

Steps:

1. Create a database with table - user
2. Create Spring starter project with dependencies:
 - a. Web
 - b. Spring security
 - c. Mysql driver
3. Create a controller for accessing the pages - Rest Controller

@Enable web security annotation

4. Create Security Configuration which extends websecurityconfigureradapter
 - a. Configure two methods by overriding
 - b. Autowire the UserDetailsService instance
 - c. Create password Encoder with **@Bean annotation**
5. Implement MyUserDetailsService class which implements UserDetailsService
6. Implement MyUserDetails class which implements UserDetails
 - a. Select methods to override
 - b. Create a constructor which maps the user details as fields.
7. Return MyUserDetails in the MyUserDetailsService

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> use springsecurity;
```

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed

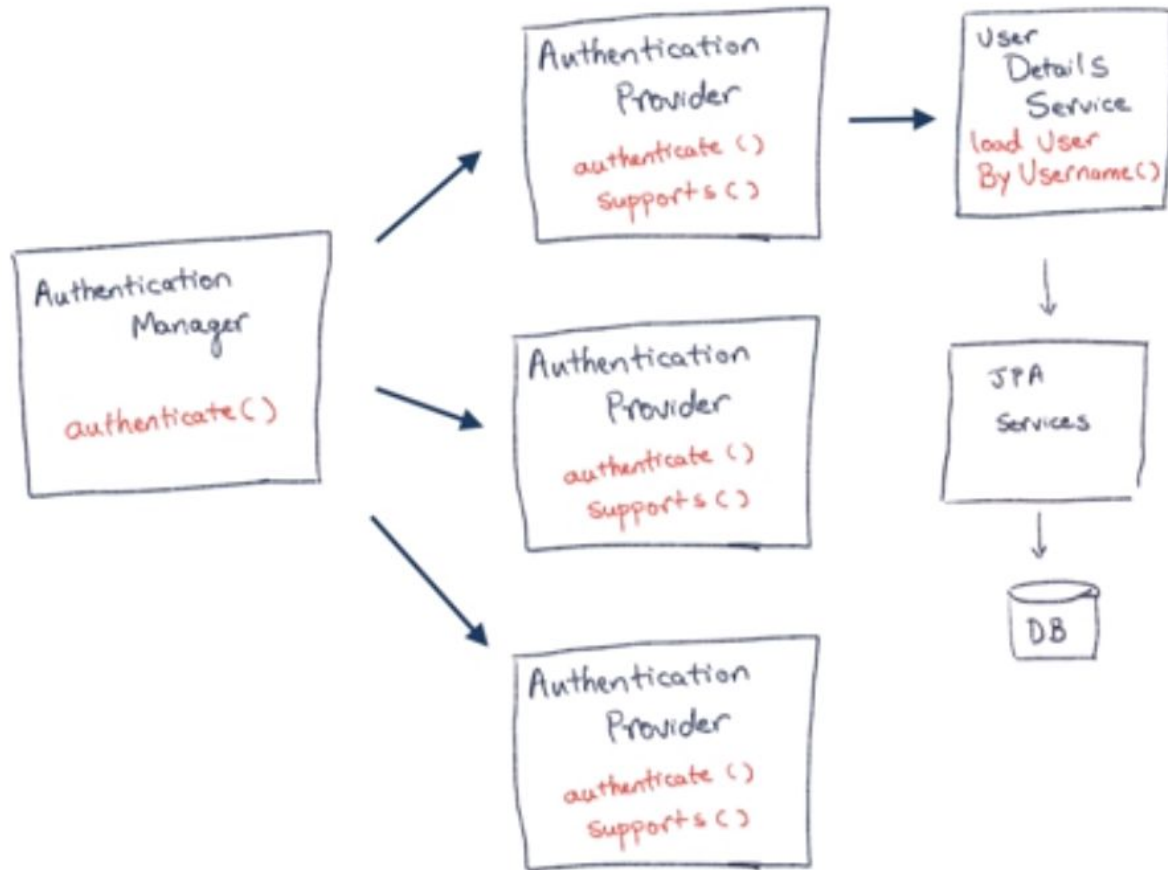
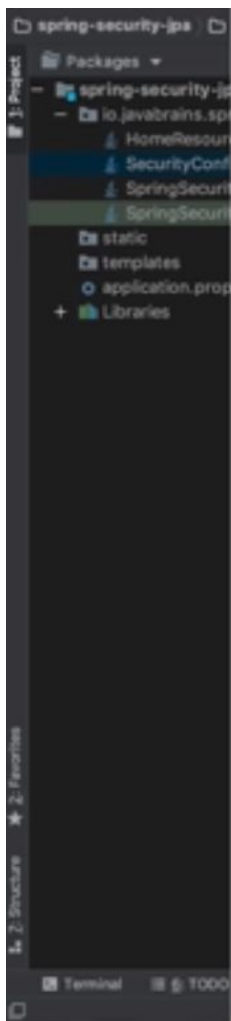
```
mysql> select * from user;
```

id	active	password	roles	user_name
1		pass	ROLE_USER	user

1 row in set (0.00 sec)

