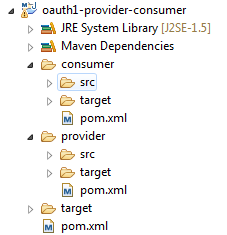
3-Legged Authorization using Spring OAUTH 1.0

# Project Structure



# **Maven Configuration for Parent(pom.xml)**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>oauth1-provider-consumer</groupId>

<artifactId>oauth1-provider-consumer</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>pom</packaging>

<name>oauth1-provider-consumer</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<java.version>1.7</java.version>

<spring.security.oauth.version>1.0.5.RELEASE</spring.security.oauth.version>

<spring.security.version>3.1.0.RELEASE</spring.security.version>

<spring.version>3.1.2.RELEASE</spring.version>

<logback.version>1.0.13</logback.version>

<servlet.version>3.0.1</servlet.version>

<jersey.version>1.18.1</jersey.version>

<slf4j.version>1.7.5</slf4j.version>

<jaxb.version>2.2.4-1</jaxb.version>

<jstl.version>1.2</jstl.version>

<jsp.version>2.2</jsp.version>

</properties>

<dependencies>

<!-- Logging Libraries -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>jcl-over-slf4j</artifactId>

<version>${slf4j.version}</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>log4j-over-slf4j</artifactId>

<version>${slf4j.version}</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>jul-to-slf4j</artifactId>

<version>${slf4j.version}</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>${slf4j.version}</version>

</dependency>

<dependency>

<groupId>ch.qos.logback</groupId>

<artifactId>logback-classic</artifactId>

<version>${logback.version}</version>

</dependency>

<!-- Servlet API -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>${servlet.version}</version>

</dependency>

<dependency>

<groupId>javax.servlet.jsp</groupId>

<artifactId>jsp-api</artifactId>

<version>${jsp.version}</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>${jstl.version}</version>

</dependency>

<!-- REST -->

<dependency>

<groupId>com.sun.jersey</groupId>

<artifactId>jersey-server</artifactId>

<version>${jersey.version}</version>

</dependency>

<dependency>

<groupId>com.sun.jersey</groupId>

<artifactId>jersey-servlet</artifactId>

<version>${jersey.version}</version>

</dependency>

<dependency>

<groupId>com.sun.jersey</groupId>

<artifactId>jersey-json</artifactId>

<version>${jersey.version}</version>

</dependency>

<dependency>

<groupId>com.sun.xml.bind</groupId>

<artifactId>jaxb-impl</artifactId>

<version>${jaxb.version}</version>

</dependency>

<dependency>

<groupId>com.sun.jersey.contribs</groupId>

<artifactId>jersey-spring</artifactId>

<version>${jersey.version}</version>

<exclusions>

<exclusion>

<artifactId>spring-aop</artifactId>

<groupId>org.springframework</groupId>

</exclusion>

<exclusion>

<artifactId>spring-web</artifactId>

<groupId>org.springframework</groupId>

</exclusion>

<exclusion>

<artifactId>spring-core</artifactId>

<groupId>org.springframework</groupId>

</exclusion>

<exclusion>

<artifactId>spring-beans</artifactId>

<groupId>org.springframework</groupId>

</exclusion>

<exclusion>

<artifactId>spring-context</artifactId>

<groupId>org.springframework</groupId>

</exclusion>

</exclusions>

</dependency>

<!-- Spring -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<!-- Spring Security -->

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-core</artifactId>

<version>${spring.security.version}</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-config</artifactId>

<version>${spring.security.version}</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-web</artifactId>

