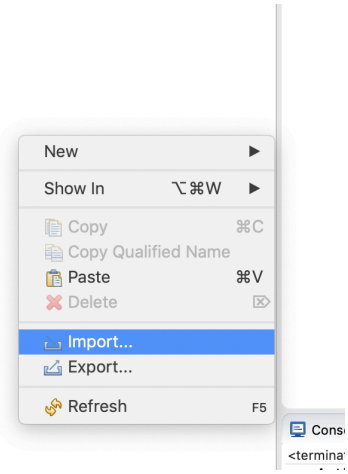
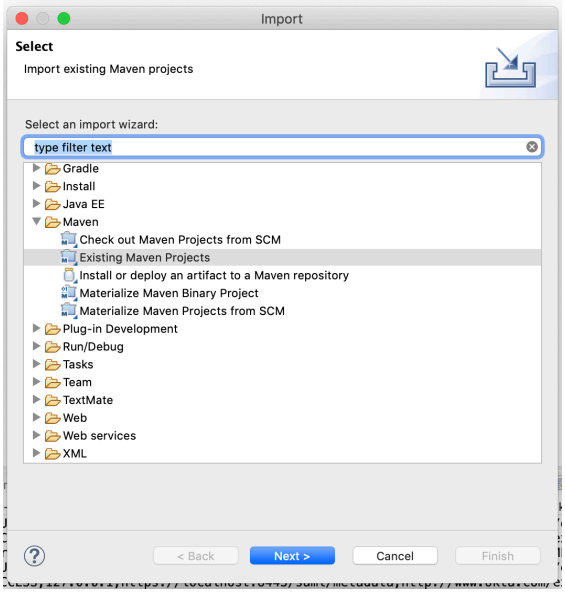
# *Create a Spring Boot Application to authenticate users with SAML 2.0 using OKTA Application for SSO*

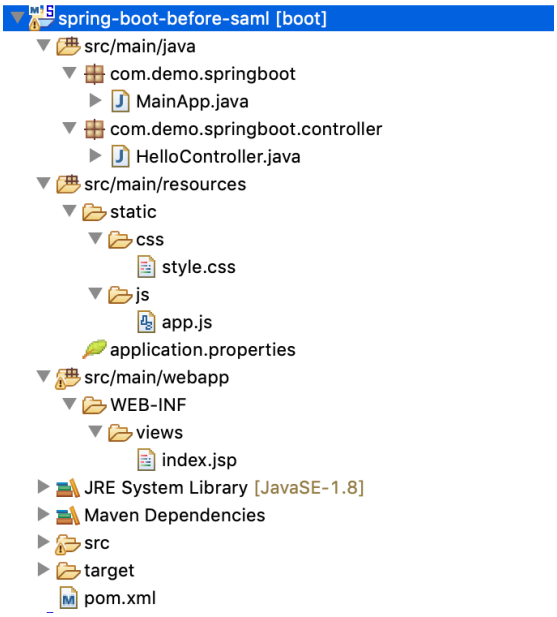
1. Import an existing sample Spring Boot Application **spring-boot-before-saml as a Maven Project**



2. Select **spring-boot-before-saml** folder and Finish

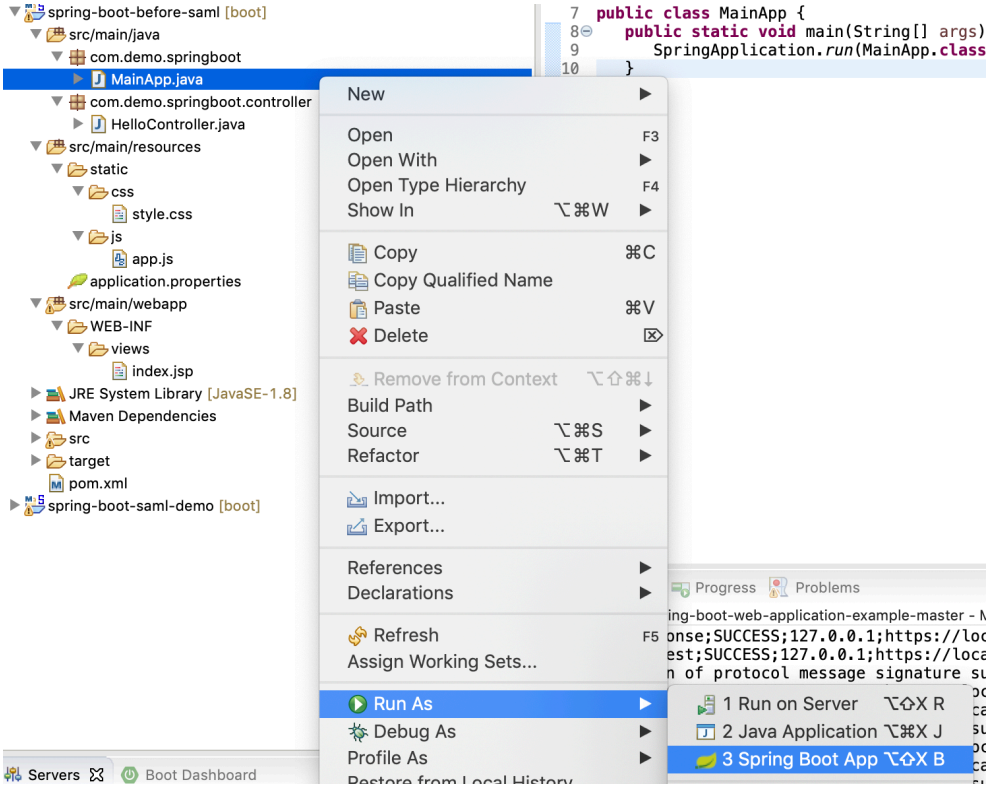
3. You will see the following project structure



4. Observe that there are some properties in **src/main/resources/application.properties**

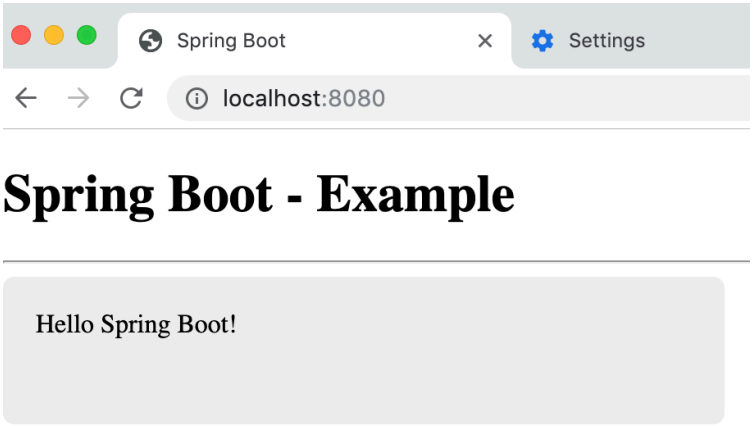
*spring.mvc.view.prefix = /WEB-INF/views/*
*spring.mvc.view.suffix = .jsp*
*spring.mvc.static-path-pattern=/resources/**

