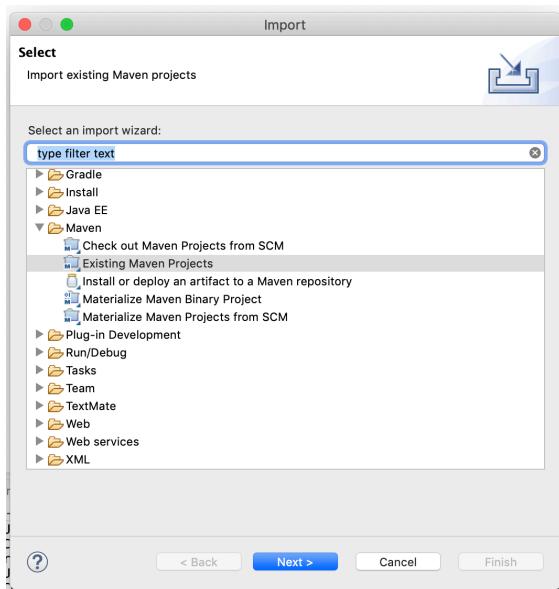
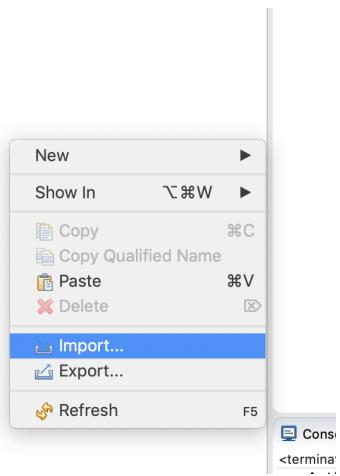


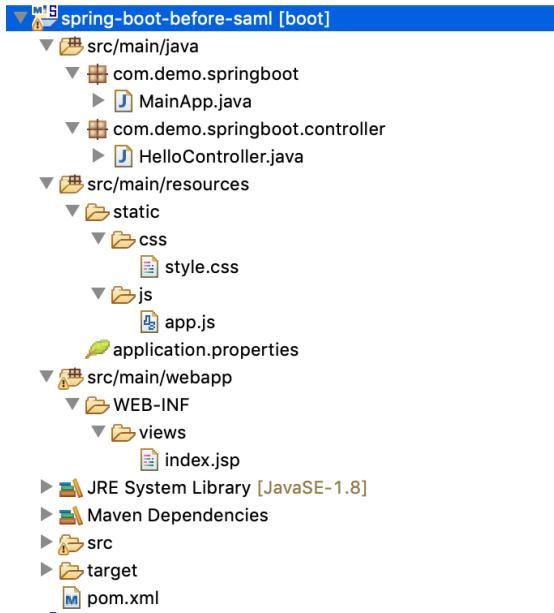
Create a Spring Boot Application to authenticate users with SAML 2.0 using OKTA Application for SSO

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

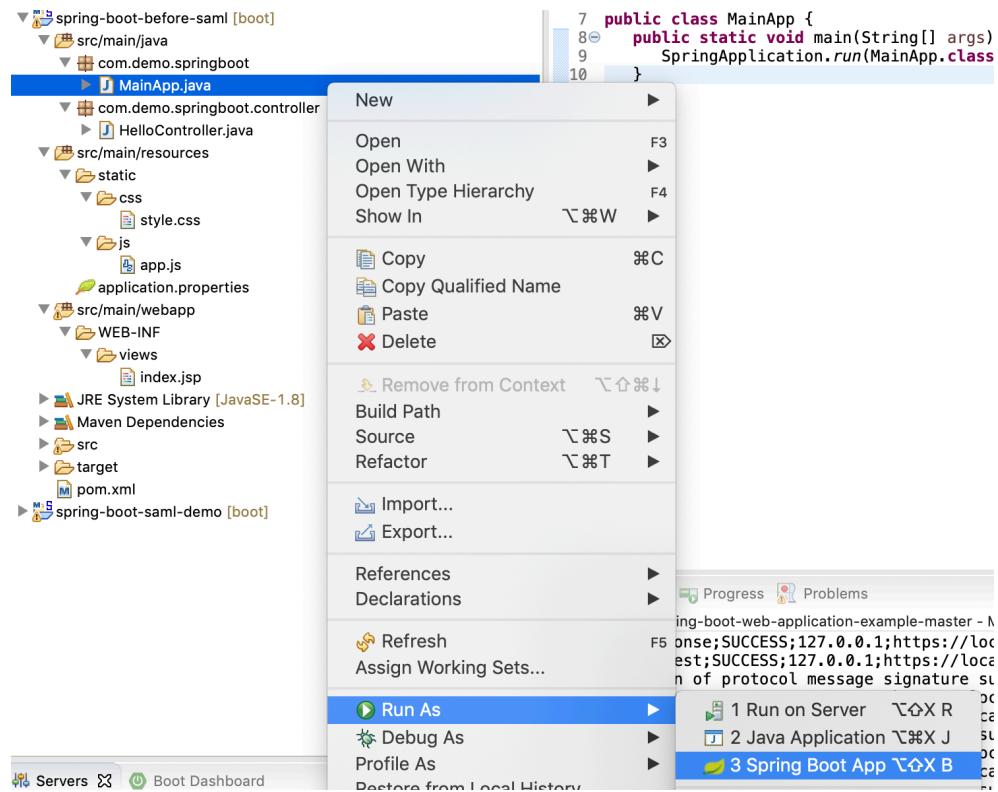
3. You will see the following project structure



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

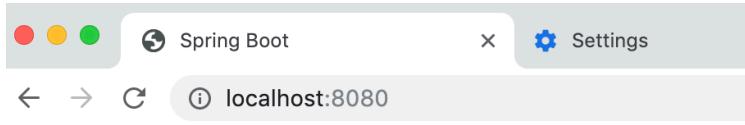
```
spring.mvc.view.prefix = /WEB-INF/views/  
spring.mvc.view.suffix = .jsp  
spring.mvc.static-path-pattern=/resources/**
```