<version>${spring.security.version}</version>

</dependency>

<!-- Spring Security for OAuth 1.0a -->

<dependency>

<groupId>org.springframework.security.oauth</groupId>

<artifactId>spring-security-oauth</artifactId>

<version>${spring.security.oauth.version}</version>

</dependency>

</dependencies>

<modules>

<module>provider</module>

<module>consumer</module>

</modules>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.3</version>

<configuration>

<source>1.7</source>

<target>1.7</target>

</configuration>

</plugin>

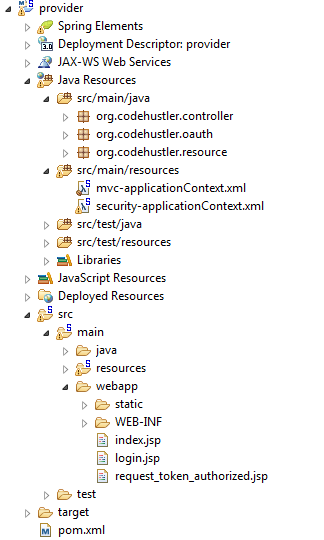
</plugins>

</build>

</project>

# **Provider Project (Auth Server)**

# Project Structure



Maven Configuration (pom.xml)

<?xml version=*"1.0"*?>

<project

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*

xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*>

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>oauth1-provider-consumer</groupId>

<artifactId>oauth1-provider-consumer</artifactId>

<version>0.0.1-SNAPSHOT</version>

</parent>

<groupId>provider</groupId>

<artifactId>provider</artifactId>

<packaging>war</packaging>

<name>provider Maven Webapp</name>

<url>http://maven.apache.org</url>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<finalName>provider</finalName>

</build>

</project>

# **Spring MVC Configuration(mvc-applicationContext.xml)**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.1.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.1.xsd*

*http://www.springframework.org/schema/mvc*

*http://www.springframework.org/schema/mvc/spring-mvc-3.1.xsd"*>

<!-- Spring Context Configuration -->

<context:annotation-config />

<context:component-scan base-package=*"org.codehustler"* />

<!-- MVC Configuration -->

<mvc:annotation-driven/>

<!-- <mvc:default-servlet-handler /> -->

<bean id=*"viewResolver"* class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"* value=*"/WEB-INF/jsp/"* />

<property name=*"suffix"* value=*".jsp"* />

</bean>

<!-- The Controller will be picked up by the component-scan. I've added it here for clarity. -->

<bean id=*"confirmAccessController"* class=*"org.codehustler.controller.ConfirmAccessController"* />

</beans>

# **Spring OAUTH1 Security Configuration(security-applicationContext.xml)**

<?xml version=*"1.0"* encoding=*"UTF-8"* ?>

<b:beans xmlns:b=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns=*"http://www.springframework.org/schema/security"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:oauth=*"http://www.springframework.org/schema/security/oauth"*

xsi:schemaLocation=*"http://www.springframework.org/schema/security*

*http://www.springframework.org/schema/security/spring-security-3.1.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.1.xsd*

*http://www.springframework.org/schema/security/oauth*

*http://www.springframework.org/schema/security/spring-security-oauth.xsd*

*http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.1.xsd"*>

<context:annotation-config />

<global-method-security pre-post-annotations=*"enabled"* />

<!-- This is here to ensure that the static content (JavaScript, CSS, etc) is accessible -->

<!-- from the login page without authentication -->

<http security=*"none"* pattern=*"/static/\*\*"* />

<!-- Form Login Security -->

<http use-expressions=*"true"* access-denied-page=*"/login.jsp"*>

<form-login authentication-failure-url=*"/login.jsp"* default-target-url=*"/index.jsp"* login-page=*"/login.jsp"*

login-processing-url=*"/login.do"* username-parameter=*"username"*

password-parameter=*"password"*/>

<logout logout-success-url=*"/login.jsp?out=1"* logout-url=*"/logout.do"*/>

<intercept-url pattern=*"/login.jsp"* access=*"permitAll()"* />

<intercept-url pattern=*"/\*\*"* access=*"hasRole('ROLE\_USER')"* />

</http>

<authentication-manager alias=*"authenticationManager"*>

<authentication-provider>

<user-service>

<user name=*"user"* password=*"user"* authorities=*"ROLE\_USER"*/>

</user-service>

</authentication-provider>

</authentication-manager>

<!-- ================================================================================== -->

