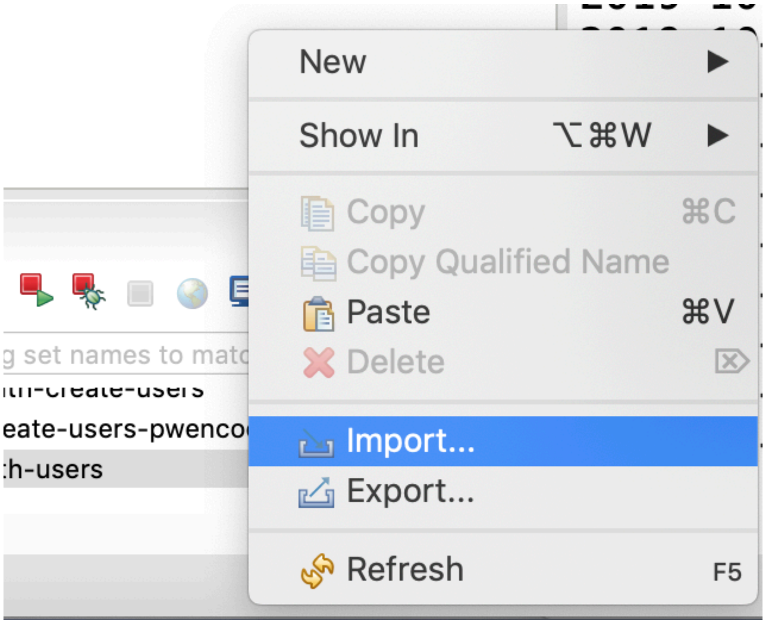
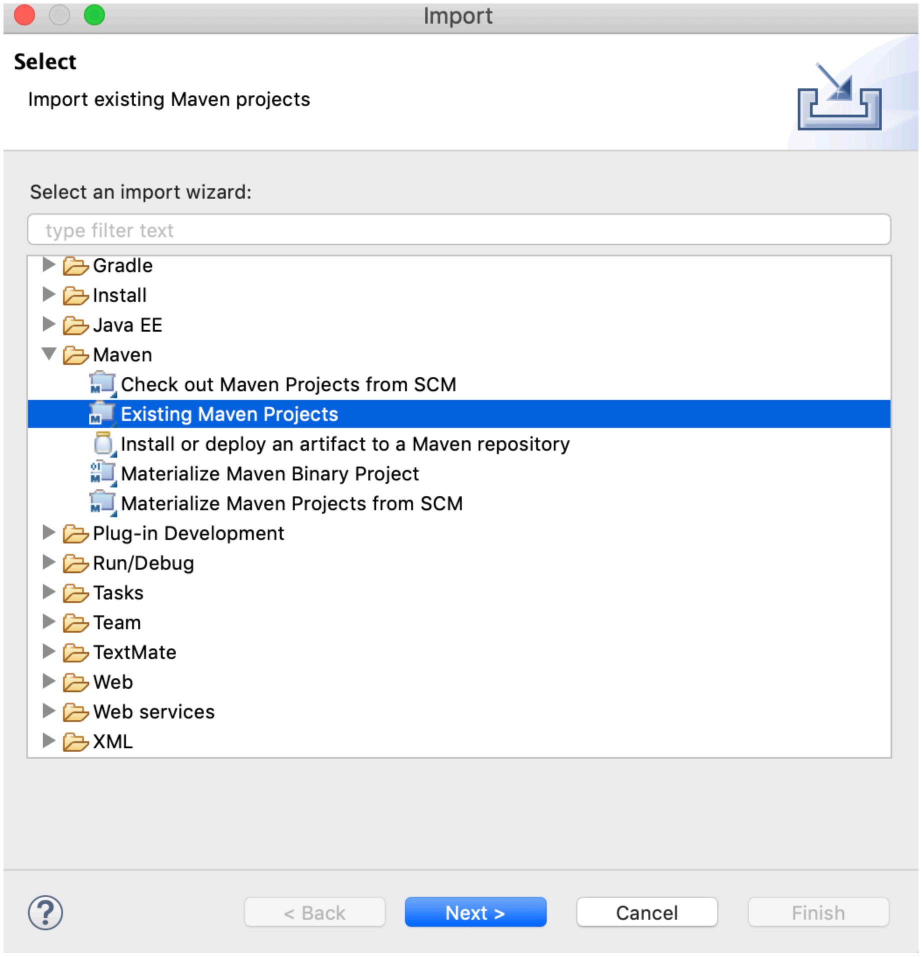
Integrate Spring Security with Spring Boot Application.

