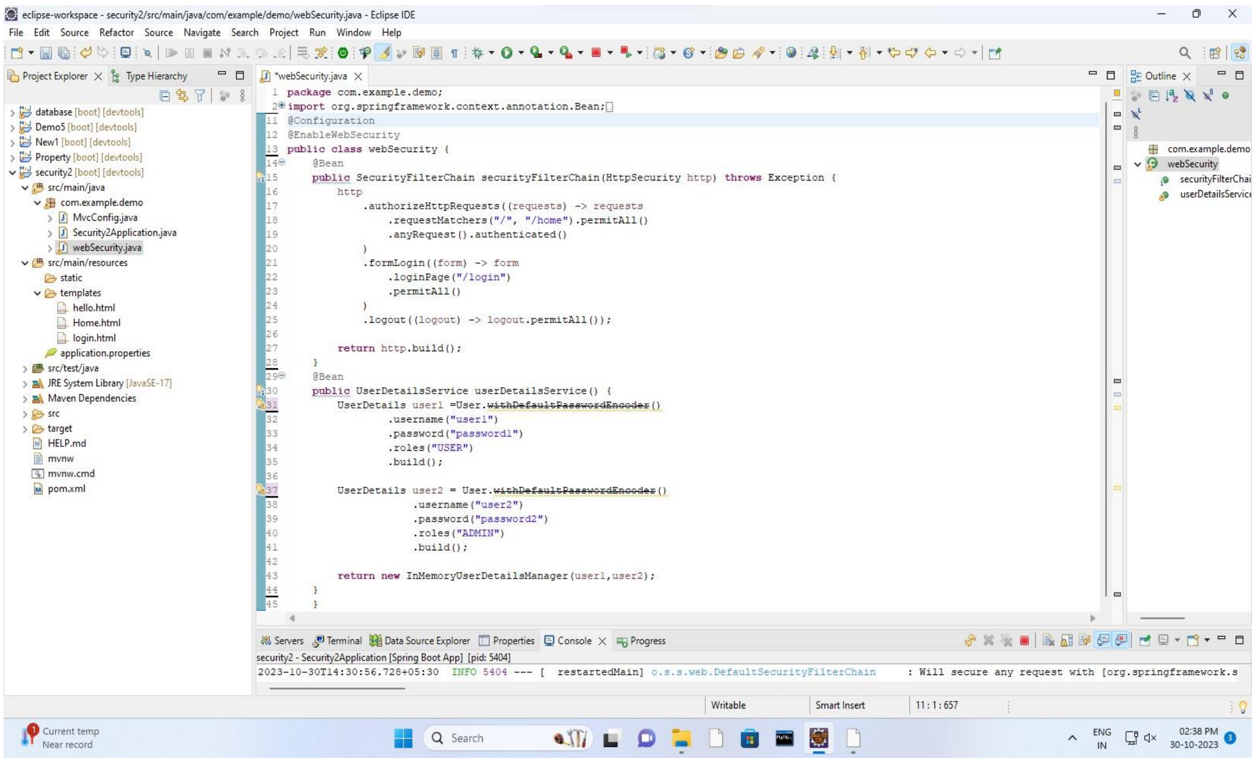


eclipse-workspace - security2/src/main/java/com/example/demo/Security2Application.java - Eclipse IDE

```java
package com.example.demo;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class Security2Application {

    public static void main(String[] args) {
        SpringApplication.run(Security2Application.class, args);
    }

}
```