<!-- OAuth 1.0 3-Legged Security -->

<!-- ================================================================================== -->

<b:bean id=*"oAuthAuthenticationHandler"* class=*"org.springframework.security.oauth.provider.DefaultAuthenticationHandler"* />

<b:bean id=*"consumerDetailsService"* class=*"org.codehustler.oauth.OAuthConsumerDetailsService"*/>

<oauth:provider consumer-details-service-ref=*"consumerDetailsService"*

token-id-param=*"oauth\_token"*

token-services-ref=*"tokenServices"*

request-token-url=*"/oauth/request\_token"*

user-approval-url=*"/oauth/confirm\_access"*

authenticate-token-url=*"/oauth/authorize\_token"*

access-token-url=*"/oauth/access\_token"*

auth-handler-ref=*"oAuthAuthenticationHandler"*

access-granted-url=*"/request\_token\_authorized.jsp"*

require10a=*"false"* />

<oauth:token-services id=*"tokenServices"* />

</b:beans>

# **Web deployment descriptor(web.xml)**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns=*"http://java.sun.com/xml/ns/javaee"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"*

version=*"3.0"* metadata-complete=*"true"*>

<!-- Spring Config File -->

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>

classpath:mvc-applicationContext.xml

classpath:security-applicationContext.xml

</param-value>

</context-param>

<!-- Spring Listeners -->

<listener>

<listener-class>org.springframework.web.context.request.RequestContextListener</listener-class>

</listener>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<!-- MVC Filter -->

<servlet>

<servlet-name>mvcDispatcher</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextAttribute</param-name>

<param-value>org.springframework.web.context.WebApplicationContext.ROOT</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>mvcDispatcher</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

<!-- Security Filter -->

<filter>

<filter-name>springSecurityFilterChain</filter-name>

<filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>

<init-param>

<param-name>contextAttribute</param-name>

<param-value>org.springframework.web.context.WebApplicationContext.ROOT</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>springSecurityFilterChain</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!--Jersey Servlet -->

<servlet>

<servlet-name>RestService</servlet-name>

<servlet-class>com.sun.jersey.spi.spring.container.servlet.SpringServlet</servlet-class>

<init-param>

<param-name>com.sun.jersey.config.property.packages</param-name>

<param-value>org.codehustler</param-value>

</init-param>

<init-param>

<param-name>com.sun.jersey.config.feature.DisableWADL</param-name>

<param-value>true</param-value>

</init-param>

<init-param>

<param-name>com.sun.jersey.api.json.POJOMappingFeature</param-name>

<param-value>true</param-value>

</init-param>

<init-param>

<param-name>com.sun.jersey.spi.container.ContainerRequestFilters</param-name>

<param-value>com.sun.jersey.api.container.filter.LoggingFilter</param-value>

</init-param>

<init-param>

<param-name>com.sun.jersey.spi.container.ContainerResponseFilters</param-name>

<param-value>com.sun.jersey.api.container.filter.LoggingFilter</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>RestService</servlet-name>

<url-pattern>/api/\*</url-pattern>

</servlet-mapping>

</web-app>

# **JSP Pages**

## **Login.jsp**

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<%@ page language=*"java"* import=*"javax.servlet.jsp.PageContext"* %>

<!DOCTYPE html>

<html>

<head>

<title>Spring Security Tutorial - 3-Legged OAuth 1.0 - Provider Server!</title>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*/>

<link href=*"*<c:url value=*'/static/fav.png'*/>*"* rel=*"shortcut icon"*>

<link href=*"*<c:url value=*'/static/style.css'*/>*"* rel=*"stylesheet"* type=*"text/css"*/>