1. Import the project ***“demo–boot-withoutsecurity”*** example as a **Maven Project**



2. Select Existing Maven Projects option



Import

Select

Import existing Maven projects

Select an import wizard:

3. Click on Next

4. Click on Browse and select the project folder

Demo-boot-withoutsecurity

5. Click on Finish

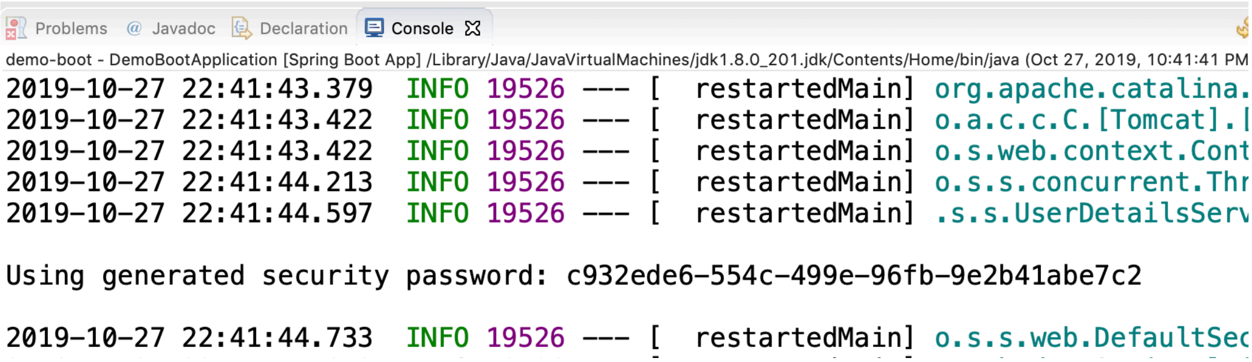
6. Add **spring-boot-starter-security** dependency

pom.xml

```
<dependency>
               <groupId>org.springframework.boot</groupId>
               <artifactId>spring-boot-starter-security</
artifactId>
          </dependency>
```

7. Run the Spring Boot App and observe in the console that it generates default password

Note: The Automatic restart of Spring Boot may not refresh password generation. So, restart the Spring Boot App.