5. RUN the APP as is to see if it is working



6. OUTPUT

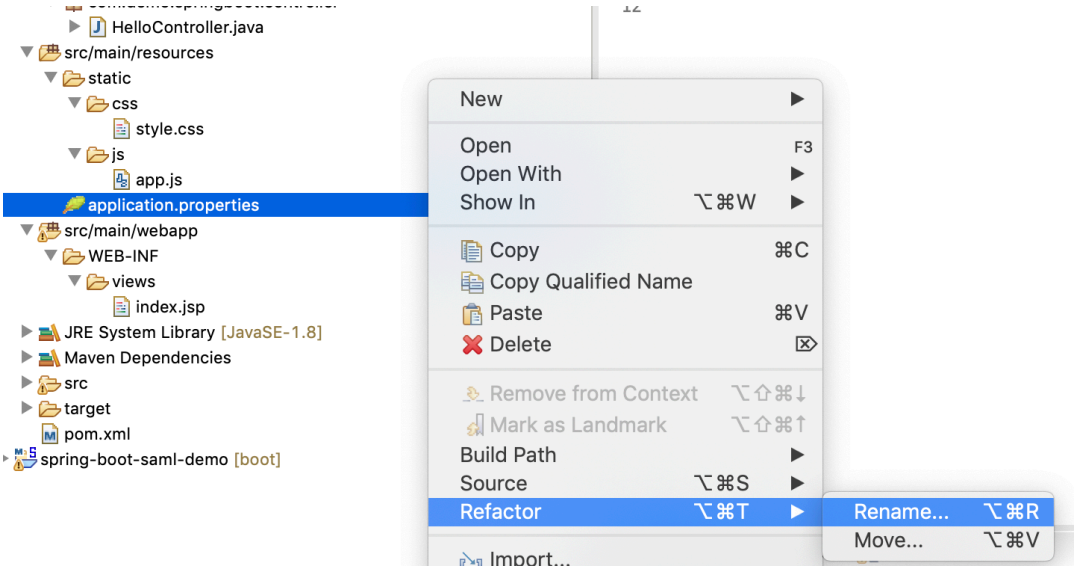
Visit ***http://localhost:8080/***

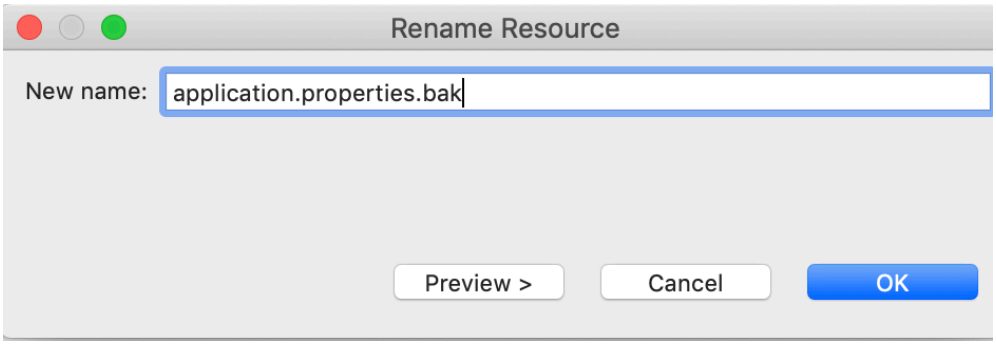



## *Replacing application.properties with application.yml*

7. **STOP** the running Spring Boot application.

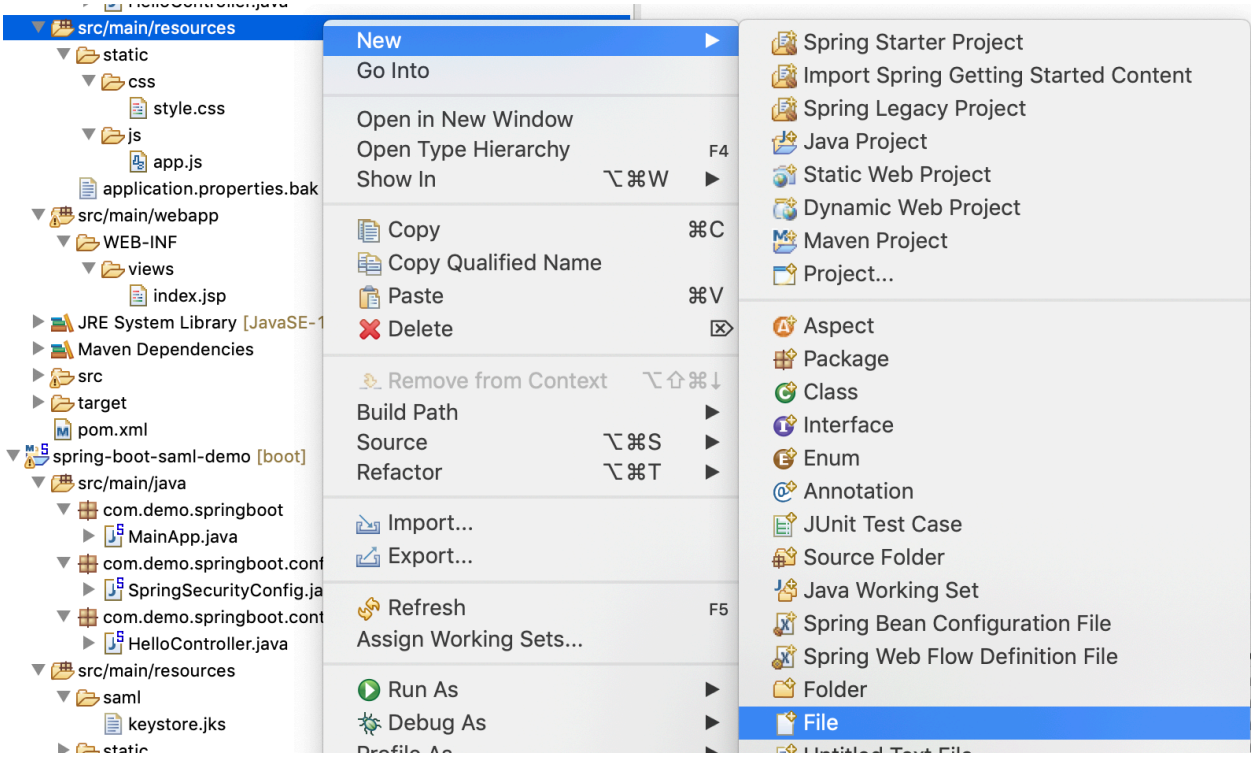
8. Rename application.properties to application.properties.bak