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



6. OUTPUT

Visit <http://localhost:8080/>



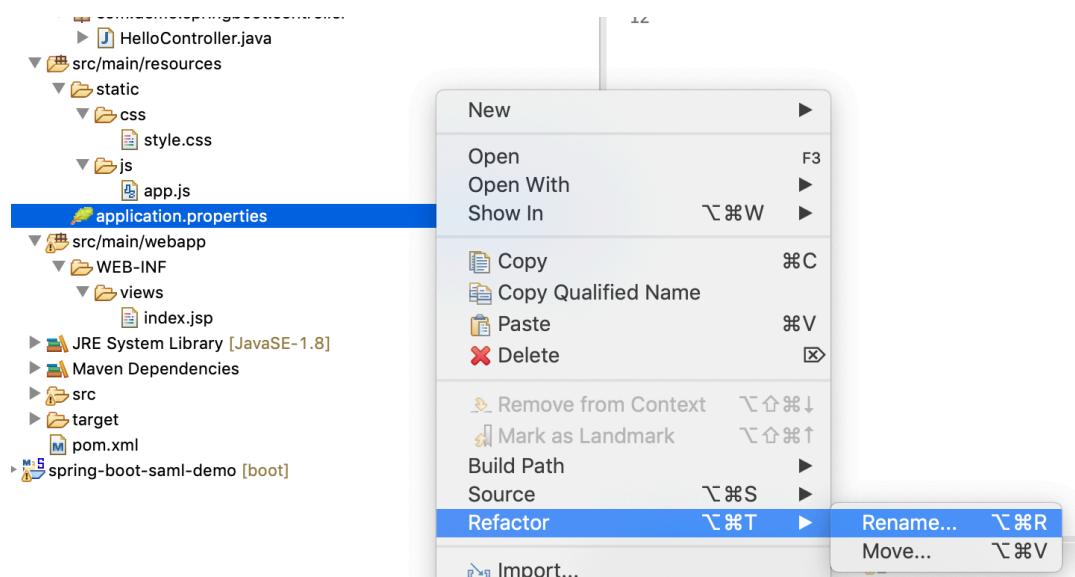
Spring Boot - Example

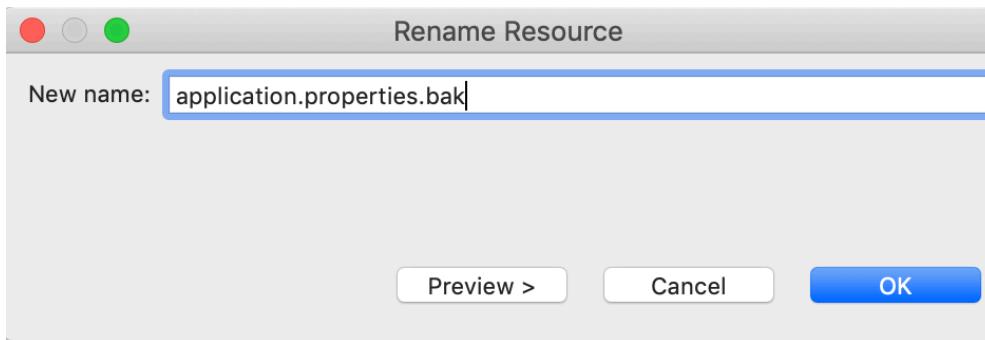
Hello Spring Boot!

Replacing application.properties with application.yml

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

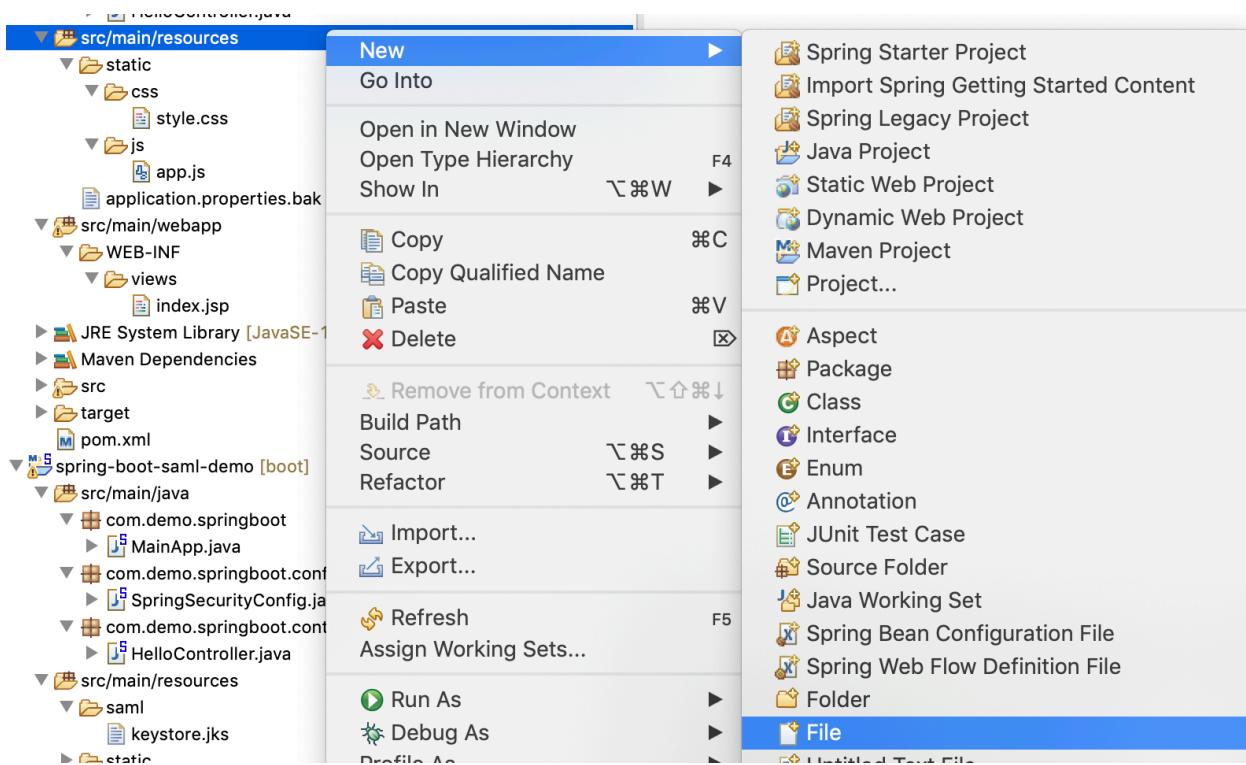
8. Rename application.properties to application.properties.bak