```
mysql> █
```

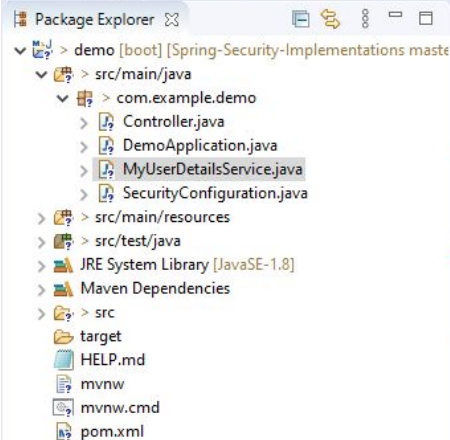




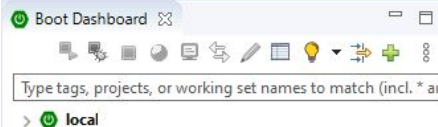
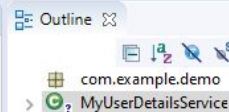
The screenshot shows the Spring Tool Suite 4 IDE with the following components:

- Package Explorer:** Shows the project structure. The selected file is `SecurityConfiguration.java` in the `com.example.demo` package.
- Outline:** Shows the class hierarchy. The selected class is `SecurityConfiguration` in the `com.example.demo` package.
- Code Editor:** Displays the `SecurityConfiguration.java` file. The code is as follows:

```
1 package com.example.demo;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
5 import org.springframework.security.config.annotation.web.builders.HttpSecurity;
6 import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
7 import org.springframework.security.core.userdetails.UserDetailsService;
8
9 public class SecurityConfiguration extends WebSecurityConfigurerAdapter{
10
11     @Autowired
12     UserDetailsService userDetailsService;
13
14     @Override
15     protected void configure(AuthenticationManagerBuilder auth) throws Exception {
16         // TODO Auto-generated method stub
17         auth.userDetailsService(userDetailsService);
18     }
19
20     @Override
21     protected void configure(HttpSecurity http) throws Exception {
22
23         http.authorizeRequests()
24             .antMatchers("/user").hasAnyRole("USER", "ADMIN")
25             .antMatchers("/admin").hasRole("ADMIN")
26             .antMatchers("/").permitAll()
27             .and().formLogin();
28     }
29 }
```
- Problems:** Shows 0 items.