</head>

<body>

<H1>Spring Security Tutorial - 3-Legged OAuth 1.0 - Provider Login!</H1>

<p>You must login in order to Authorize the Request Token.</p>

<p><strong>Username</strong>: user</p>

<p><strong>Password</strong>: user<br><br></p>

<form id=*"form"* action=*"*<c:url value=*'/login.do'*/>*"* method=*"POST"*>

<c:if test=*"*${**not empty** SPRING\_SECURITY\_LAST\_EXCEPTION}*"*>

<div class=*"msg-container error"*>

<c:out value=*"*${SPRING\_SECURITY\_LAST\_EXCEPTION.message}*"*/>

</div>

</c:if>

<c:if test=*"*${**not empty** param.out}*"*>

<div class=*"msg-container logout"*>

You've logged out successfully.

</div>

</c:if>

Username:<br>

<input type=*"text"* name=*"username"* value=*""* class=*"input-text input-email*<c:if test="${**not empty** SPRING\_SECURITY\_LAST\_EXCEPTION}"> *input-error*</c:if>*"*/><br><br>

Password:<br>

<input type=*"password"* name=*"password"* value=*""* class=*"input-text input-pass*<c:if test="${**not empty** SPRING\_SECURITY\_LAST\_EXCEPTION}"> *input-error*</c:if>*"*/>

<div class=*"submit-container"*>

<input value=*"Login"* name=*"submit"* type=*"submit"* class=*"submit-btn"*/>

</div>

</form>

</body>

</html>

## **Confirm\_access.jsp**

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<!DOCTYPE html>

<html>

<head>

<title>Spring Security Tutorial - 3-Legged OAuth 1.0 - Provider Server!</title>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<meta http-equiv=*"Pragma"* content=*"no-cache"*>

<meta http-equiv=*"Cache-Control"* content=*"no-cache"*>

<meta http-equiv=*"Expires"* content=*"Mon, 22 Jul 1970 11:12:01 GMT"*>

<link href=*"*<c:url value=*'/static/fav.png'*/>*"* rel=*"shortcut icon"*>

<link href=*"*<c:url value=*'/static/style.css'*/>*"* rel=*"stylesheet"* type=*"text/css"*/>

</head>

<body>

<H1>Authorize the Request Token!</H1>

<p>Please authorize the Request Token.</p>

<form action=*"*<c:url value=*"/oauth/authorize\_token"*/>*"* method=*"post"*>

<input name=*"oauth\_token"* value=*"*<c:out value="${oauth\_token}"/>*"* type=*"hidden"*/>

<c:if test=*"*${!**empty** oauth\_callback}*"*>

<input name=*"callbackURL"* value=*"*<c:out value="${oauth\_callback}"/>*"* type=*"hidden"*/>

</c:if>

<div class=*"submit-container"*>

<input value=*"Authorize"* name=*"submit"* type=*"submit"* />

</div>

</form>

</body>

</html>

# **Java Code**

## **ProtectedResource.java**

**package** org.codehustler.resource;

**import** javax.ws.rs.GET;

**import** javax.ws.rs.Path;

**import** javax.ws.rs.Produces;

**import** javax.ws.rs.core.MediaType;

**import** org.springframework.security.access.prepost.PreAuthorize;

**import** com.sun.jersey.spi.resource.Singleton;

@Singleton

@Path("/protected")

@PreAuthorize("hasRole('ROLE\_OAUTH')")

@Produces(MediaType.***APPLICATION\_JSON***)

**public** **class** ProtectedResource {

**public** ProtectedResource() { }

@GET

**public** String getOAuthProtectedData() {

**return** "{\"protected\":\"This is some OAuth protected data coming from the Provider!\"}";

}

}

## **OAuthConsumerDetails.java**

**package** org.codehustler.oauth;

**import** java.util.List;

**import** org.springframework.security.core.GrantedAuthority;

**import** org.springframework.security.oauth.common.signature.SharedConsumerSecretImpl;

**import** org.springframework.security.oauth.common.signature.SignatureSecret;

**import** org.springframework.security.oauth.provider.ConsumerDetails;

**public** **class** OAuthConsumerDetails **implements** ConsumerDetails {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** **final** String consumerName;