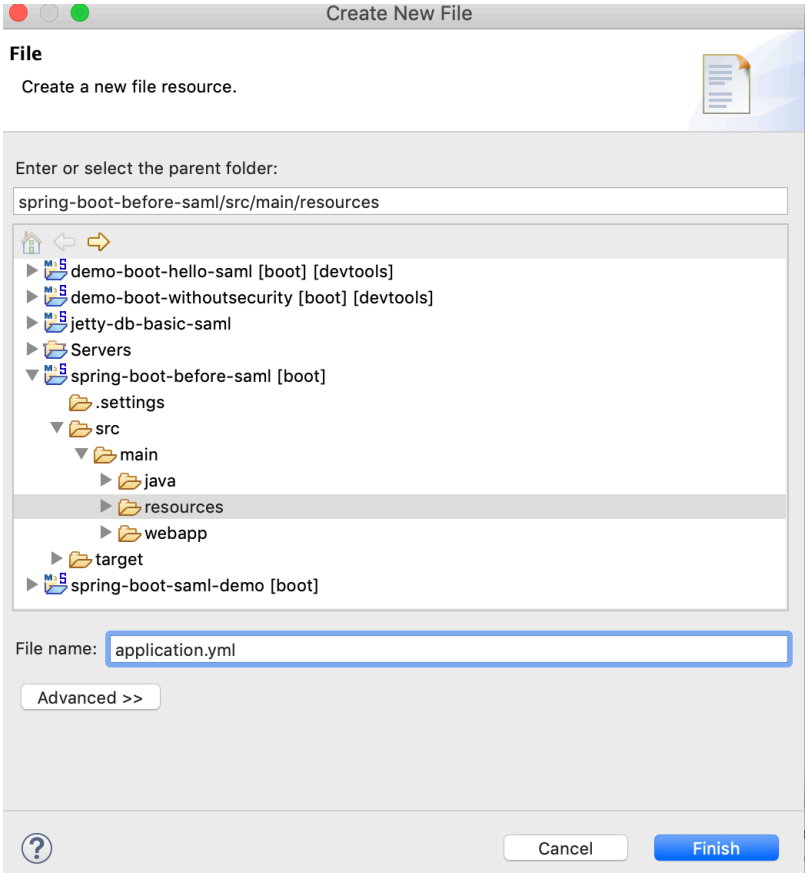



9. Click on OK

10. Create a YAML file



11. Enter the file name as **application.yml**

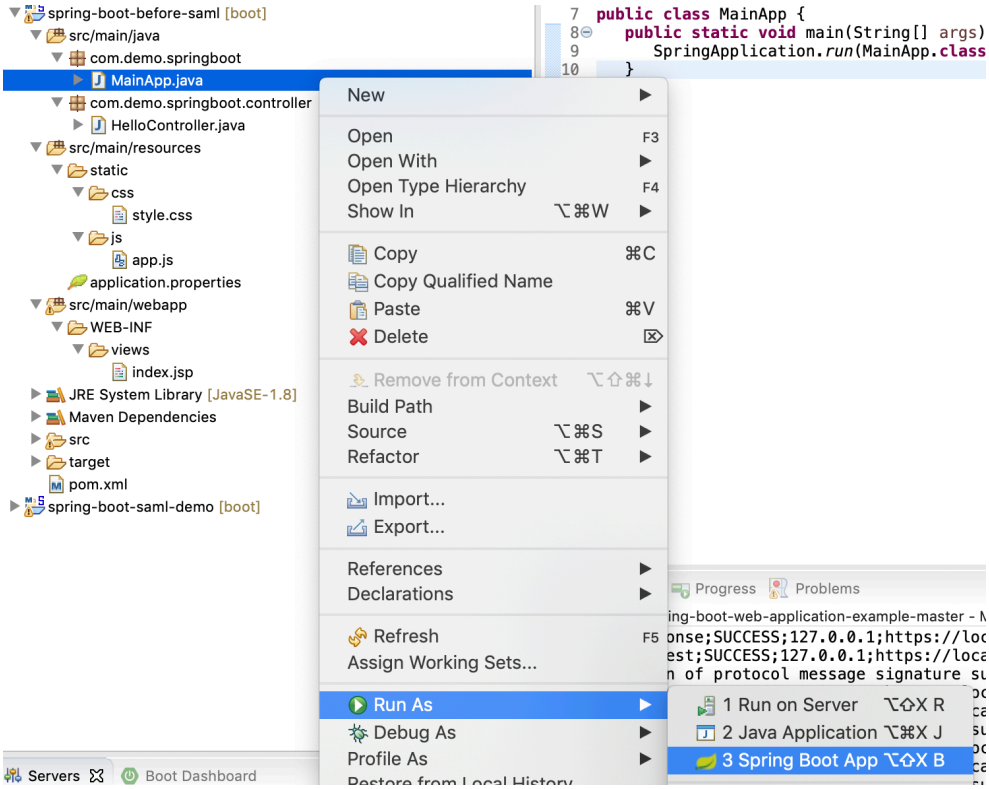



12. Enter the following properties in the **application.yml** file

```
spring:
  mvc:
    view:
      prefix: /WEB-INF/views/
      suffix: .jsp
    static-path-pattern: /resources/**
```

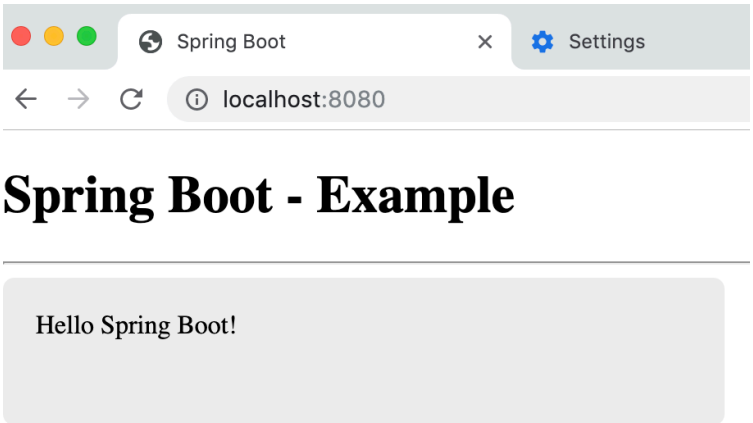
***NOTE: Be very very careful about the indentation and blank spaces in any .yml file***

13. RUN the APP to ensure it works as earlier




14. OUTPUT

Visit ***http://localhost:8080/***



# *Adding Spring Security with application.yml*

15. Lets add **spring-boot-starter-security** in

**pom.xml**

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

16. RUN the APP again to check Spring Boot default security

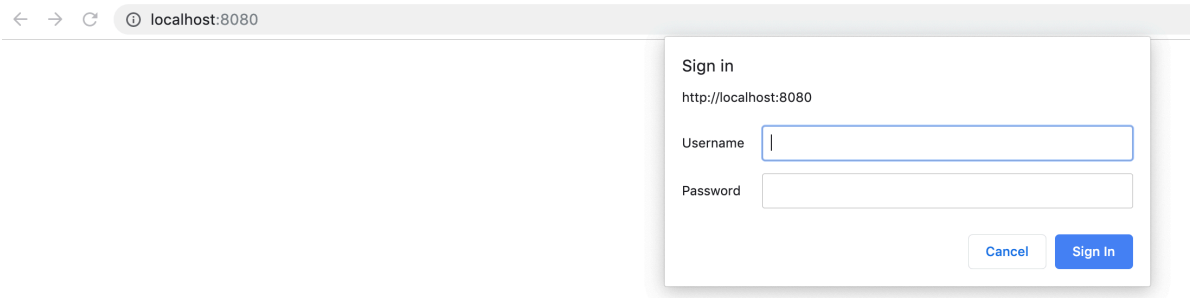
Right Click -> MainApp.java
     -> Run As
          -> Spring Boot App

17. OUTPUT

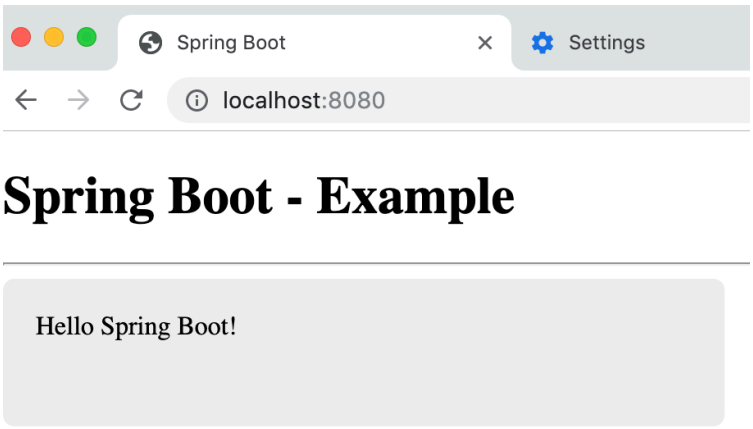
Observe the console for Password


18. OUTPUT

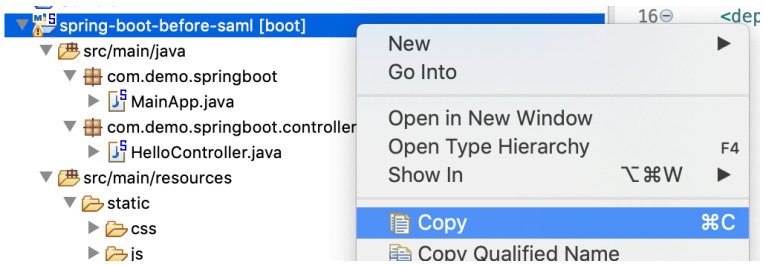
Visit ***http://localhost:8080/***




user / f426e3d7-8ff9-4cf5-8c05-eeb3aaf60dc8




## 19. Copy this project and create a copy called **spring-boot-saml-demo**



# *Integrating SAML for SSO Authentication with OKTA*

## Close all OPEN Files

20. Collapse the earlier project **spring-boot-before-saml**

## Start Making changes to **spring-boot-saml-demo**

21. Add SAML dependencies in **pom.xml**

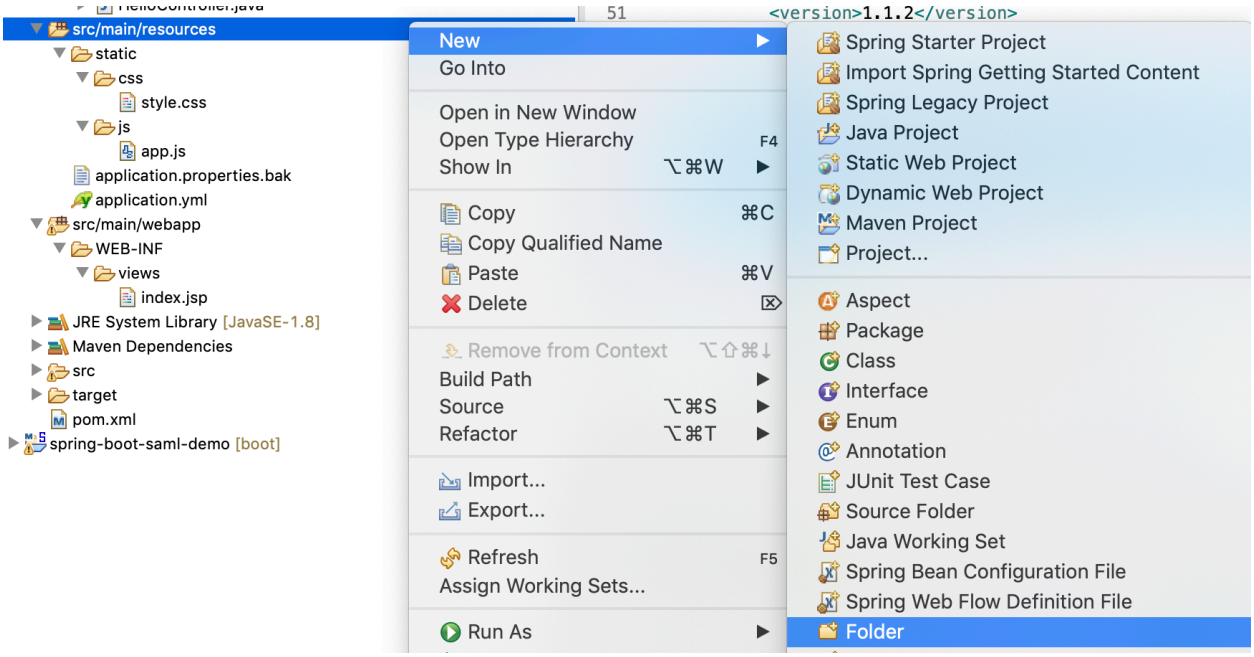
```
<!-- SAML2 DEPENDENCIES -->
<dependency>
	<groupId>org.springframework.security.extensions</groupId>
	<artifactId>spring-security-saml2-core</artifactId>
	<version>1.0.3.RELEASE</version>
</dependency>
<dependency>
	<groupId>org.springframework.security.extensions</groupId>
	<artifactId>spring-security-saml-dsl-core</artifactId>
	<version>1.0.5.RELEASE</version>
</dependency>
```