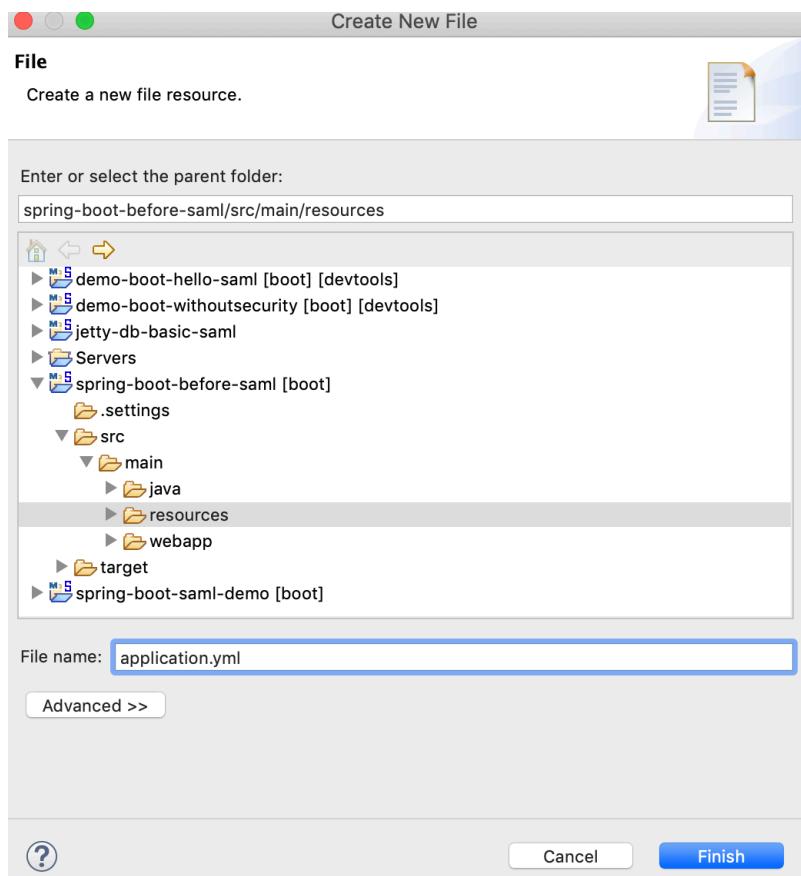


9. Click on OK

10. Create a YAML file



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

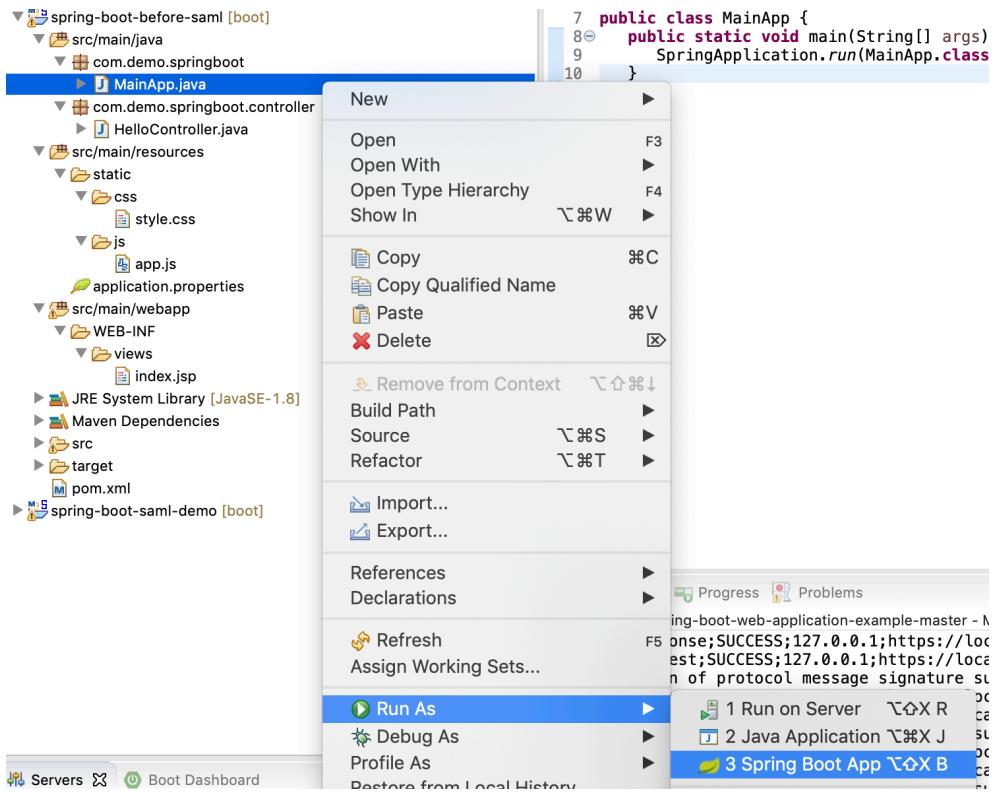


12. Enter the following properties in the **application.yml** file

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

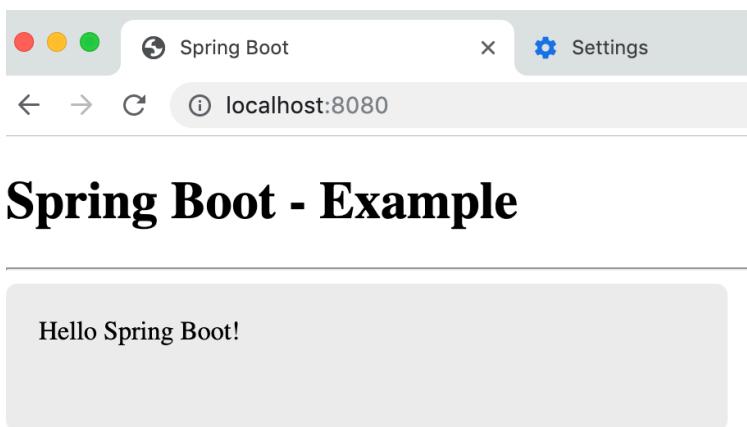
NOTE: Be very very careful about the indentation and blank spaces in any .yml file

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 Ap

17. OUTPUT

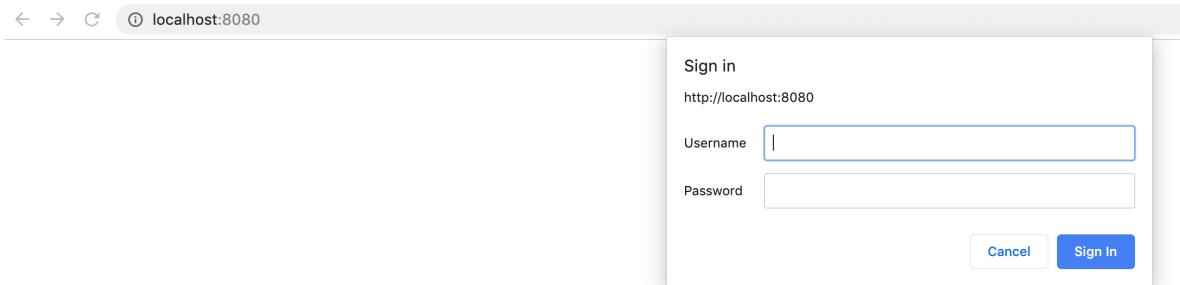
Observe the console for Password

```
Console Progress Problems
spring-boot-before-saml - Main App [Spring Boot App] /Library/Java/JavaVirtualMachines/jdk1.8.0_201.jdk/Contents/Home/bin/java (Oct 29, 2019, 1:04:48 PM)
2019-10-29 13:04:49.908 INFO 22229 --- [           main] s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @Controller
2019-10-29 13:04:49.936 INFO 22229 --- [           main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/]}" onto public
2019-10-29 13:04:49.937 INFO 22229 --- [           main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/hello], metl
2019-10-29 13:04:49.939 INFO 22229 --- [           main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error]} or
2019-10-29 13:04:49.939 INFO 22229 --- [           main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error], pro
2019-10-29 13:04:49.952 INFO 22229 --- [           main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/web
2019-10-29 13:04:49.952 INFO 22229 --- [           main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/res
2019-10-29 13:04:49.968 INFO 22229 --- [           main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**
2019-10-29 13:04:50.058 INFO 22229 --- [           main] b.a.AuthenticationManagerConfiguration : Using default security password: f426e3d7-8ff9-4cf5-8c05-eeb3aaf60dc8

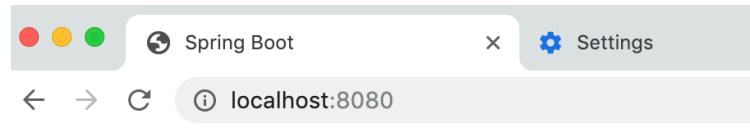
2019-10-29 13:04:50.085 INFO 22229 --- [           main] o.s.s.web.DefaultSecurityFilterChain : Creating filter chain
2019-10-29 13:04:50.125 INFO 22229 --- [           main] o.s.s.web.DefaultSecurityFilterChain : Creating filter chain
2019-10-29 13:04:50.187 INFO 22229 --- [           main] o.s.l.e.AnnotationMBeanExporter : Registering beans for
```

18. OUTPUT

Visit <http://localhost:8080/>



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

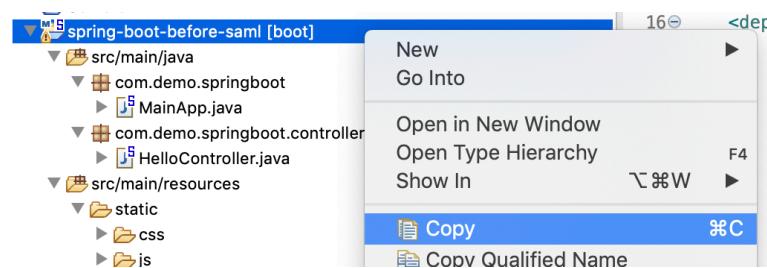


Spring Boot - Example

Hello Spring Boot!

