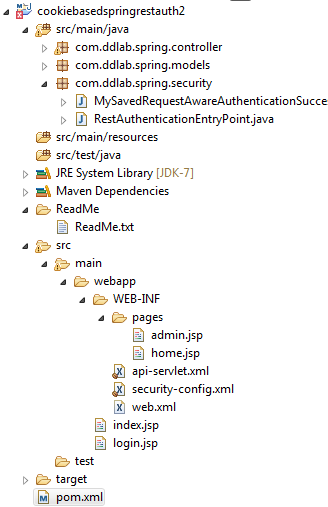
Cookie Based Authentication in Restful Web Service in Spring Security

## Project Structure



## Maven pom.xml

<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/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>cookiebasedspringrestauth2</groupId>

<artifactId>cookiebasedspringrestauth2</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>cookiebasedspringrestauth2 Maven Webapp</name>

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

<properties>

<!-- Spring -->

<org.springframework.version>4.0.5.RELEASE</org.springframework.version>

<org.springframework.security.version>3.2.4.RELEASE</org.springframework.security.version>

<!-- persistence -->

<hibernate.version>4.3.5.Final</hibernate.version>

<mysql-connector-java.version>5.1.30</mysql-connector-java.version>

<!-- logging -->

<org.slf4j.version>1.7.6</org.slf4j.version>

<logback.version>1.1.1</logback.version>

<!-- various -->

<hibernate-validator.version>5.1.1.Final</hibernate-validator.version>

<!-- util -->

<guava.version>17.0</guava.version>

<commons-lang3.version>3.3.2</commons-lang3.version>

<!-- testing -->

<org.hamcrest.version>1.3</org.hamcrest.version>

<junit.version>4.11</junit.version>

<mockito.version>1.9.5</mockito.version>

<httpcore.version>4.3.2</httpcore.version>

<httpclient.version>4.3.3</httpclient.version>

<rest-assured.version>2.3.1</rest-assured.version>

<!-- Maven plugins -->

<maven-compiler-plugin.version>3.1</maven-compiler-plugin.version>

<maven-war-plugin.version>2.4</maven-war-plugin.version>

<maven-surefire-plugin.version>2.17</maven-surefire-plugin.version>

<cargo-maven2-plugin.version>1.4.8</cargo-maven2-plugin.version>

</properties>

<dependencies>

<!-- Spring Security -->

<dependency>

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

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

<version>${org.springframework.security.version}</version>

</dependency>

<dependency>

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

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

<version>${org.springframework.security.version}</version>

</dependency>

<!-- Spring -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${org.springframework.version}</version>

<exclusions>

<exclusion>

<artifactId>commons-logging</artifactId>

<groupId>commons-logging</groupId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-expression</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<!-- web -->

<dependency>

<groupId>javax.servlet</groupId>

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

<version>3.0.1</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

<scope>runtime</scope>

</dependency>

<!-- marshalling -->

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-core</artifactId>

<version>2.2.3</version>

</dependency>

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

<version>2.2.3</version>

</dependency>

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-annotations</artifactId>

<version>2.2.3</version>

</dependency>

<!-- util -->

<dependency>

<groupId>com.google.guava</groupId>

<artifactId>guava</artifactId>

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

</dependency>

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons-lang3</artifactId>

<version>3.1</version>

</dependency>

<!-- logging -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

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

</dependency>

<dependency>

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

<artifactId>logback-classic</artifactId>

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

<!-- <scope>runtime</scope> -->

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

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

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

<!-- <scope>runtime</scope> --> <!-- some spring dependencies need to compile against jcl -->

</dependency>

<dependency> <!-- needed to bridge to slf4j for projects that use the log4j APIs directly -->

<groupId>org.slf4j</groupId>

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

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

</dependency>

<!-- test scoped -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit-dep</artifactId>

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

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.hamcrest</groupId>

<artifactId>hamcrest-core</artifactId>

<version>${org.hamcrest.version}</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.hamcrest</groupId>

<artifactId>hamcrest-library</artifactId>

<version>${org.hamcrest.version}</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

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

<scope>test</scope>

</dependency>

</dependencies>

<build>

<finalName>cookiebasedspringrestauth2</finalName>

</build>

</project>

## Spring Configuration (WEB-INF)

### Api-servlet.xml

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

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

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

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

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

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-3.0.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd*

*http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.0.xsd*

*http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.0.xsd*

*http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd"*>

<context:component-scan base-package=*"com.ddlab.spring.\*"* />

<context:annotation-config />

<mvc:annotation-driven />

<bean

class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"*>

<value>/WEB-INF/pages/</value>

</property>

<property name=*"suffix"*>

<value>.jsp</value>

</property>

</bean>

</beans>

### Security-config.xml

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

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

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

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

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

xsi:schemaLocation=*"*

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

*http://www.springframework.org/schema/security/spring-security-3.2.xsd*

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

*http://www.springframework.org/schema/beans/spring-beans-4.0.xsd"*>

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

<beans:bean id=*"restAuthenticationEntryPoint"* class=*"com.ddlab.spring.security.RestAuthenticationEntryPoint"* />