```
SecurityConfiguration.java Controller.java MyUserDetailsService.java
1 package com.example.demo;
2
3 import org.springframework.security.core.userdetails.UserDetails;
4 import org.springframework.security.core.userdetails.UserDetailsService;
5 import org.springframework.security.core.userdetails.UsernameNotFoundException;
6
7 public class MyUserDetailsService implements UserDetailsService{
8
9     @Override
10     public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
11         // TODO Auto-generated method stub
12         return null;
13     }
14
15 }
16
```



Problems @ Javadoc Declaration

0 items

Description	Resource	Path	Location	Type

Writable

Smart Insert

14:1:483



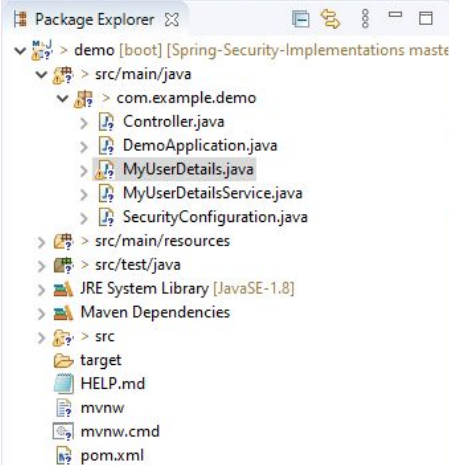
Type here to search



18:32

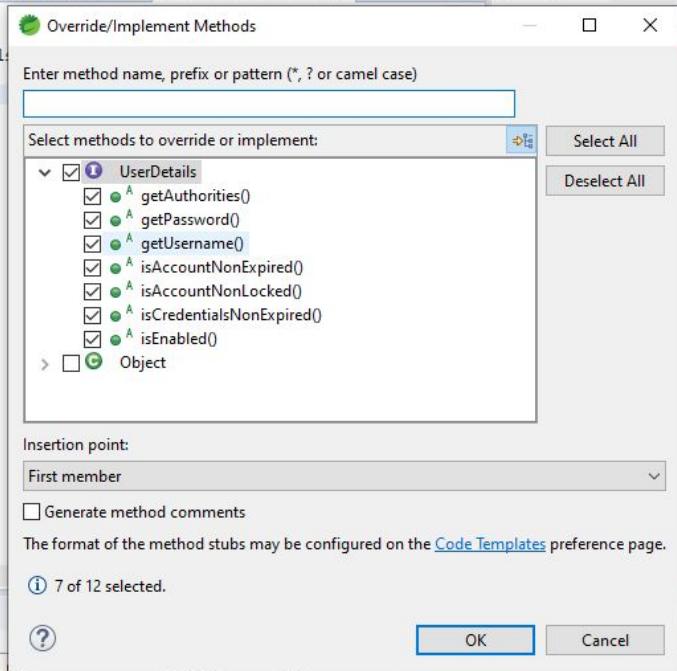
23-06-2020





SecurityConfiguration.java Controller.java *MyUserDetailsService.java *MyUserDetails.java

```
1 package com.example.demo;  
2  
3 import org.springframework.security.core.userdetails;  
4  
5 public class MyUserDetails implements UserDetails{  
6  
7  
8  
9  
10
```



Problems Javadoc Declaration

0 errors, 1 warning, 0 others

Description

> Warnings (1 item)

Resource

Writable

Smart Insert

5: 51: 148



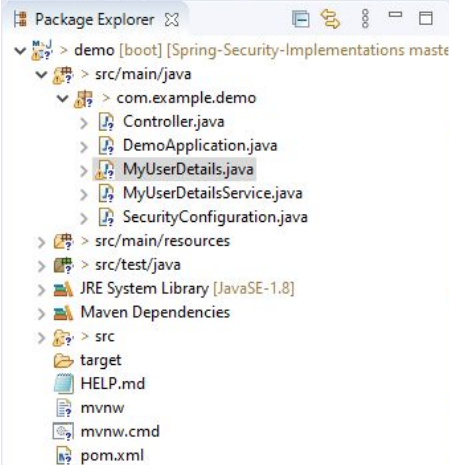
Type here to search



18:37

23-06-2020





Boot Dashboard

Type tags, projects, or working set names to match (incl. *a)

> local

```
SecurityConfiguration.java Controller.java *MyUserDetailsService.java MyUserDetails.java
4
5 import org.springframework.security.core.GrantedAuthority;
6 import org.springframework.security.core.userdetails.UserDetails;
7
8 public class MyUserDetails implements UserDetails{
9
10     public String username;
11
12     //Empty constructor
13     public MyUserDetails() {
14
15     }
16
17     //Parametrised Constructor
18     public MyUserDetails(String username) {
19         super();
20         this.username = username;
21     }
22
23     @Override
24     public Collection<? extends GrantedAuthority> getAuthorities() {
25         // TODO Auto-generated method stub
26         return null;
27     }
28
29     @Override
30     public String getPassword() {
31         // TODO Auto-generated method stub
```

Outline