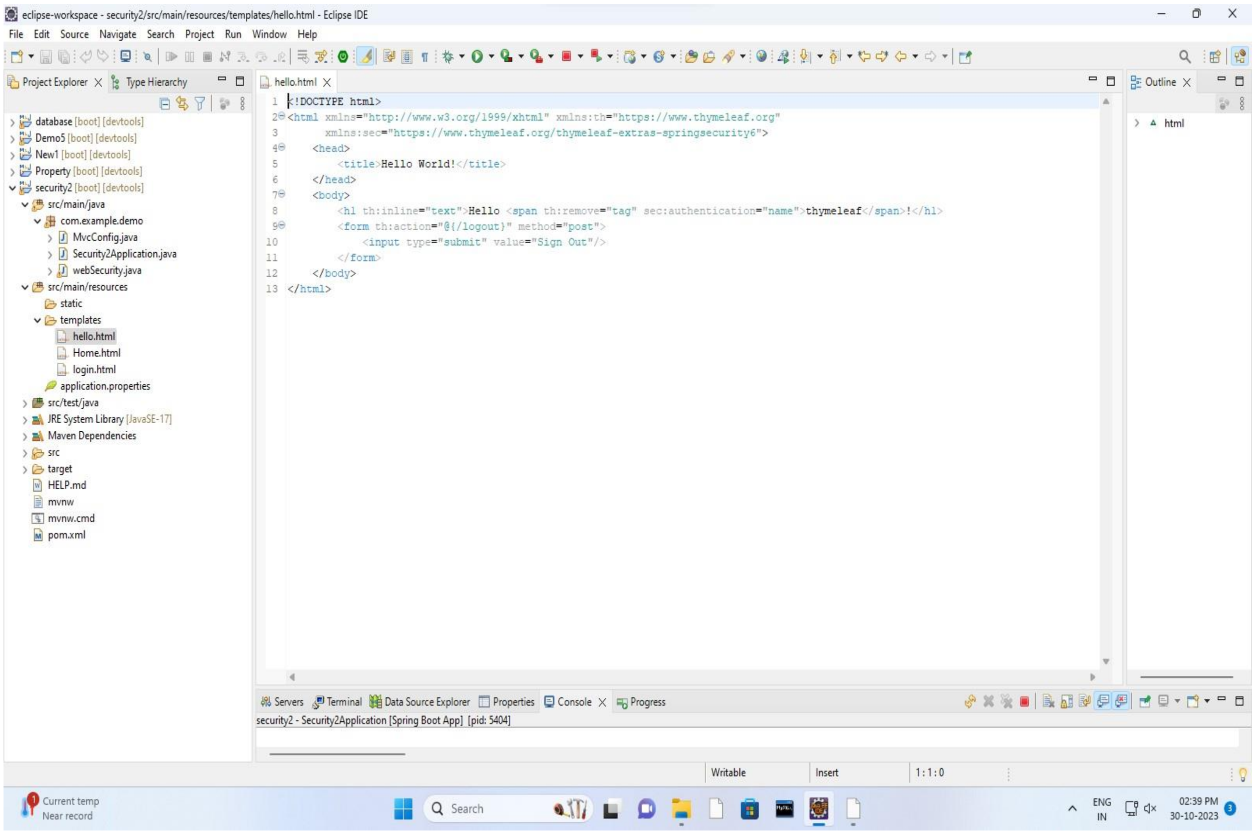
security2 - Security2Application [Spring Boot App] [pid: 5404]

```
2023-10-30T14:30:56.728+05:30  INFO 5404 --- [  restartedMain] o.s.s.web.DefaultSecurityFilterChain     : Will secure any request with [org.springframework.s
```