22. Also need other dependencies as shown below:

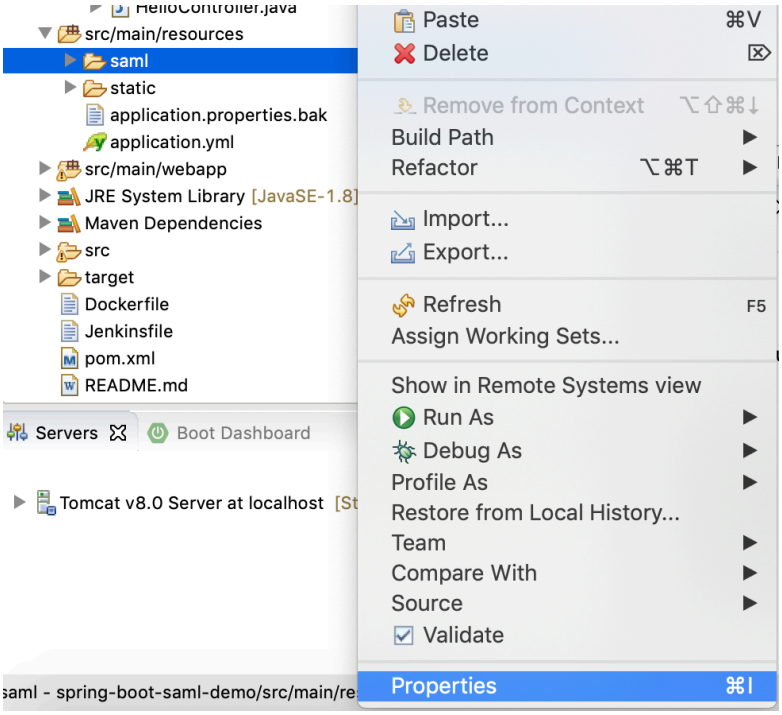
```
<!-- Servlet and JSP related DEPENDENCIES -->
		<dependency>
			<groupId>javax.servlet</groupId>
			<artifactId>javax.servlet-api</artifactId>
		</dependency>
		<dependency>
			<groupId>javax.servlet.jsp</groupId>
			<artifactId>javax.servlet.jsp-api</artifactId>
			<version>2.3.1</version>
		</dependency>
```

23. Navigate to your project (in the `src/main/resources` folder).

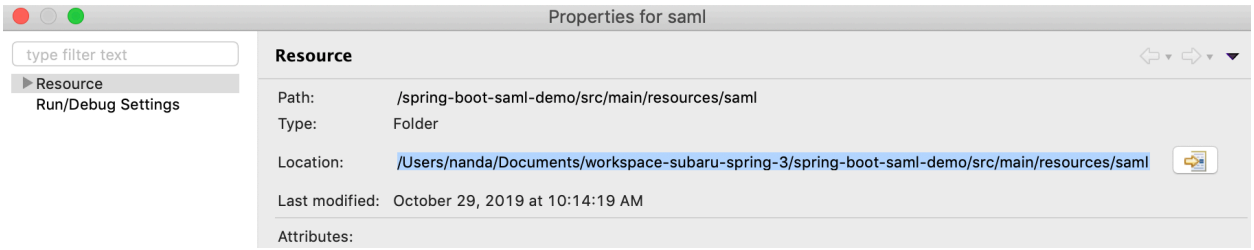
Create a folder named `saml`



24. Open the Command Prompt in `saml` location.




Copy the Resource path:



25. Open CMD

**# cd <path-of-saml-folder>**

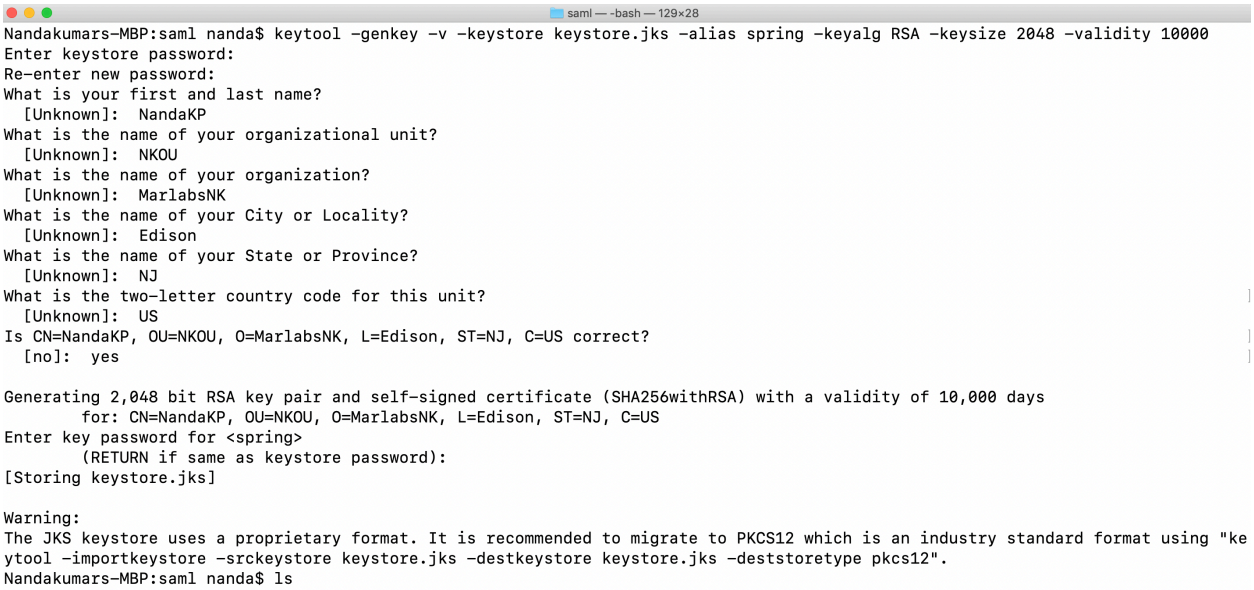
Example, in my project, I will be here in CMD

/Users/nanda/Documents/workspace-subaru-spring-3/spring-boot-saml-demo/src/main/resources/saml

26. Execute the following command:

# keytool -genkey -v -keystore keystore.jks -alias spring -keyalg RSA -keysize 2048 -validity 10000

When prompted, give the required details and create the `keystore.jks` file within the `src/main/resources/saml` folder