**private** **final** String consumerKey;

**private** **final** SignatureSecret signatureSecret;

**private** **final** List<GrantedAuthority> authorities;

**public** OAuthConsumerDetails(String consumerName, String consumerKey,

String signatureSecret, List<GrantedAuthority> authorities) {

**this**.consumerName = consumerName;

**this**.consumerKey = consumerKey;

**this**.signatureSecret = **new** SharedConsumerSecretImpl(signatureSecret);

**this**.authorities = authorities;

}

**public** String getConsumerName() {

**return** consumerName;

}

**public** String getConsumerKey() {

**return** consumerKey;

}

**public** SignatureSecret getSignatureSecret() {

**return** signatureSecret;

}

**public** List<GrantedAuthority> getAuthorities() {

**return** authorities;

}

}

## **OAuthConsumerDetailsService.java**

**package** org.codehustler.oauth;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.springframework.security.core.GrantedAuthority;

**import** org.springframework.security.core.authority.SimpleGrantedAuthority;

**import** org.springframework.security.oauth.common.OAuthException;

**import** org.springframework.security.oauth.provider.ConsumerDetails;

**import** org.springframework.security.oauth.provider.ConsumerDetailsService;

**public** **class** OAuthConsumerDetailsService **implements** ConsumerDetailsService {

String consumerName = "John";

String consumerKey = "3a4393c3da1a4e316ee66c0cc61c71";

String consumerSecret = "fe1372c074185b19c309964812bb8f3f2256ba514aea8a318f05f9d703d524b8";

**public** OAuthConsumerDetailsService() {

}

@Override

**public** ConsumerDetails loadConsumerByConsumerKey(String consumerKey)

**throws** OAuthException {

**if** (consumerKey == **null**)

**throw** **new** OAuthException(

"No credentials found for the consumer key [" + consumerKey

+ "]");

**if** (!consumerKey.equals(**this**.consumerKey))

**throw** **new** OAuthException(

"No credentials found for the consumer key [" + consumerKey

+ "]");

List<GrantedAuthority> authorities = **new** ArrayList<GrantedAuthority>();

authorities.add(**new** SimpleGrantedAuthority("ROLE\_OAUTH"));

**return** **new** OAuthConsumerDetails(consumerName, consumerKey,

consumerSecret, authorities);

}

}

## **ConfirmAccessController.java**

**package** org.codehustler.controller;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.servlet.ModelAndView;

@Controller

**public** **class** ConfirmAccessController

{

**public** ConfirmAccessController() {}

@RequestMapping( value = "/oauth/confirm\_access", method = RequestMethod.***GET*** )

**public** ModelAndView confirmAccess( HttpServletRequest request, HttpServletResponse response )

{

String token = request.getParameter( "oauth\_token" );

String callback = request.getParameter( "oauth\_callback" );

**if**( token == **null** )

**throw** **new** IllegalArgumentException("No Request Token provided for authorization.");

ModelAndView mav = **new** ModelAndView( "confirm\_access" );

mav.addObject( "oauth\_token", token );

mav.addObject( "oauth\_callback", callback );

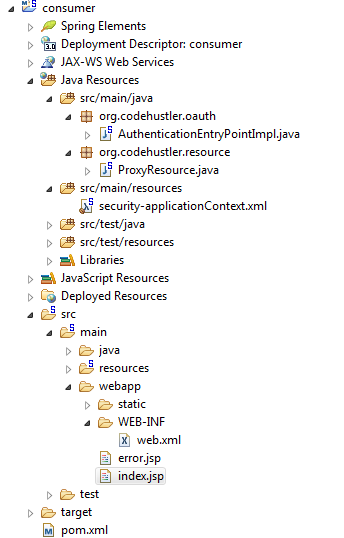
**return** mav;

}

}

## **Consumer**

## **Project Structure**



## **Maven Configuration(pom.xml)**

Same as provider.