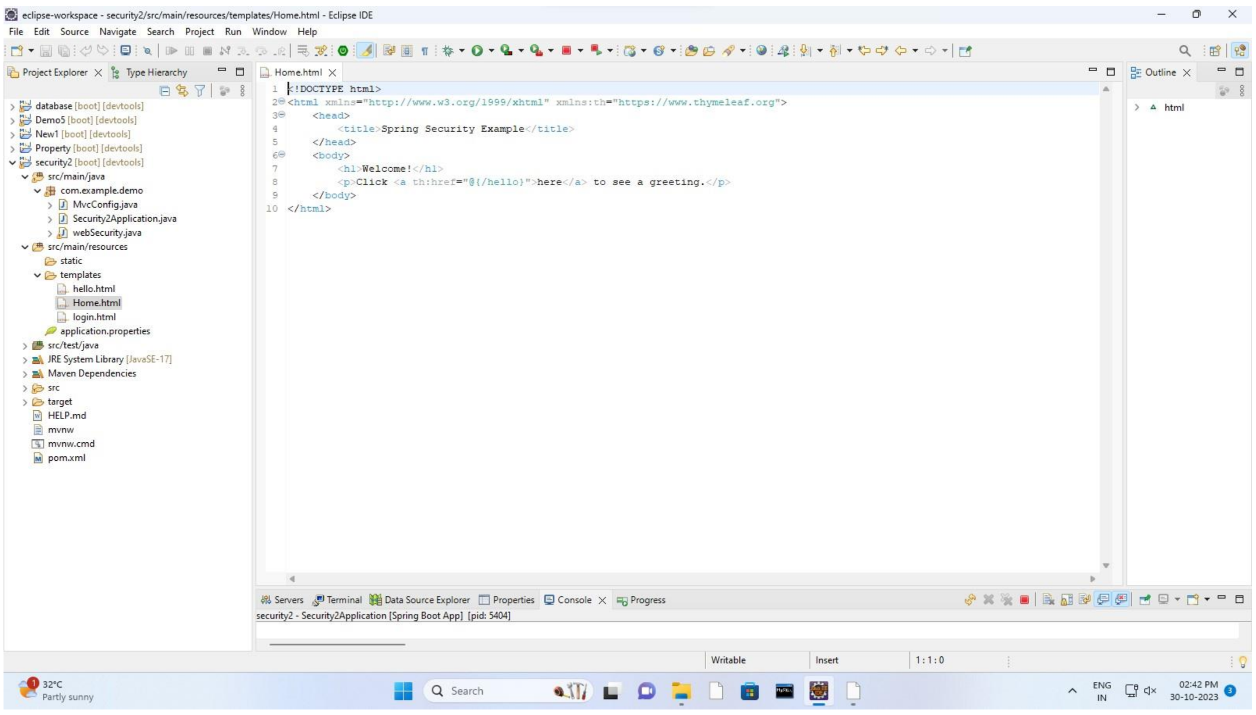
```
package com.example.demo;
import org.springframework.context.annotation.Bean;
@Configuration
@EnableWebSecurity
public class webSecurity {
    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
        http
            .authorizeHttpRequests((requests) -> requests
                .requestMatchers("/", "/home").permitAll()
                .anyRequest().authenticated()
            )
            .formLogin((form) -> form
                .loginPage("/login")
                .permitAll()
            )
            .logout((logout) -> logout.permitAll());

        return http.build();
    }
    @Bean
    public UserDetailsService userDetailsService() {
        UserDetails user1 =User.~~withDefaultPasswordEncoder~~()
                .username("user1")
                .password("password1")
                .roles("USER")
                .build();

        UserDetails user2 = User.~~withDefaultPasswordEncoder~~()
                    .username("user2")
                    .password("password2")
                    .roles("ADMIN")
                    .build();

        return new InMemoryUserDetailsManager(user1,user2);
    }
    }
```

```
security2 - Security2Application [Spring Boot App] [pid: 5404]
2023-10-30T14:30:56.728+05:30  INFO 5404 --- [  restartedMain] o.s.s.web.DefaultSecurityFilterChain     : Will secure any request with [org.springframework.s
```



```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="https://www.thymeleaf.org"
      xmlns:sec="https://www.thymeleaf.org/thymeleaf-extras-springsecurity6">
    <head>
        <title>Hello World!</title>
    </head>
    <body>
        <h1 th:inline="text">Hello <span th:remove="tag" sec:authentication="name">thymeleaf</span>!</h1>
        <form th:action="@{/logout}" method="post">
            <input type="submit" value="Sign Out"/>
        </form>
    </body>
</html>
```