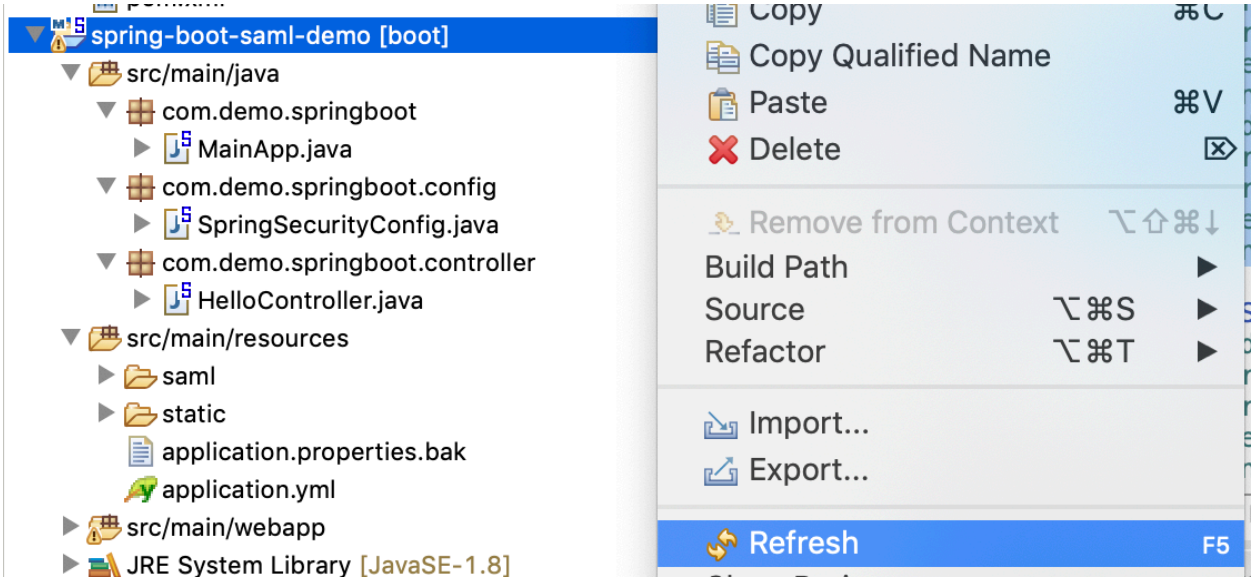


27. Password must be **testsaml**

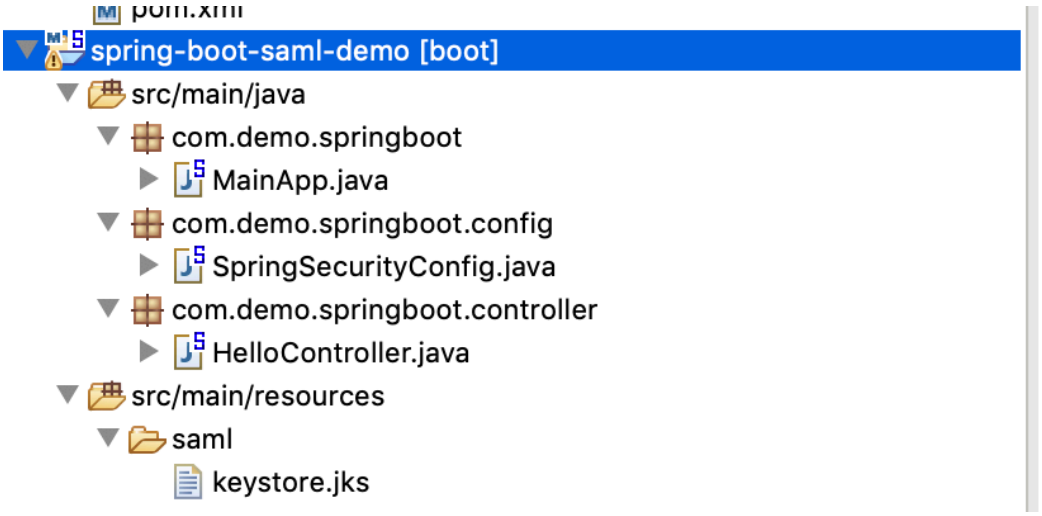
28. Check if the **keystore.jks** is created in the **saml** folder

```
# ls
```

**29. REFRESH your project in Eclipse**



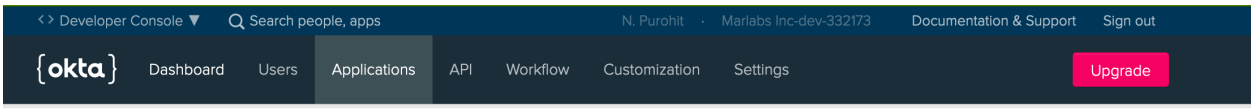

**30. You must see keystore.jks under saml folder**



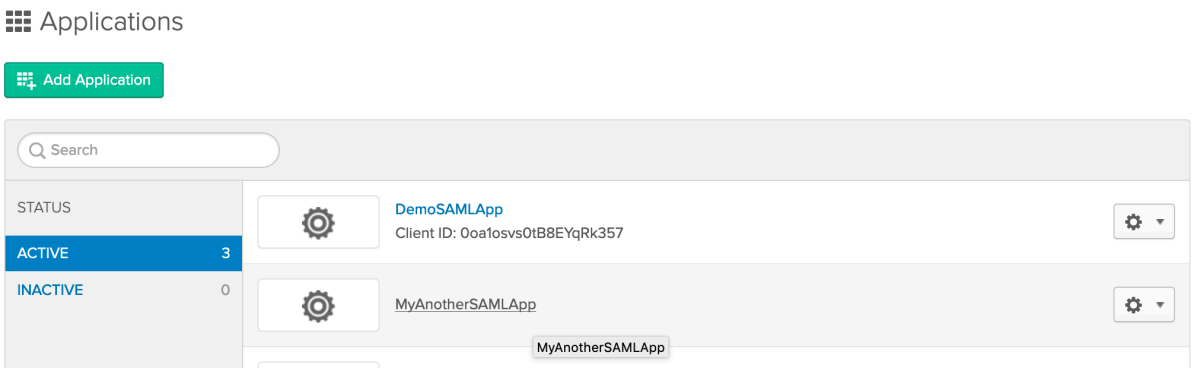
31. Create a **SpringSecurityConfig** class in **com.demo.springboot.config** package

```
package com.demo.springboot.config;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Configuration;
import
org.springframework.security.config.annotation.method.configuration.EnableGl
obalMethodSecurity;
import
org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.web.configuration.EnableWebSe
curity;
import
org.springframework.security.config.annotation.web.configuration.WebSecurity
ConfigurerAdapter;

import static
org.springframework.security.extensions.saml2.config.SAMLConfigurer.saml;

@EnableWebSecurity
@Configuration
@EnableGlobalMethodSecurity(securedEnabled = true)
public class SpringSecurityConfig extends WebSecurityConfigurerAdapter {

    //@Value("${security.saml2.metadata-url}")
      @Value("https://dev-332173.okta.com/app/exk1oua3jharTMEfy357/sso/saml/
metadata")
    String metadataUrl;

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http
                .authorizeRequests()
                .antMatchers("/saml/**").permitAll()
                .anyRequest().authenticated()
                .and()
                .apply(saml())
                .serviceProvider()
                .keyStore()
                .storeFilePath("saml/keystore.jks")
                .password("testsaml")
                .keyname("spring")
                .keyPassword("testsaml")
                .and()
                .protocol("https")
                .hostname("localhost:8443")
                .basePath("/")
                .and()
                .identityProvider()
                .metadataFilePath(metadataUrl)
                .and();
    }

}
```

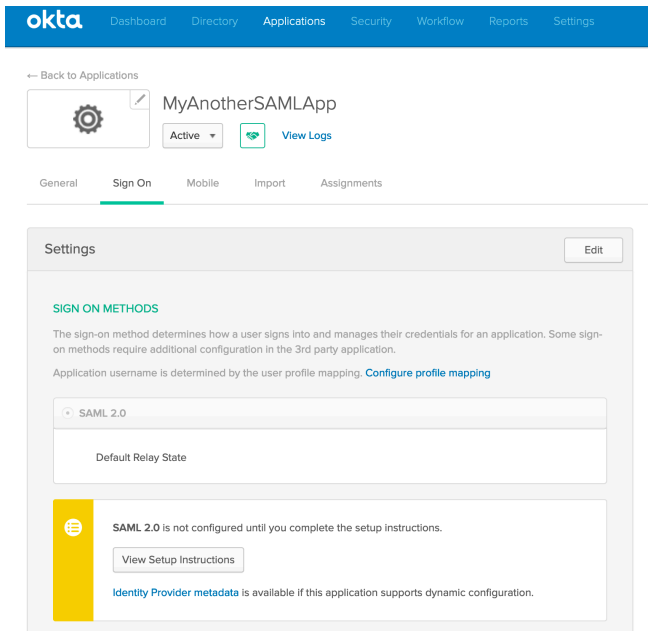
31.1 You must specify the METADATA path of your SAML Assigned Application from OKTA for the metadataUrl variable as shown below:




31.2 Click on your Application Name

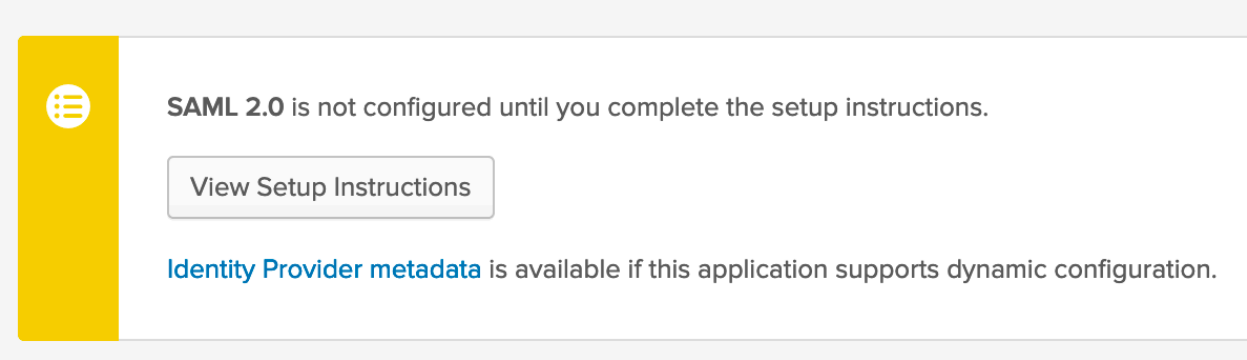



31.3 Go to **Sign On** Tab as shown here



31.4 In the Settings section, You will see a link called **Identity Provider Metadata**

Right click on this link -> Copy Link Address / location




31.5 Paste this Metadata URL in your Config file for @Value of metadataUrl variable as shown below:

**SpringSecurityConfig.java**

```
@EnableGlobalMethodSecurity(securedEnabled = true)
public class SpringSecurityConfig extends WebSecurityConfigurerAdapter {

    //@Value("${security.saml2.metadata-url}")
    @Value("https://dev-332173.okta.com/app/exk1oua3jharTMEfy357/sso/saml/metadata")
    String metadataUrl;
```

32. Add **HTTPS & Keystore attributes** in **application.yml**

Add on top of existing **spring** attribute

```
server:
  port: 8443
  ssl:
    enabled: true
    key-alias: spring
    key-store: src/main/resources/saml/keystore.jks
    key-store-password: testsaml
  tomcat:
    additional-tld-skip-patterns: '*.jar'

security:
  metadata-url: https://dev-332173.okta.com/app/exk1oua3jharTMEfy357/sso/
saml/metadata
```

32.1 Paste your metadata url in .yml file too.

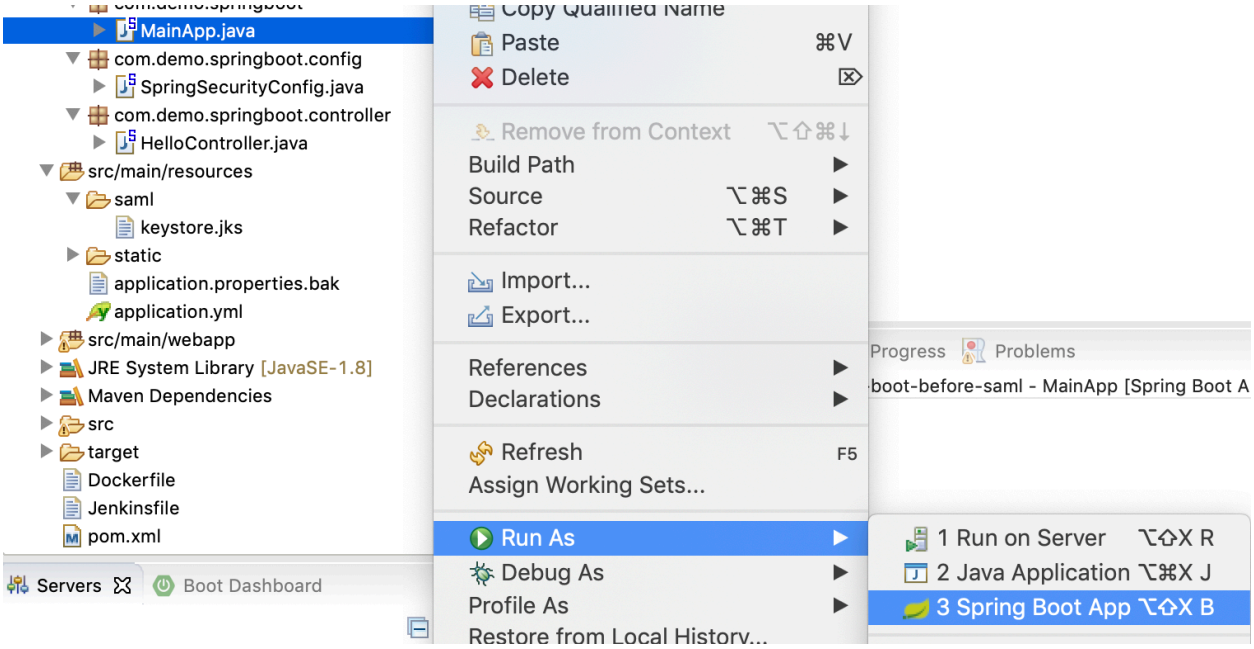
**application.yml**

```
10
11 security:
12   metadata-url: https://dev-332173.okta.com/app/exk1oua3jharTMEfy357/sso/saml/metadata
13
```

MainApp.java | application.yml | spring-boot-bef | spring-boot-sam | SpringSecurityC

```yaml
server:
  port: 8443
  ssl:
    enabled: true
    key-alias: spring
    key-store: src/main/resources/saml/keystore.jks
    key-store-password: testsaml
  tomcat:
    additional-tld-skip-patterns: '*.jar'

security:
  metadata-url: https://dev-332173.okta.com/app/exk1oua3jharTMEfy357/sso/saml/metadata

spring:
  mvc:
    view:
      prefix: /WEB-INF/views/
      suffix: .jsp
    static-path-pattern: /resources/**
```