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

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

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

<!-- <intercept-url pattern="/" access="permitAll" /> -->

<!-- <sec:form-login authentication-success-handler-ref="mySuccessHandler" authentication-failure-handler-ref="myFailureHandler" username-parameter="username" password-parameter="password"/> -->

<sec:form-login authentication-success-handler-ref=*"mySuccessHandler"*

authentication-failure-handler-ref=*"myFailureHandler"* />

<logout />

<!-- <logout logout-url="/mylogout"/> -->

</http>

<beans:bean id=*"mySuccessHandler"* class=*"com.ddlab.spring.security.MySavedRequestAwareAuthenticationSuccessHandler"* />

<beans:bean id=*"myFailureHandler"* class=*"org.springframework.security.web.authentication.SimpleUrlAuthenticationFailureHandler"* />

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

<authentication-provider>

<user-service>

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

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

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

</user-service>

</authentication-provider>

</authentication-manager>

<beans:bean id=*"bankApiImpl"* class=*"com.ddlab.spring.models.BankApiImpl"*/>

</beans:beans>

### Web.xml

<web-app id=*"WebApp\_ID"* version=*"2.4"*

xmlns=*"http://java.sun.com/xml/ns/j2ee"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/j2ee*

*http://java.sun.com/xml/ns/j2ee/web-app\_2\_4.xsd"*>

<display-name>spring-security-login</display-name>

<servlet>

<servlet-name>api</servlet-name>

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

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

</servlet>

<servlet-mapping>

<servlet-name>api</servlet-name>

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

</servlet-mapping>

<listener>

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

</listener>

<context-param>

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

<param-value>

/WEB-INF/security-config.xml

</param-value>

</context-param>

<!-- Spring Security -->

<filter>

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

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

</filter>

<filter-mapping>

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

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

</filter-mapping>

<welcome-file-list>

<welcome-file>login.jsp</welcome-file>

</welcome-file-list>

</web-app>

## Jsp Pages

### Index.jsp

<html>

<body>

<h2>Hello World!</h2>

</body>

<a href=*"/login.jsp"*>Login</a>

</html>

### Login.jsp

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

<html>

<head>

<title>Login</title>

</head>

<body>

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

<font color=*"red"*>

Login error. <br />

Reason : ${sessionScope["SPRING\_SECURITY\_LAST\_EXCEPTION"].message}

</font>

</c:if>

<form method=*"POST"* action=*"*<c:url value=*"/j\_spring\_security\_check"* />*"*>

<table>

<tr>

<td align=*"right"*>Username</td>

<td><input type=*"text"* name=*"j\_username"* /></td>

</tr>

<tr>

<td align=*"right"*>Password</td>

<td><input type=*"password"* name=*"j\_password"* /></td>

</tr>

<tr>

<td colspan=*"2"* align=*"right"*>

<input type=*"submit"* value=*"Login"* />

</td>

</tr>

</table>

</form>

</body>

</html>

### Admin.jsp (WEB-INF/pages/admin.jsp)

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

<%@ page contentType=*"text/html;charset=UTF-8"* language=*"java"* %>

<html>

<head>

<title>Admin</title>

</head>

<body>

<a href=<c:url value=*"/j\_spring\_security\_logout"*/>>Logout</a><br/>

<h1>Only Admin allowed here</h1>

</body>

</html>

### Home.jsp (Web-INF/pages/home.jsp)

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

<html>

<head>

<title>Home</title>

</head>

<body>

<a href=<c:url value=*"/j\_spring\_security\_logout"*/>>Logout</a><br/>

<h1>Welcome, This page is for normal user</h1>

</body>

</html>

## Java Code

### BankApiController.java

**package** com.ddlab.spring.controller;

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

**import** org.springframework.http.MediaType;

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

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

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

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

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

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

**import** com.ddlab.spring.models.BankApi;

**import** com.ddlab.spring.models.BankCustomerModel;

**import** com.ddlab.spring.models.UserNamePasswordModel;

**import** com.ddlab.spring.models.WithdrawAmountModel;

@Controller

@RequestMapping(value = "/bankapi")

**public** **class** BankApiController {

@Autowired

**private** BankApi bankApi;

@RequestMapping(value = "/withdraw/{accountNo}/{amount}", method = RequestMethod.***GET***, produces = { MediaType.***APPLICATION\_JSON\_VALUE*** })

@ResponseBody()

**public** WithdrawAmountModel withdraw(

@PathVariable(value = "accountNo") String accountNo,

@PathVariable(value = "amount") Integer amount) {

**return** bankApi.withdrawAmount(accountNo, amount);

}

@RequestMapping(value = "/reset", method = RequestMethod.***PUT***, produces = { MediaType.***APPLICATION\_JSON\_VALUE*** })

@ResponseBody()

**public** UserNamePasswordModel resetPasswod(@RequestBody BankCustomerModel customer) {

**return** bankApi.resetPasswod(customer);

}

}

### HomePageController.java

package com.ddlab.spring.controller;