```
com.example.demo
└─ MyUserDetails
   ├── username : String
   ├── MyUserDetails()
   ├── MyUserDetails(String)
   ├── getAuthorities() : Collection<? ext
   ├── getPassword() : String
   ├── getUsername() : String
   ├── isAccountNonExpired() : boolean
   ├── isAccountNonLocked() : boolean
   ├── isCredentialsNonExpired() : boolea
   └── isEnabled() : boolean
```

Problems @ Javadoc Declaration

0 errors, 1 warning, 0 others

Description	Resource	Path	Location	Type
> ⚠ Warnings (1 item)				

Writable

Smart Insert

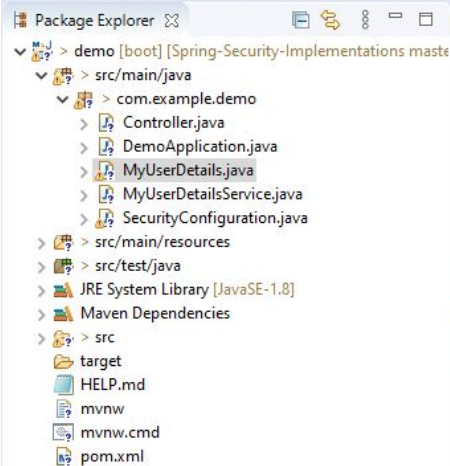
17:31:360

Spring Boot + Spring Security with JPA authentication and MySQL from scratch - Java Brains

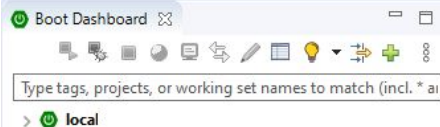
Project
- spring-security-jpa
 - io.javabrainz.springsecurityjpa
 - HomeResource
 - MyUserDetails
 - MyUserDetailsService
 - SecurityConfiguration
 - SpringSecurityJpaApplication
 - SpringSecurityJpaApplicationTests
 - static
 - templates
 - application.properties
 + Libraries

```
9
10 public class MyUserDetails implements UserDetails {
11
12     private String userName;
13
14     @
15     public MyUserDetails(String userName) {
16         this.userName = userName;
17     }
18
19     @
20     public MyUserDetails() {
21     }
22
23     @Override
24     public Collection<? extends GrantedAuthority> getAuthorities() {
25         return Arrays.asList(new SimpleGrantedAuthority("ROLE_USER"));
26     }
27
28     @Override
29     public String getPassword() {
30         return "pass";
31     }
32
33     @Override
34     public String getUsername() {
```





```
SecurityConfiguration.java Controller.java MyUserDetailsService.java MyUserDetails.java
1 package com.example.demo;
2
3 import org.springframework.security.core.userdetails.UserDetails;
4 import org.springframework.security.core.userdetails.UserDetailsService;
5 import org.springframework.security.core.userdetails.UsernameNotFoundException;
6 import org.springframework.stereotype.Service;
7
8
9 @Service
10 public class MyUserDetailsService implements UserDetailsService{
11
12     @Override
13     public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
14         // TODO Auto-generated method stub
15         return new MyUserDetails(username);
16     }
17
18 }
19
```



Problems @ Javadoc Declaration

0 errors, 4 warnings, 0 others

Description	Resource	Path	Location	Type
> ⚠ Warnings (4 items)				

Writable

Smart Insert

15:42:558



Type here to search



18:54

23-06-2020



Spring Boot + Spring Security with JPA authentication and MySQL from scratch - Java Brains

javabrainz

Project
- spring-security-jpa
- io.javabrainz.springsecurityjpa
- HomeResource
- MyUserDetails
- MyUserDetailsService
- SecurityConfiguration
- SpringSecurityJpaApplication
- SpringSecurityJpaApplicationTests
- static
- templates
- application.properties
+ Libraries

```
@EnableWebSecurity
public class SecurityConfiguration extends WebSecurityConfigurerAdapter {

    @Autowired
    UserDetailsService userDetailsService;

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {
        auth.userDetailsService(userDetailsService);
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests()
            .antMatchers("/admin").hasRole("ADMIN")
            .antMatchers("/user").hasAnyRole("ADMIN", "USER")
            .antMatchers("/").permitAll()
            .and().formLogin();
    }

    @Bean
    public PasswordEncoder getPasswordEncoder() { return NoOpPasswordEncoder.getInstance(); }
```

16:37 / 37:44

Build completed successfully in 1 s 146 ms (a minute ago)