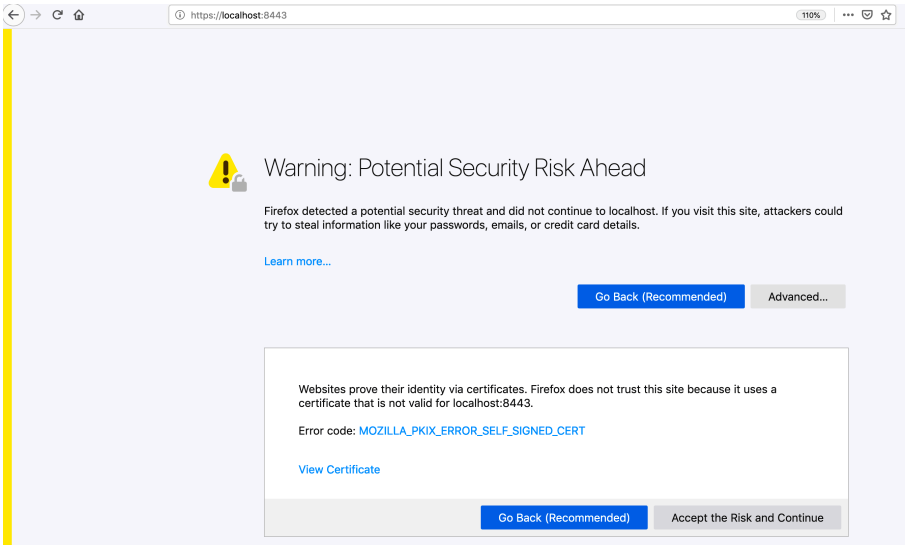
## 33. RUN the APP



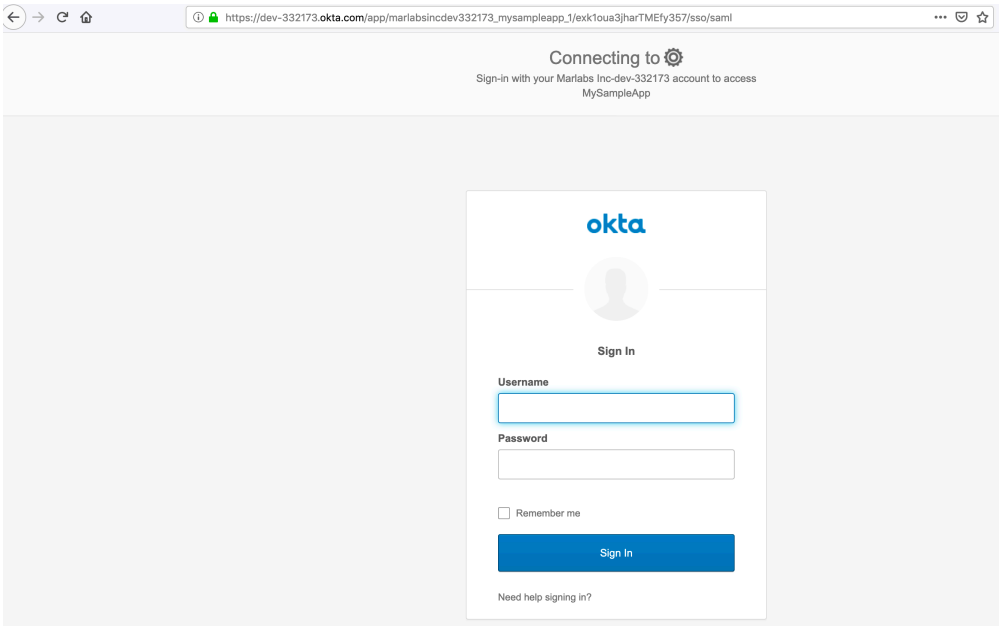

## 34. OUTPUT

Visit https://localhost:8443

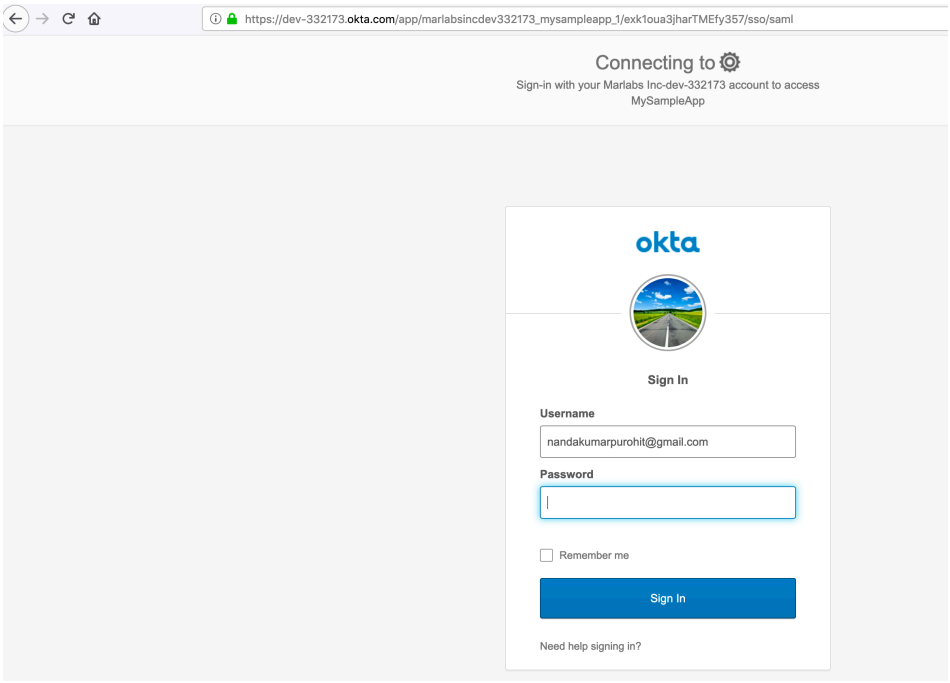
Click on **Accept the Risk & Continue**




35. It should redirect to your okra login page

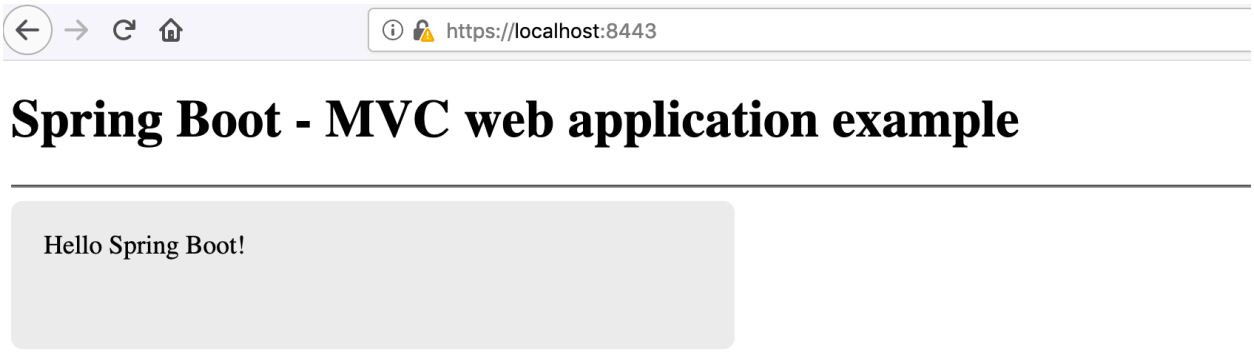



36. When you enter your email id, it shows your profile image



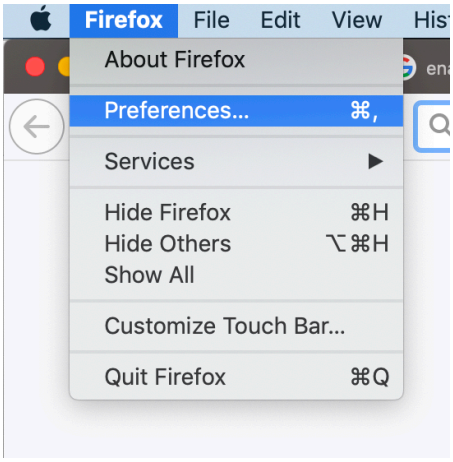

37. Enter the password as well.

38. It must authenticate and redirect back to the requested page as shown:

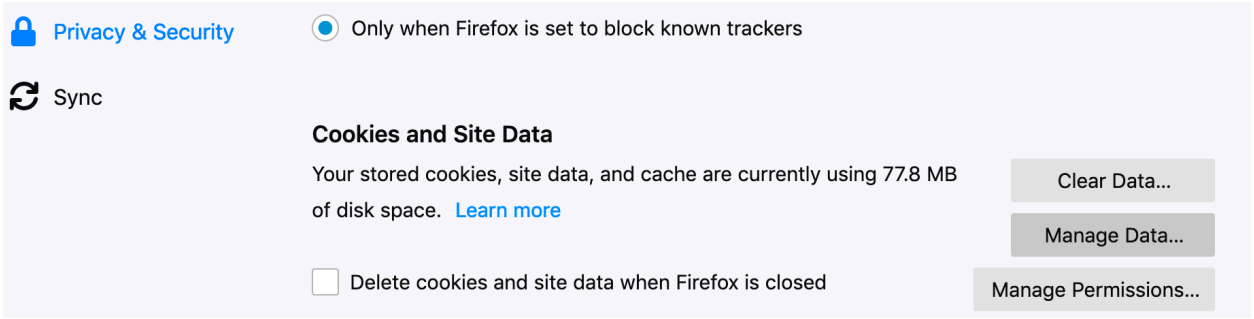


## *If you want to re-login, then do the following*

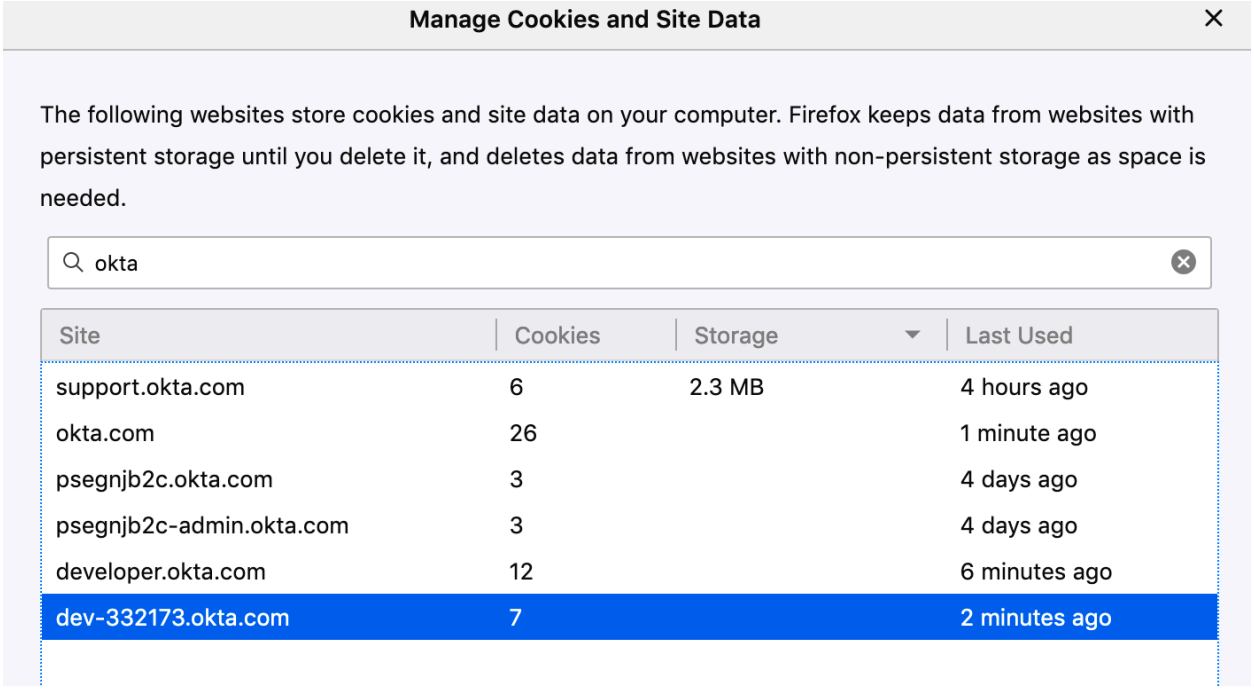
39. Clear the browser cache for both localhost & okra site