## **JSP Pages**

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<!DOCTYPE html>

<html>

<head>

<title>Spring Security Tutorial - 3-Legged OAuth 1.0 - Consumer Server!</title>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<meta http-equiv=*"Pragma"* content=*"no-cache"*>

<meta http-equiv=*"Cache-Control"* content=*"no-cache"*>

<meta http-equiv=*"Expires"* content=*"Mon, 22 Jul 1970 11:12:01 GMT"*>

<link href=*"*<c:url value=*'/static/fav.png'*/>*"* rel=*"shortcut icon"*>

<link href=*"*<c:url value=*'/static/style.css'*/>*"* rel=*"stylesheet"* type=*"text/css"*/>

</head>

<body>

<H1>Spring Security Tutorial - 3-Legged OAuth 1.0 - Consumer Server!</H1>

<p>Click the link below to make a call to the Provider Server's protected API end point.</p>

<p><a href=*"*<c:url value=*"/api/proxy"*/>*"*>Get Protected Resource</a></p>

</body>

</html>

## **Deployment Descriptor(web.xml)**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns=*"http://java.sun.com/xml/ns/javaee"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"*

version=*"3.0"* metadata-complete=*"true"*>

<!-- Spring Config File -->

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>

classpath:security-applicationContext.xml

</param-value>

</context-param>

<!-- Spring Listeners -->

<listener>

<listener-class>org.springframework.web.context.request.RequestContextListener</listener-class>

</listener>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<!-- Security Filter -->

<filter>

<filter-name>springSecurityFilterChain</filter-name>

<filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>

<init-param>

<param-name>contextAttribute</param-name>

<param-value>org.springframework.web.context.WebApplicationContext.ROOT</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>springSecurityFilterChain</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!--Jersey Servlet -->

<servlet>

<servlet-name>RestService</servlet-name>

<servlet-class>com.sun.jersey.spi.spring.container.servlet.SpringServlet</servlet-class>

<init-param>

<param-name>com.sun.jersey.config.property.packages</param-name>

<param-value>org.codehustler.resource</param-value>

</init-param>

<init-param>

<param-name>com.sun.jersey.config.feature.DisableWADL</param-name>

<param-value>true</param-value>

</init-param>

<init-param>

<param-name>com.sun.jersey.api.json.POJOMappingFeature</param-name>

<param-value>true</param-value>

</init-param>

<init-param>

<param-name>com.sun.jersey.spi.container.ContainerRequestFilters</param-name>

<param-value>com.sun.jersey.api.container.filter.LoggingFilter</param-value>

</init-param>

<init-param>

<param-name>com.sun.jersey.spi.container.ContainerResponseFilters</param-name>

<param-value>com.sun.jersey.api.container.filter.LoggingFilter</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>RestService</servlet-name>

<url-pattern>/api/\*</url-pattern>

</servlet-mapping>

<!-- JSPs -->

<servlet>

<servlet-name>IndexJsp</servlet-name>

<jsp-file>/index.jsp</jsp-file>

</servlet>

<servlet-mapping>

<servlet-name>IndexJsp</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

<!-- These need to be mapped against the default servlet for the CSS / JS

to work -->

<servlet-mapping>

<servlet-name>default</servlet-name>

<url-pattern>/static/\*</url-pattern>

</servlet-mapping>

</web-app>

## **Spring Security Configuration(security-applicationContext.xml)**

<?xml version=*"1.0"* encoding=*"UTF-8"* ?>

<b:beans xmlns:b=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns=*"http://www.springframework.org/schema/security"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:oauth=*"http://www.springframework.org/schema/security/oauth"*

xsi:schemaLocation=*"http://www.springframework.org/schema/security*

*http://www.springframework.org/schema/security/spring-security-3.1.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.1.xsd*