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**

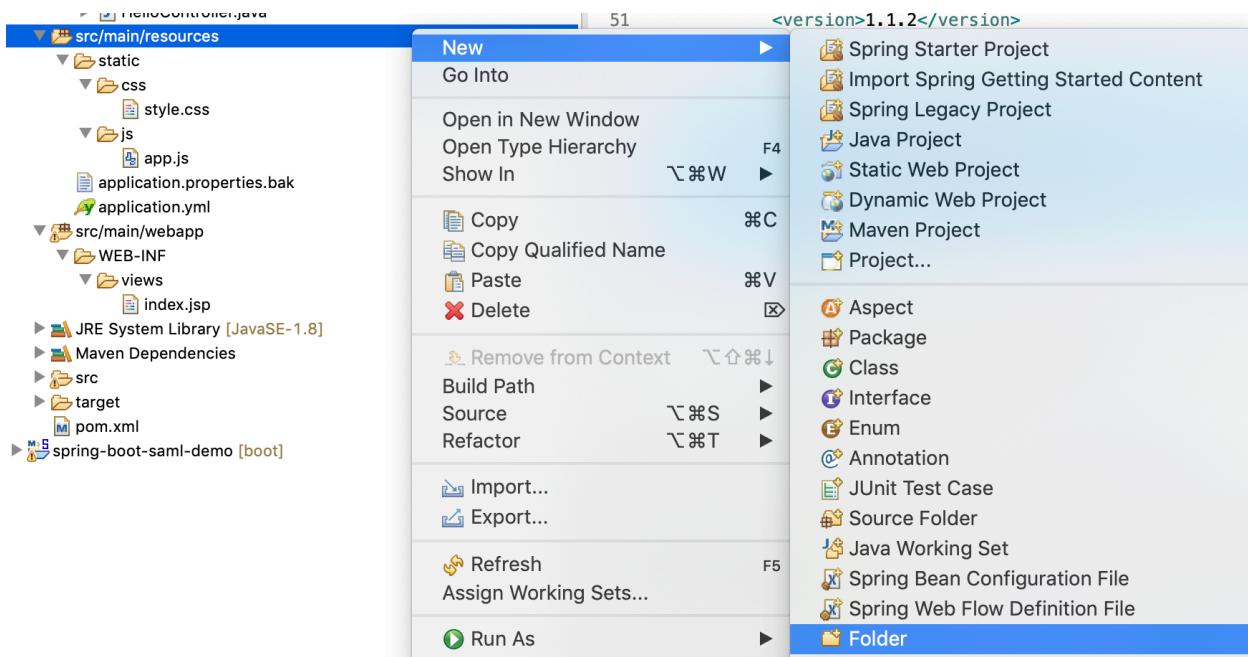
```
<!-- SAML2 DEPENDENCIES -->
<dependency>
    <groupId>org.springframework.security.extensions</groupId>
    <artifactId>spring-security-saml2-core</artifactId>
    <version>1.0.3.RELEASE</version>
</dependency>
<dependency>
    <groupId>org.springframework.security.extensions</groupId>
    <artifactId>spring-security-saml-dsl-core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

22. Also need other dependencies as shown below:

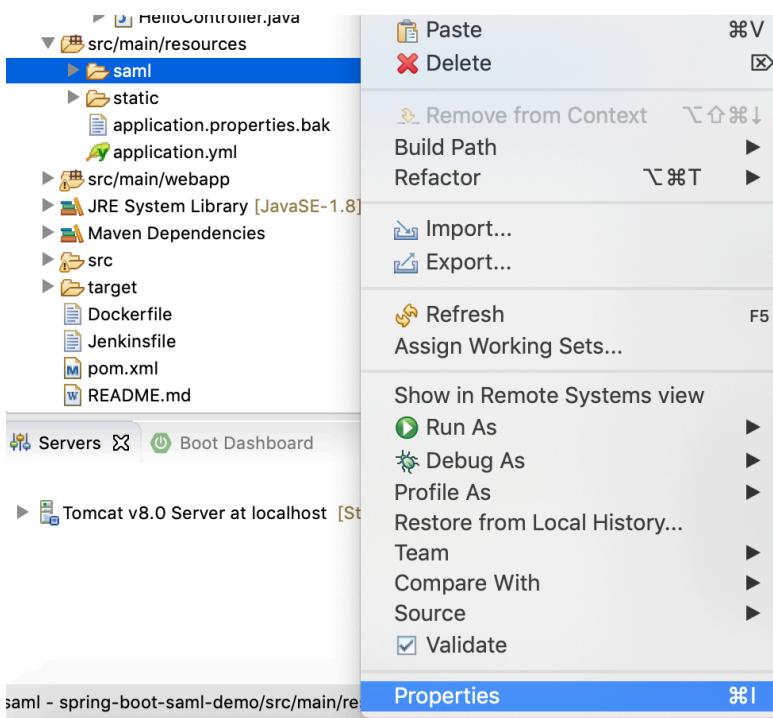
```
<!-- Servlet and JSP related DEPENDENCIES -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
</dependency>
<dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>javax.servlet.jsp-api</artifactId>
    <version>2.3.1</version>
</dependency>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