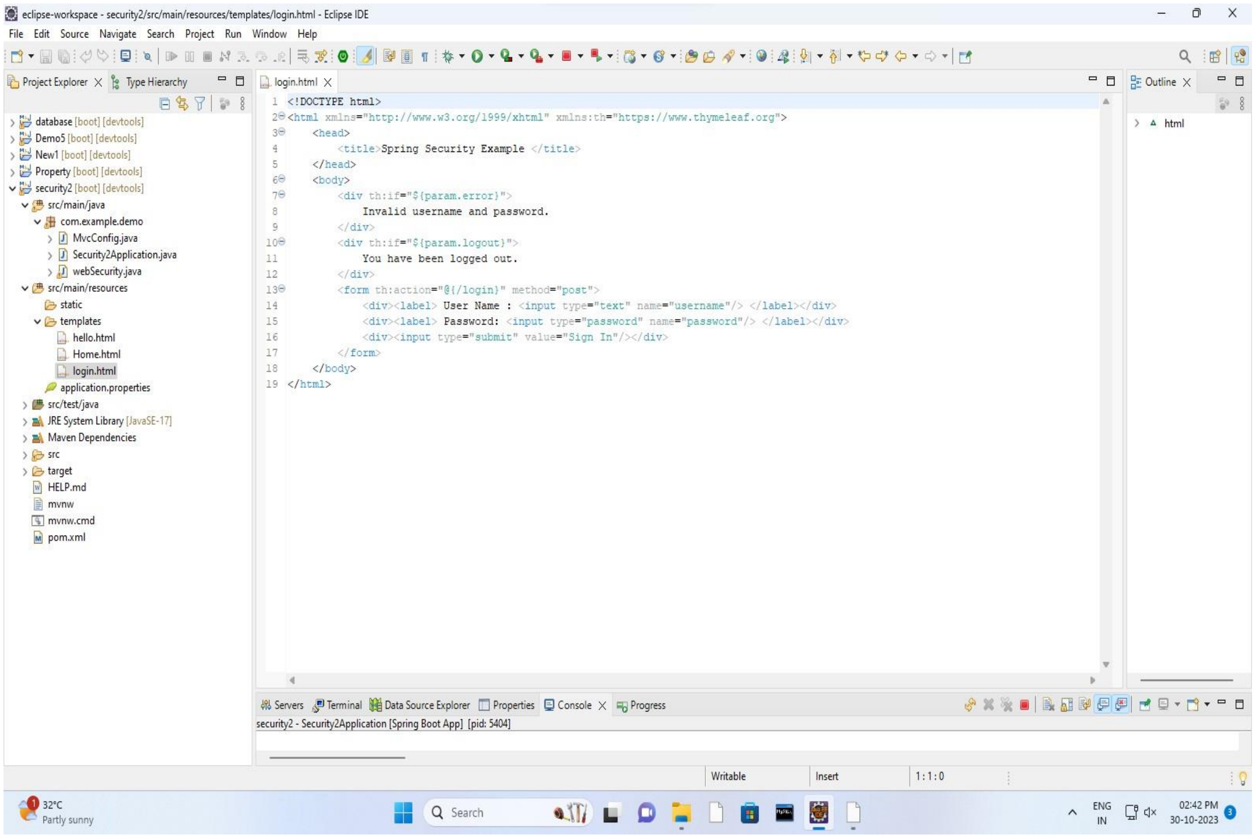
```
security2 - Security2Application [Spring Boot App] [pid: 5404]
```



eclipse-workspace - security2/src/main/resources/templates/Home.html - Eclipse IDE