*http://www.springframework.org/schema/security/oauth*

*http://www.springframework.org/schema/security/spring-security-oauth.xsd*

*http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.1.xsd"*>

<context:annotation-config />

<context:component-scan base-package=*"org.codehustler"* />

<!-- ================================================================================== -->

<!-- Static Content Security -->

<!-- ================================================================================== -->

<!-- This is here to ensure that the static content (JavaScript, CSS, etc) is accessible -->

<!-- from the login page without authentication -->

<http security=*"none"* pattern=*"/static/\*\*"* />

<!-- ================================================================================== -->

<!-- Client Security Configuration -->

<!-- ================================================================================== -->

<http use-expressions=*"true"* entry-point-ref=*"authenticationEntryPoint"*>

<intercept-url pattern=*"/\*\*"* access=*"permitAll()"* />

</http>

<b:bean id=*"authenticationEntryPoint"* class=*"org.codehustler.oauth.AuthenticationEntryPointImpl"* />

<authentication-manager alias=*"authenticationManager"* />

<!-- ================================================================================== -->

<!-- OAuth 1.0 3-Legged Client -->

<!-- ================================================================================== -->

<oauth:consumer resource-details-service-ref=*"resourceDetails"* requireAuthenticated=*"false"* oauth-failure-page=*"/error.jsp"* >

<oauth:url pattern=*"/api/\*\*"* resources=*"protectedResource"*/>

</oauth:consumer>

<oauth:resource-details-service id=*"resourceDetails"*>

<oauth:resource id=*"protectedResource"*

key=*"3a4393c3da1a4e316ee66c0cc61c71"*

secret=*"fe1372c074185b19c309964812bb8f3f2256ba514aea8a318f05f9d703d524b8"*

request-token-url=*"http://localhost:8080/provider/oauth/request\_token"*

user-authorization-url=*"http://localhost:8080/provider/oauth/confirm\_access"*

access-token-url=*"http://localhost:8080/provider/oauth/access\_token"*

user-authorization-token-param=*"oauth\_token"*/>

</oauth:resource-details-service>

<b:bean id=*"oauthRestTemplate"* class=*"org.springframework.security.oauth.consumer.client.OAuthRestTemplate"*>

<b:constructor-arg ref=*"protectedResource"* />

</b:bean>

</b:beans>

## **Java Code**

## **AuthenticationEntryPointImpl.java**

**package** org.codehustler.oauth;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** org.springframework.security.core.AuthenticationException;

**import** org.springframework.security.web.AuthenticationEntryPoint;

**public** **class** AuthenticationEntryPointImpl **implements** AuthenticationEntryPoint {

**public** **void** commence(HttpServletRequest request,

HttpServletResponse response, AuthenticationException failure)

**throws** IOException, ServletException {

}

}

## **ProxyResource.java**

**package** org.codehustler.resource;

**import** java.net.URI;

**import** javax.ws.rs.Consumes;

**import** javax.ws.rs.GET;

**import** javax.ws.rs.Path;

**import** javax.ws.rs.Produces;

**import** javax.ws.rs.core.MediaType;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.security.oauth.consumer.client.OAuthRestTemplate;

**import** org.springframework.stereotype.Component;

**import** com.sun.jersey.spi.resource.Singleton;

@Singleton

@Component

@Path("/proxy")

@Consumes(MediaType.***APPLICATION\_JSON***)

@Produces(MediaType.***APPLICATION\_JSON***)

**public** **class** ProxyResource {

**private** String endpointUrl = "http://localhost:8080/provider/api/protected";

@Autowired

**private** OAuthRestTemplate oauthRestTemplate;

**public** ProxyResource() {

}

@GET

**public** String fetchOAuthProtectedData() {

**try** {

**return** oauthRestTemplate.getForObject(URI.*create*(endpointUrl),

String.**class**);

} **catch** (Exception ex) {

**return** ex.toString();

}

}

}