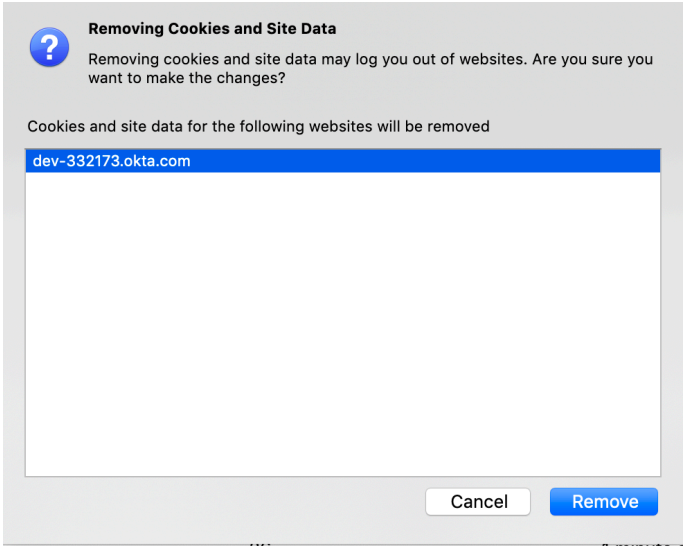

40.



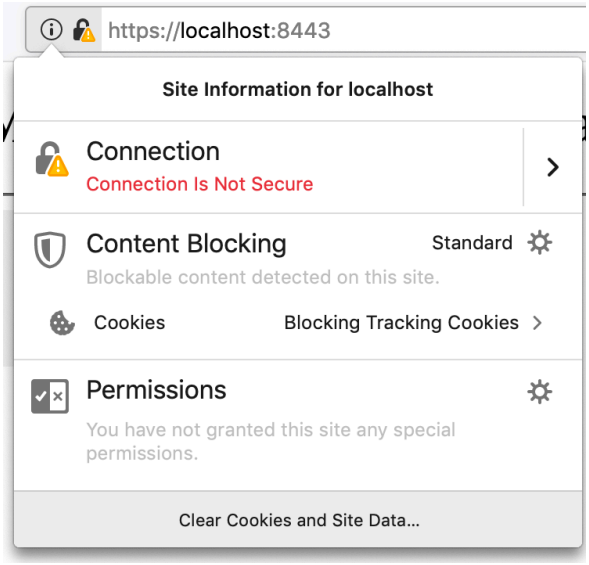
41.


Manage Cookies and Site Data
The following websites store cookies and site data on your computer. Firefox keeps data from websites with persistent storage until you delete it, and deletes data from websites with non-persistent storage as space is needed.
okta
Site
Cookies
Storage
Last Used
support.okta.com
6
2.3 MB
4 hours ago
okta.com
26
1 minute ago
psegnjb2c.okta.com
3
4 days ago
psegnjb2c-admin.okta.com
3
4 days ago
developer.okta.com
12
6 minutes ago
dev-332173.okta.com
7
2 minutes ago

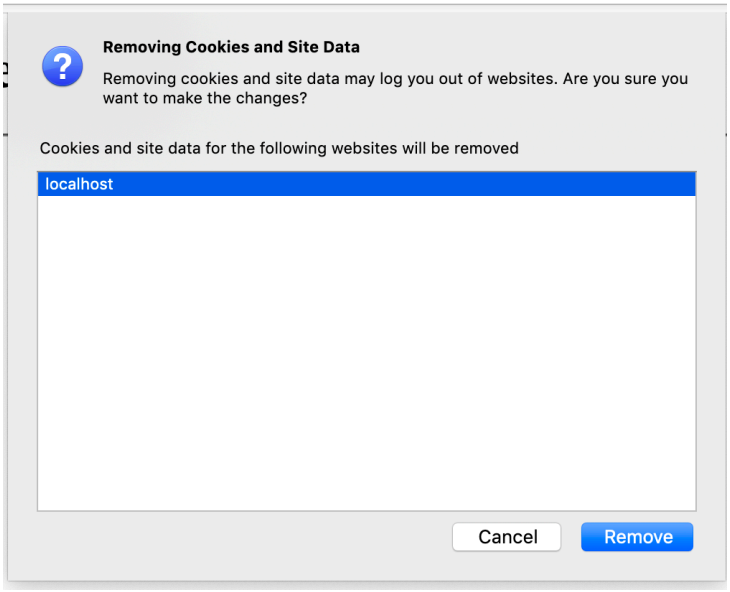
42.


Removing Cookies and Site Data
Removing cookies and site data may log you out of websites. Are you sure you want to make the changes?
Cookies and site data for the following websites will be removed
dev-332173.okta.com
Cancel
Remove

43.


https://localhost:8443
Site Information for localhost
Connection
Connection Is Not Secure
Content Blocking
Standard
Blockable content detected on this site.
Cookies
Blocking Tracking Cookies
Permissions
You have not granted this site any special permissions.
Clear Cookies and Site Data...

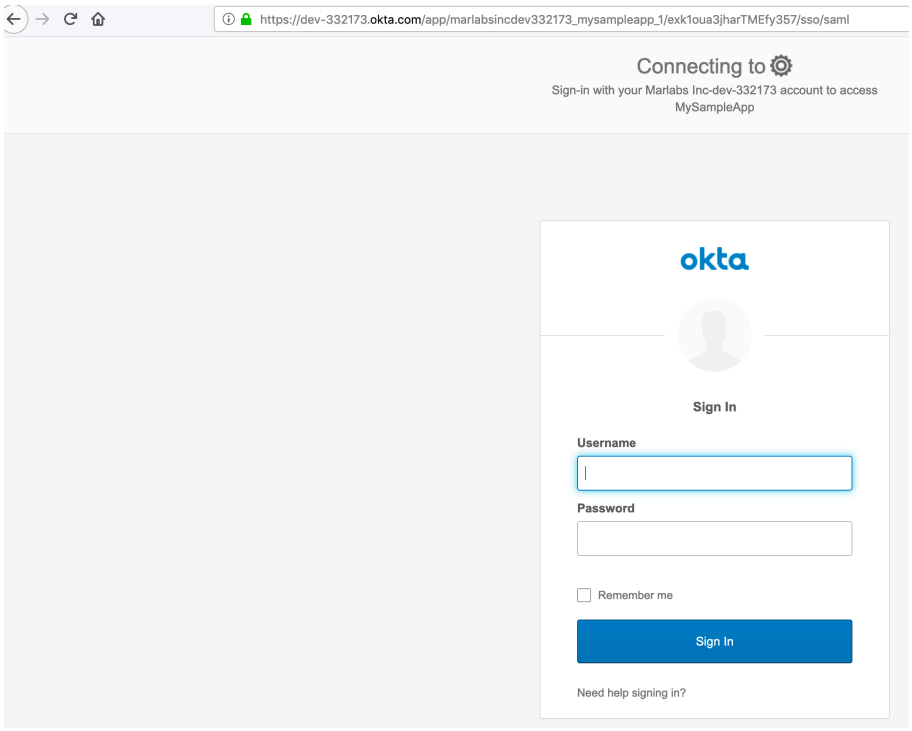
44.


Removing Cookies and Site Data
Removing cookies and site data may log you out of websites. Are you sure you want to make the changes?
Cookies and site data for the following websites will be removed
localhost
Cancel
Remove

45.


Config
enable
Setting
java.io
https://localhost:8443
Reload current page (⌘R)

46.


https://dev-332173.okta.com/app/marlabsincdev332173_mysampleapp_1/exk1oua3jharTMEfy357/sso/saml
Connecting to
Sign-in with your Marlabs Inc-dev-332173 account to access MySampleApp
okta
Sign In
Username
Password
Remember me
Sign In
Need help signing in?