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

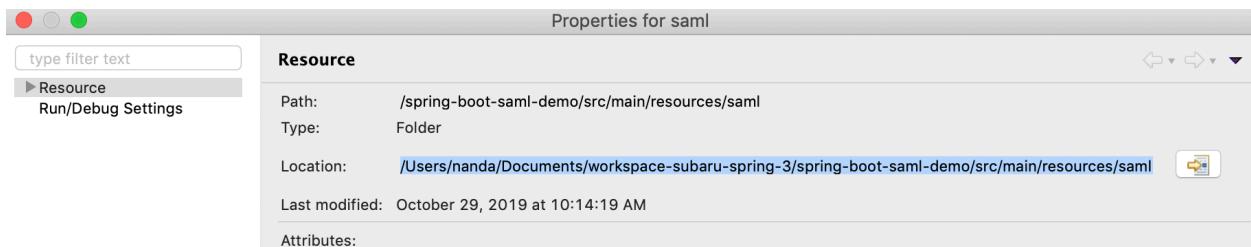
Create a folder named `saml`



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

```
Nandakumars-MBP:saml nanda$ keytool -genkey -v -keystore keystore.jks -alias spring -keyalg RSA -keysize 2048 -validity 10000
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: NandaKP
What is the name of your organizational unit?
[Unknown]: NKOU
What is the name of your organization?
[Unknown]: MarlabsNK
What is the name of your City or Locality?
[Unknown]: Edison
What is the name of your State or Province?
[Unknown]: NJ
What is the two-letter country code for this unit?
[Unknown]: US
Is CN=NandaKP, OU=NKOU, O=MarlabsNK, L=Edison, ST=NJ, C=US correct?
[no]: yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
for: CN=NandaKP, OU=NKOU, O=MarlabsNK, L=Edison, ST=NJ, C=US
Enter key password for <spring>
  (RETURN if same as keystore password):
[Storing keystore.jks]

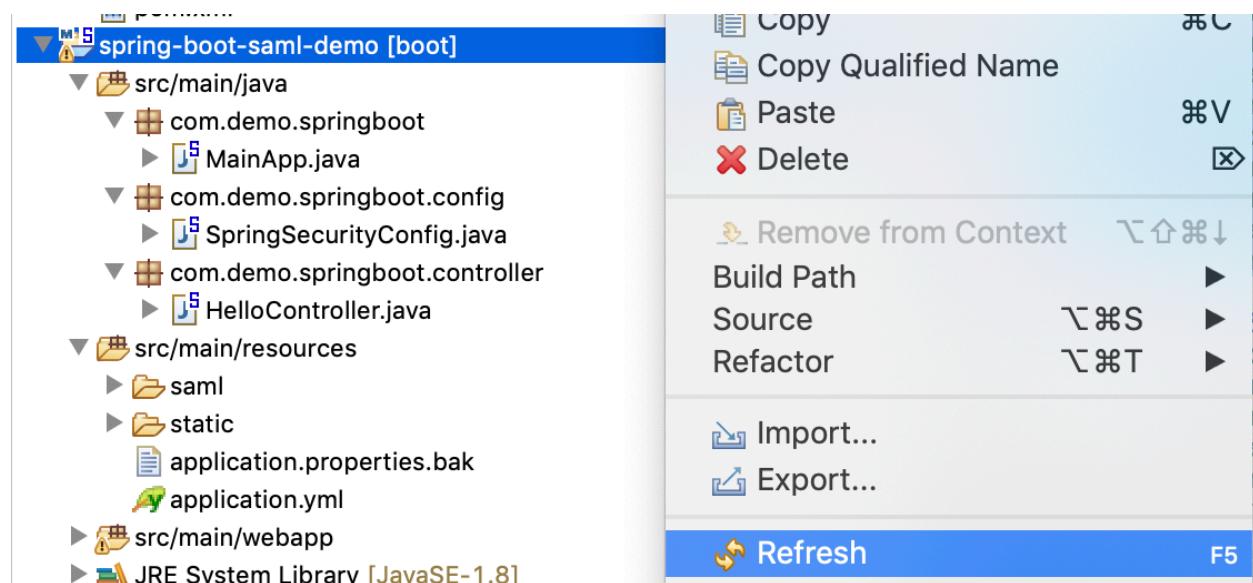
Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool -importkeystore -srckeystore keystore.jks -destkeystore keystore.jks -deststoretype pkcs12".
Nandakumars-MBP:saml nanda$ ls
```

27. Password must be **testsaml**

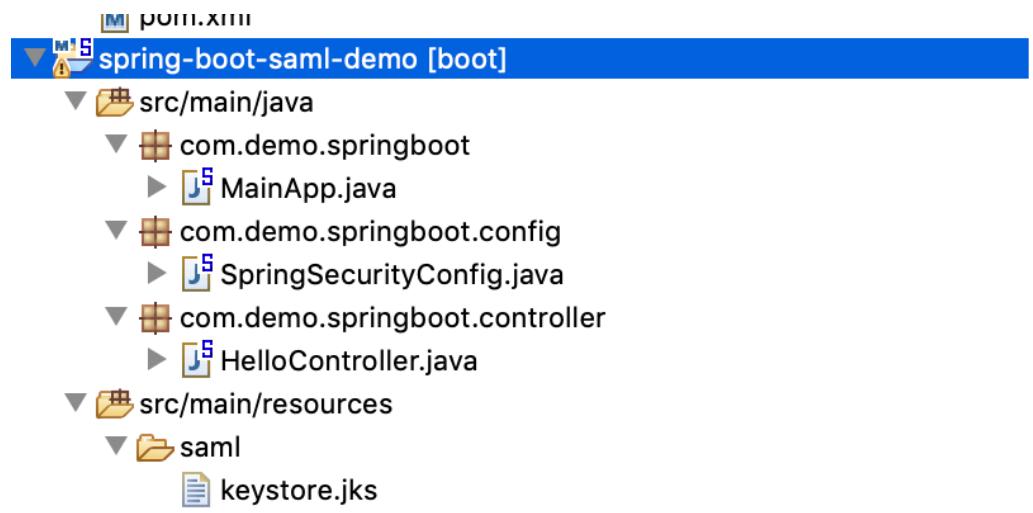
28. Check if the **keystore.jks** is created in the **saml** folder

```
# 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.EnableGlobalMethodSecurity;
import
org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import
org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

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

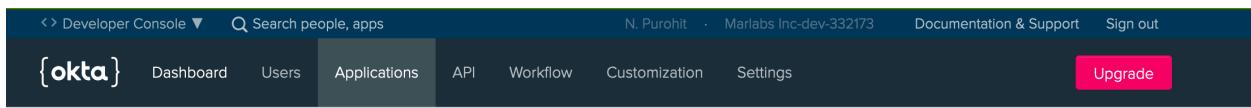
@Configuration
@EnableWebSecurity
@EnableGlobalMethodSecurity(securedEnabled = true)
public class SpringSecurityConfig extends WebSecurityConfigurerAdapter {

    //@Value("${security.saml2.metadata-url}")
    @Value("https://dev-332173.okta.com/app/exk1oua3jharcTMEfy357/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();
    }

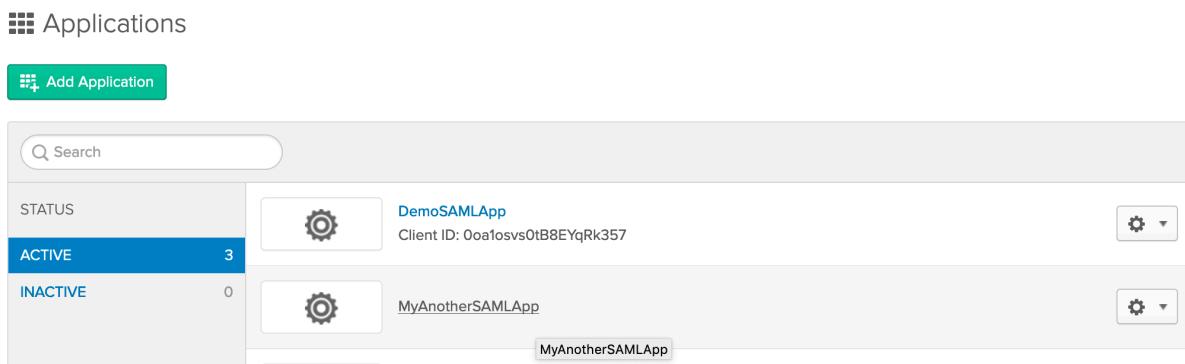
}
```