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="https://www.thymeleaf.org">
    <head>
        <title>Spring Security Example</title>
    </head>
    <body>
        <h1>Welcome!</h1>
        <p>Click <a th:href="@{/hello}">here</a> to see a greeting.</p>
    </body>
</html>
```

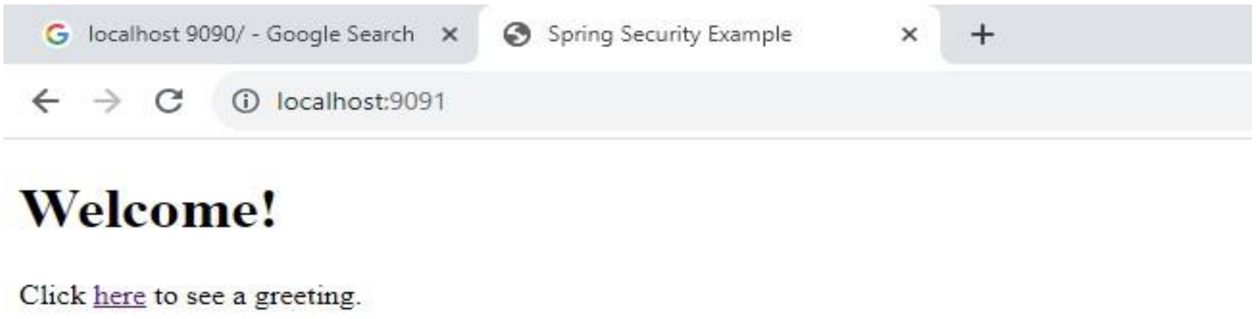
security2 - Security2Application [Spring Boot App] [pid: 5404]



eclipse-workspace - security2/src/main/resources/templates/login.html - Eclipse IDE

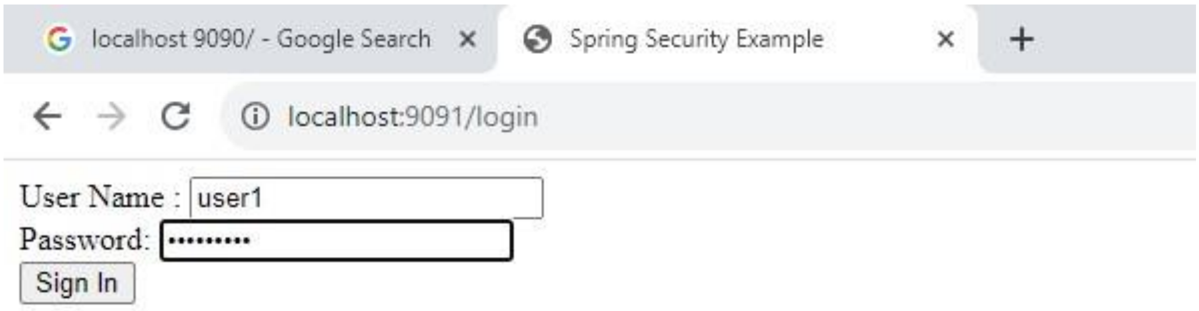
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="https://www.thymeleaf.org">
    <head>
        <title>Spring Security Example </title>
    </head>
    <body>
        <div th:if="${param.error}">
            Invalid username and password.
        </div>
        <div th:if="${param.logout}">
            You have been logged out.
        </div>
        <form th:action="@{/login}" method="post">
            <div><label> User Name : <input type="text" name="username"/> </label></div>
            <div><label> Password: <input type="password" name="password"/> </label></div>
            <div><input type="submit" value="Sign In"/></div>
        </form>
    </body>
</html>
```

security2 - Security2Application [Spring Boot App] [pid: 5404]

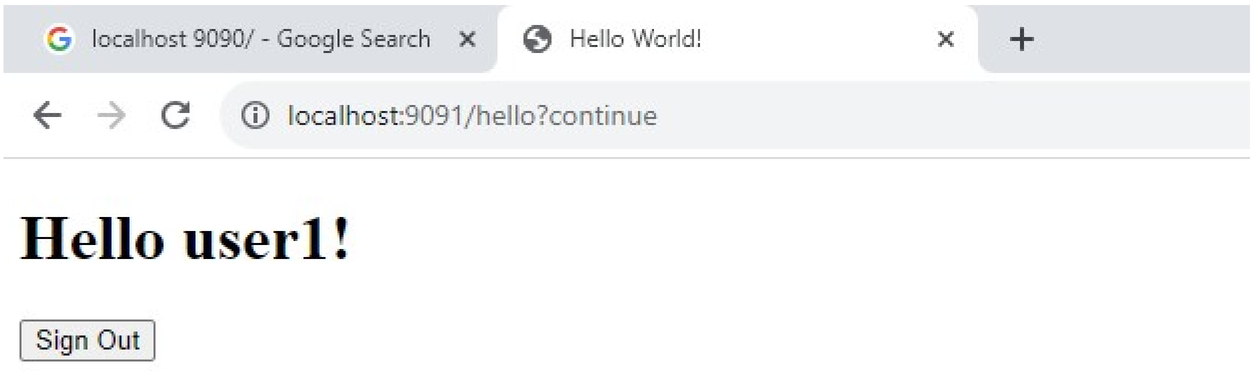


Welcome!

Click here to see a greeting.



User Name : user1
Password: ••••••••••
Sign In


localhost 9090/ - Google Search
Hello World!
localhost:9091/hello?continue
Sign Out

Hello user1!