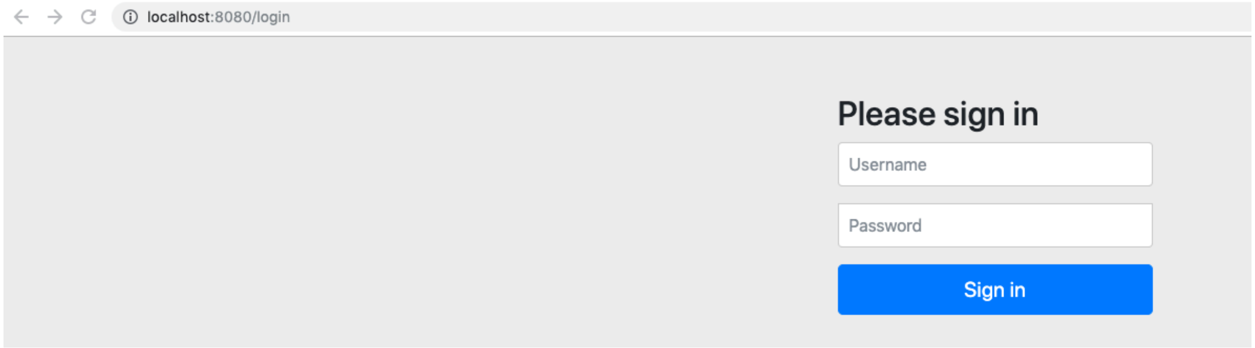
```
demo-boot - DemoBootApplication [Spring Boot App] /Library/Java/JavaVirtualMachines/jdk1.8.0_201.jdk/Contents/Home/bin/java (Oct 27, 2019, 10:41:41 PM
2019-10-27 22:41:43.379  INFO 19526 --- [  restartedMain] org.apache.catalina.
2019-10-27 22:41:43.422  INFO 19526 --- [  restartedMain] o.a.c.c.C.[Tomcat].[
2019-10-27 22:41:43.422  INFO 19526 --- [  restartedMain] o.s.web.context.Cont
2019-10-27 22:41:44.213  INFO 19526 --- [  restartedMain] o.s.s.concurrent.Thr
2019-10-27 22:41:44.597  INFO 19526 --- [  restartedMain] .s.s.UserDetailsServ

Using generated security password: c932ede6-554c-499e-96fb-9e2b41abe7c2

2019-10-27 22:41:44.733  INFO 19526 --- [  restartedMain] o.s.s.web.DefaultSec
```

Here, the Username is **"user"** by default

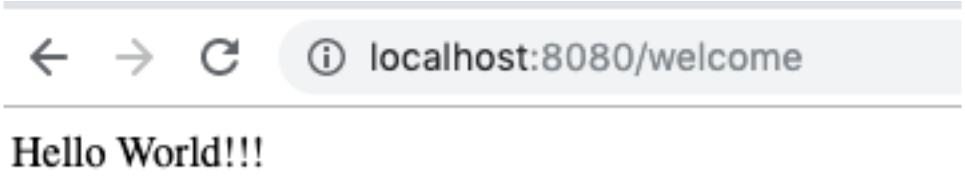
Spring boot generates password for the default user **"user"**

8. Visit ***http://localhost:8080/welcome***

9. The Spring Boot Security login page is displayed



10. Enter “user” and the given password code from the console



Hello World!!!

11. You can specify your own username and password in ***application.properties*** file

So, enter the following properties in:

application.properties

```
spring.security.user.name=john
spring.security.user.password=smith
```

12. RUN the App again

Note: The Automatic restart of Spring Boot may not refresh password generation. So, restart the Spring Boot App.

13. Clear the Cookie in Chrome by following steps mentioned in earlier Use case.

14. You should be able to login using your given credentials.

john/smith

15. You can now configure an AuthenticationManager and have the credentials in-memory.

```
package com.example.demo.config;

import org.springframework.context.annotation.Bean;
import
org.springframework.security.config.annotation.authentication.builders.Authe
nticationManagerBuilder;
import
org.springframework.security.config.annotation.web.configuration.EnableWebSe
curity;
import
org.springframework.security.config.annotation.web.configuration.WebSecurity
ConfigurerAdapter;
import org.springframework.security.crypto.password.~~NoOpPasswordEncoder~~;
import org.springframework.security.crypto.password.PasswordEncoder;

@EnableWebSecurity
public class SecurityConfiguration extends WebSecurityConfigurerAdapter {

	@Override
	protected void configure(AuthenticationManagerBuilder
authenticationManagerBuilder) throws Exception {
		authenticationManagerBuilder.inMemoryAuthentication()
			.withUser("admin")
			.password("admin@123")
			.roles("USER");

	}

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

}
```

Spring Boot Security bypasses default password and also the application.properties credentials once AuthenticationManager is configured.

16. RUN the APP

17. Must be able to login with in-memory credentials