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



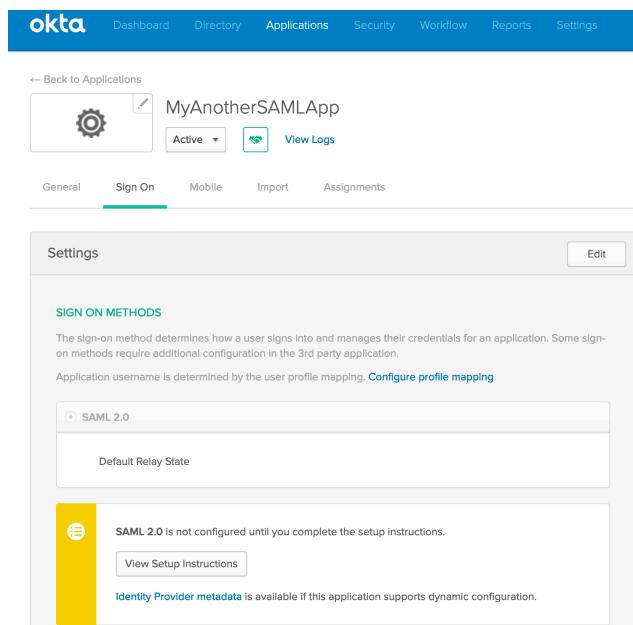
The screenshot shows the Okta Developer Console interface. The top navigation bar includes links for 'Developer Console', 'Search people, apps', 'N. Purohit', 'Marlabs Inc-dev-332173', 'Documentation & Support', 'Sign out', and a 'Upgrade' button. The main menu has tabs for 'Dashboard', 'Users', 'Applications' (which is selected), 'API', 'Workflow', 'Customization', and 'Settings'. Below the menu, there is a search bar and a button labeled 'Add Application'. The 'Applications' section displays two entries: 'DemoSAMLApp' (ACTIVE, Client ID: Ooa1osvs0tB8EYqRk357) and 'MyAnotherSAMLApp' (INACTIVE). Each application entry has a gear icon for settings and a dropdown arrow.

31.2 Click on your Application Name



The screenshot shows the 'Applications' page in the Okta developer console. At the top, there is a 'Search' bar and a green 'Add Application' button. Below the search bar, there is a table with columns for 'STATUS' (ACTIVE or INACTIVE) and 'Name' (Client ID). The table shows two rows: 'DemoSAMLApp' (ACTIVE, Client ID: Ooa1osvs0tB8EYqRk357) and 'MyAnotherSAMLApp' (INACTIVE, Client ID: MyAnotherSAMLApp). Each row has a gear icon for settings and a dropdown arrow.

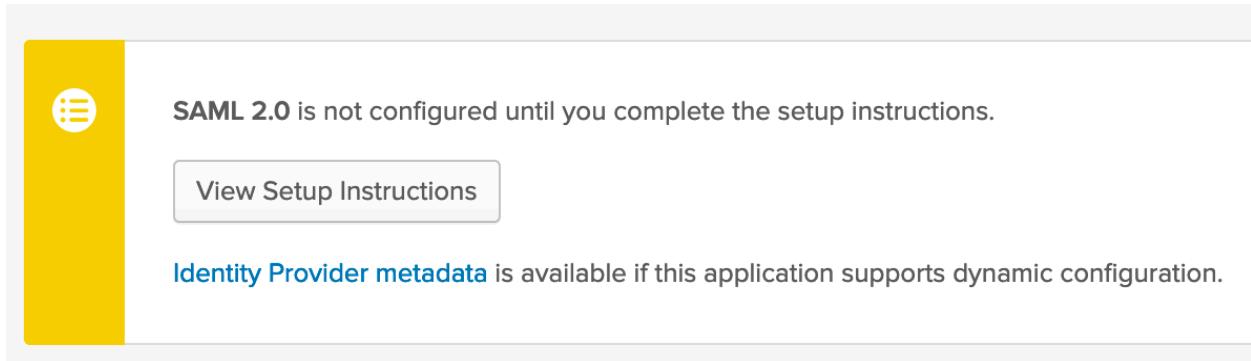
31.3 Go to **Sign On** Tab as shown here



The screenshot shows the 'Sign On' tab of the 'MyAnotherSAMLApp' application settings in the Okta developer console. The top navigation bar includes links for 'Back to Applications', 'MyAnotherSAMLApp', 'Active', 'View Logs', and tabs for 'General', 'Sign On' (which is selected), 'Mobile', 'Import', and 'Assignments'. The 'Sign On' tab has a 'Settings' section with an 'Edit' button. Below this is a 'SIGN ON METHODS' section. It shows a radio button for 'SAML 2.0' which is selected. There is also a 'Default Relay State' field. A yellow warning box states: 'SAML 2.0 is not configured until you complete the setup instructions.' with a 'View Setup Instructions' button. A note at the bottom says: 'Identity Provider metadata is available if this application supports dynamic configuration.'

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

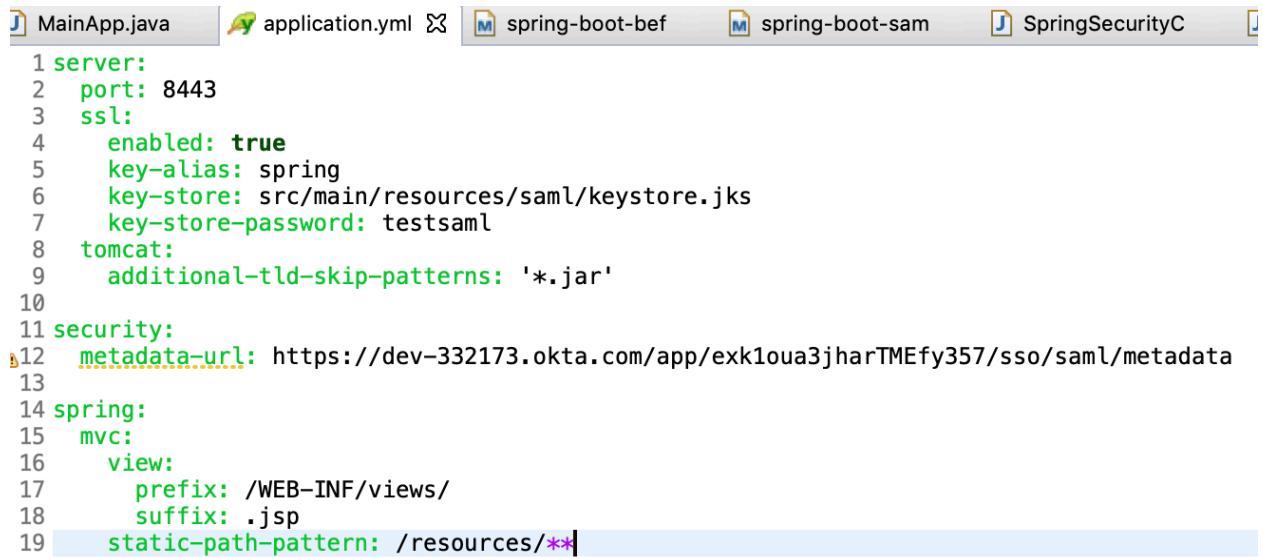
```
server:
  port: 8443
  ssl:
    enabled: true
    key-alias: spring
    key-store: src/main/resources/saml/keystore.jks
    key-store-password: testsaml
  tomcat:
    additional-tld-skip-patterns: '*.jar'

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

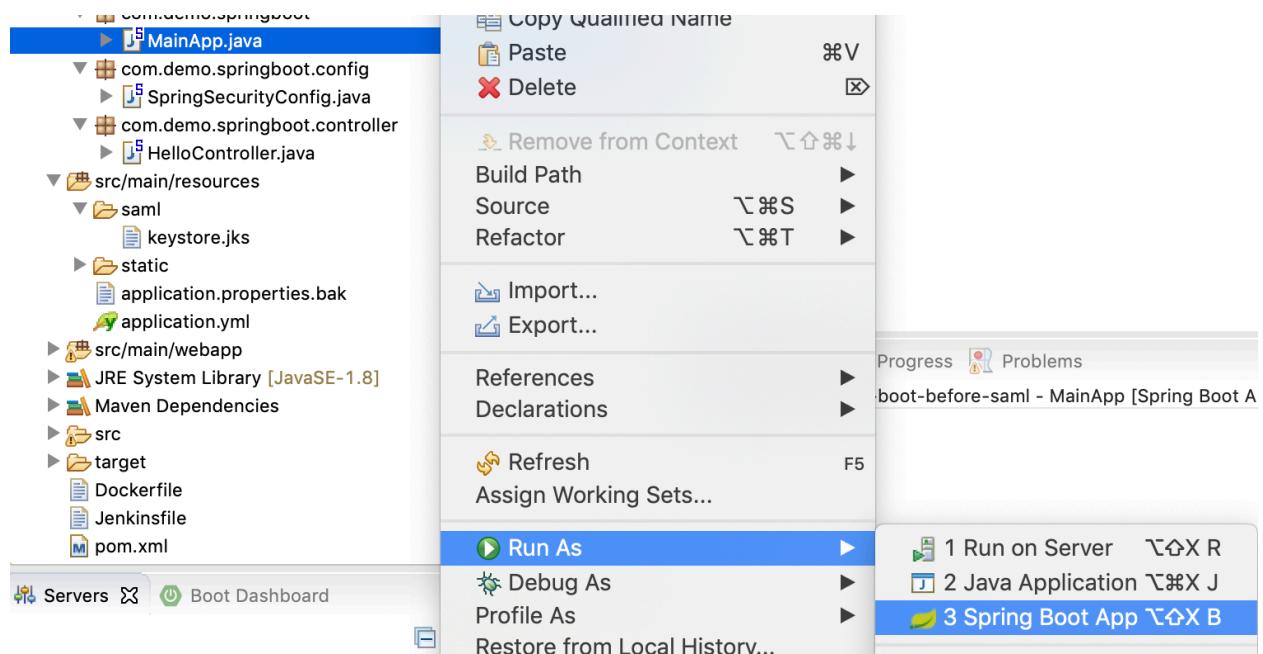
```
10
11 security:
12   metadata-url: https://dev-332173.okta.com/app/exk1oua3jharTMEfy357/sso/saml/metadata
13
```



```

1 server:
2   port: 8443
3   ssl:
4     enabled: true
5     key-alias: spring
6     key-store: src/main/resources/saml/keystore.jks
7     key-store-password: testsaml
8   tomcat:
9     additional-tld-skip-patterns: '*.jar'
10
11 security:
12   metadata-url: https://dev-332173.okta.com/app/exk1oua3jharTMEfy357/sso/saml/metadata
13
14 spring:
15   mvc:
16     view:
17       prefix: /WEB-INF/views/
18       suffix: .jsp
19     static-path-pattern: /resources/**
```

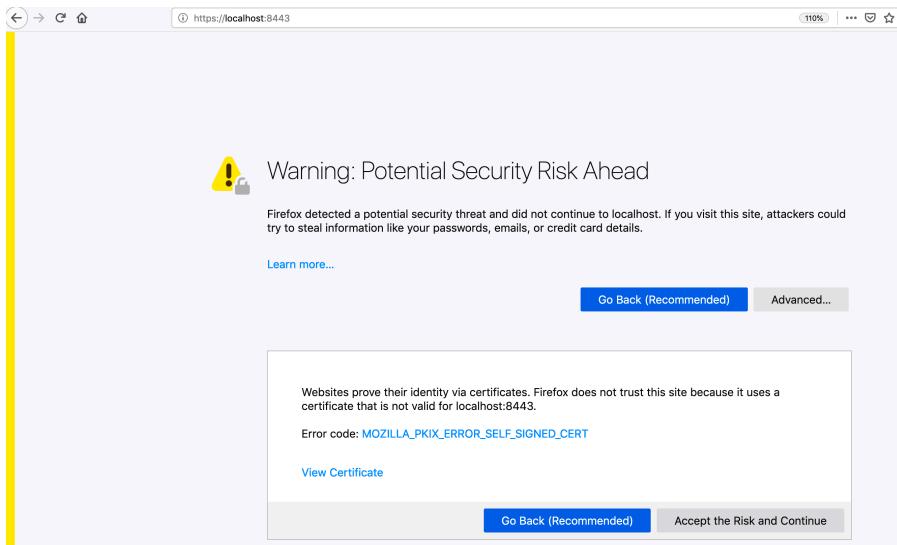
33. RUN the APP



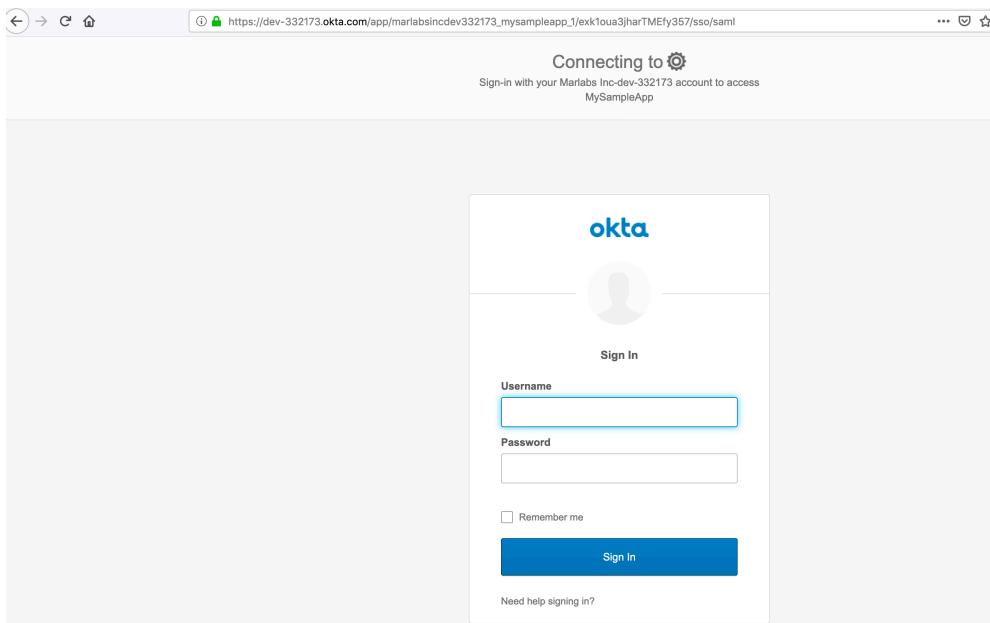
34. OUTPUT

Visit <https://localhost:8443>

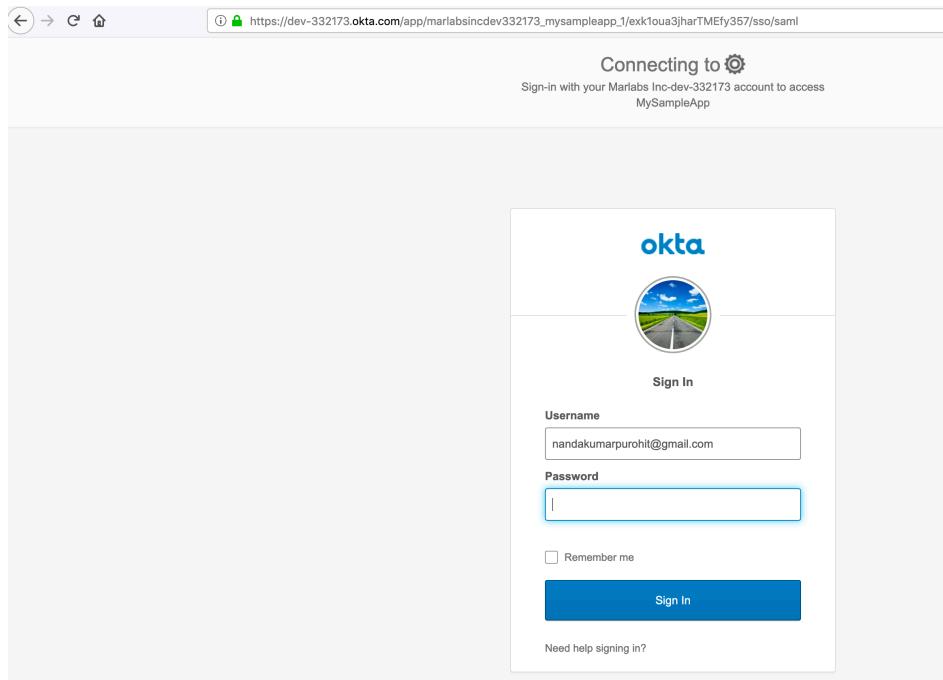
Click on Accept the Risk & Continue



35. It should redirect to your okra login page



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



well.

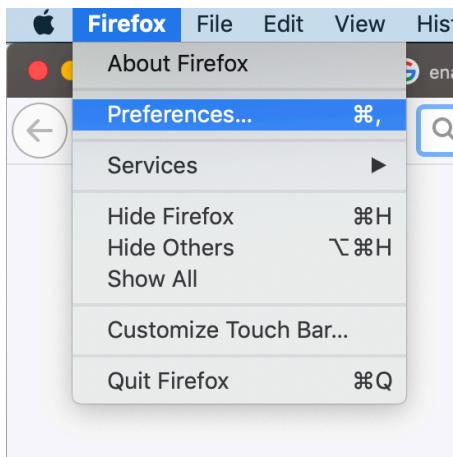
37. Enter the password as

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

A screenshot of a web browser showing a Spring Boot - MVC web application example. The URL in the address bar is https://localhost:8443. The page title is "Spring Boot - MVC web application example". The main content area displays the text "Hello Spring Boot!".

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

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



40.

A screenshot of the Firefox 'Privacy & Security' settings page. At the top, there are two radio button options: 'Only when Firefox is set to block known trackers' (selected) and 'Always ask me before blocking trackers'. Below this is a 'Sync' section. The main area is titled 'Cookies and Site Data' and displays a message: 'Your stored cookies, site data, and cache are currently using 77.8 MB of disk space.' It includes a 'Learn more' link, a 'Clear Data...' button, a 'Manage Data...' button, and a checkbox for 'Delete cookies and site data when Firefox is closed'. There is also a 'Manage Permissions...' button.

41.

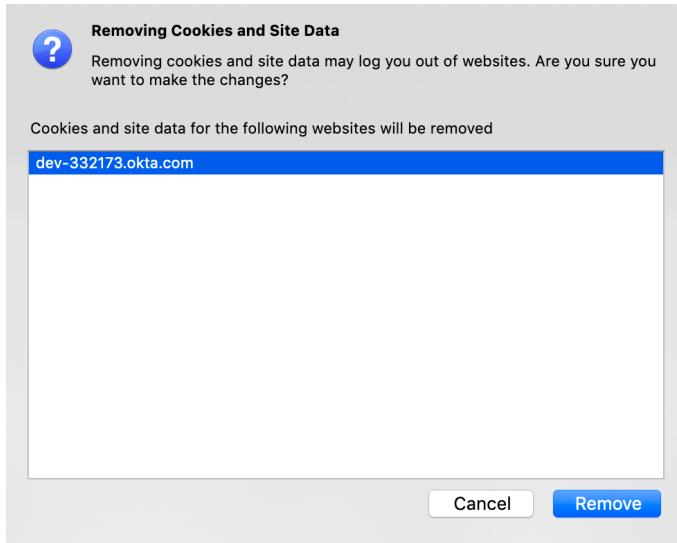
Manage Cookies and Site Data X

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.

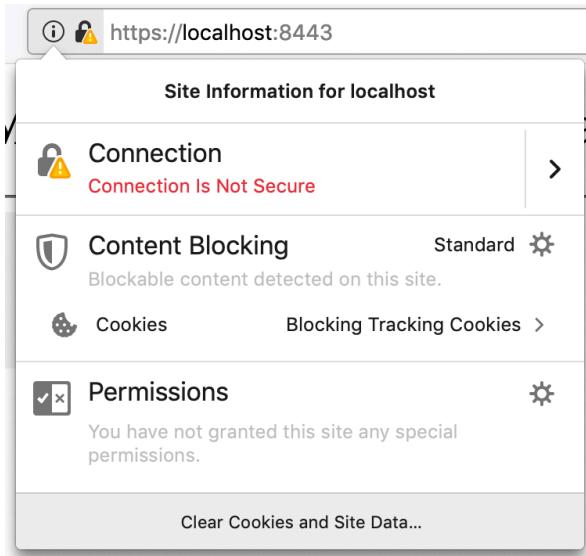
×

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

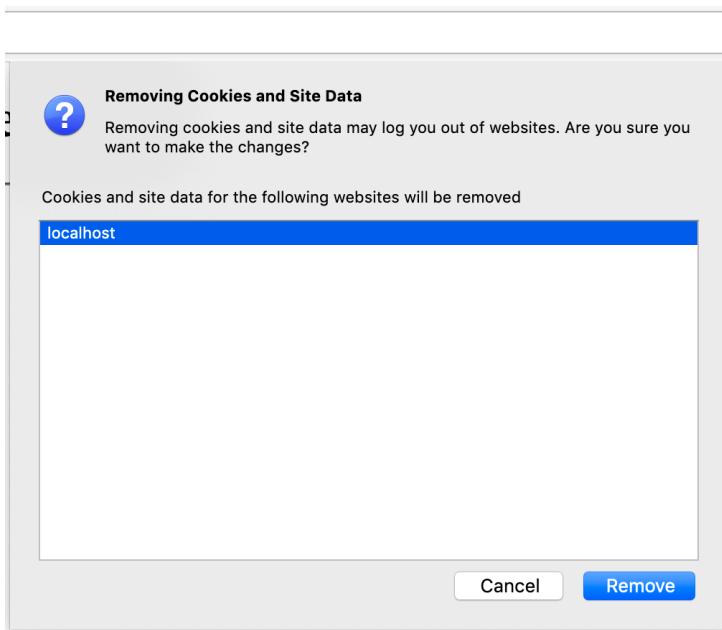
42.



43.



44.



45.



46.