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 HomePageController {

@RequestMapping(value = "/home" , method = RequestMethod.GET)

public ModelAndView getHomePage() {

ModelAndView model = new ModelAndView();

model.setViewName("home");

return model;

}

}

### BankApi.java

package com.ddlab.spring.models;

import org.springframework.security.access.annotation.Secured;

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

public interface BankApi {

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

public WithdrawAmountModel withdrawAmount(String accountNo,int amount);

@Secured({"ROLE\_ADMIN"})

public UserNamePasswordModel resetPasswod(BankCustomerModel customer);

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

public String approveLoan(BankCustomerModel customer);

}

### BankApiImpl.java

**package** com.ddlab.spring.models;

**public** **class** BankApiImpl **implements** BankApi {

@Override

**public** WithdrawAmountModel withdrawAmount(String accountNo, **int** amount) {

WithdrawAmountModel amountModel = **new** WithdrawAmountModel();

amountModel.setAmountWithdrawn(amount);

amountModel.setAvailableBalance(50000);

**return** amountModel;

}

@Override

**public** UserNamePasswordModel resetPasswod(BankCustomerModel customer) {

UserNamePasswordModel unamePwdModel = **new** UserNamePasswordModel();

unamePwdModel.setPassword("abcd$pass00rd");

unamePwdModel.setUserName(customer.getUserName());

**return** unamePwdModel;

}

@Override

**public** String approveLoan(BankCustomerModel customer) {

**return** "The loan account for "

+ customer.getUserName()

+ " has been approved and \n+ loan account has been created in the system";

}

}

### BankCustomerModel.java

**package** com.ddlab.spring.models;

**public** **class** BankCustomerModel {

**private** String accountNo;

**private** String userName;

**private** String userId;

**public** String getAccountNo() {

**return** accountNo;

}

**public** **void** setAccountNo(String accountNo) {

**this**.accountNo = accountNo;

}

**public** String getUserName() {

**return** userName;

}

**public** **void** setUserName(String userName) {

**this**.userName = userName;

}

**public** String getUserId() {

**return** userId;

}

**public** **void** setUserId(String userId) {

**this**.userId = userId;

}

}

### UserNamePasswordModel.java

package com.ddlab.spring.models;

public class UserNamePasswordModel {

private String userName;

private String password;

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

}

### WithdrawAmountModel.java

**package** com.ddlab.spring.models;

**public** **class** WithdrawAmountModel {

**private** **int** amountWithdrawn;

**private** **int** availableBalance;

**public** **int** getAmountWithdrawn() {

**return** amountWithdrawn;

}

**public** **void** setAmountWithdrawn(**int** amountWithdrawn) {

**this**.amountWithdrawn = amountWithdrawn;

}

**public** **int** getAvailableBalance() {

**return** availableBalance;

}

**public** **void** setAvailableBalance(**int** availableBalance) {

**this**.availableBalance = availableBalance;

}

}

### MySavedRequestAwareAuthenticationSuccessHandler.java

package com.ddlab.spring.security;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.security.core.Authentication;

import org.springframework.security.web.authentication.SimpleUrlAuthenticationSuccessHandler;

import org.springframework.security.web.savedrequest.HttpSessionRequestCache;

import org.springframework.security.web.savedrequest.RequestCache;

import org.springframework.security.web.savedrequest.SavedRequest;

import org.springframework.util.StringUtils;

public class MySavedRequestAwareAuthenticationSuccessHandler extends SimpleUrlAuthenticationSuccessHandler {

private RequestCache requestCache = new HttpSessionRequestCache();

@Override

public void onAuthenticationSuccess(final HttpServletRequest request, final HttpServletResponse response, final Authentication authentication) throws ServletException, IOException {

final SavedRequest savedRequest = requestCache.getRequest(request, response);

if (savedRequest == null) {

clearAuthenticationAttributes(request);

setDefaultTargetUrl("/api/home");

getRedirectStrategy().sendRedirect(request, response, "/api/home");

return;

}

final String targetUrlParameter = getTargetUrlParameter();

if (isAlwaysUseDefaultTargetUrl() || (targetUrlParameter != null && StringUtils.hasText(request.getParameter(targetUrlParameter)))) {

requestCache.removeRequest(request, response);

clearAuthenticationAttributes(request);

return;

}

clearAuthenticationAttributes(request);

// Use the DefaultSavedRequest URL

// final String targetUrl = savedRequest.getRedirectUrl();

// logger.debug("Redirecting to DefaultSavedRequest Url: " + targetUrl);

// getRedirectStrategy().sendRedirect(request, response, targetUrl);

}

public void setRequestCache(final RequestCache requestCache) {

this.requestCache = requestCache;

}

}

### RestAuthenticationEntryPoint.java

package com.ddlab.spring.security;

import java.io.IOException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.security.core.AuthenticationException;

import org.springframework.security.web.AuthenticationEntryPoint;

import org.springframework.stereotype.Component;

/\*\*

\* The Entry Point will not redirect to any sort of Login - it will return the 401

\*/

@Component

public final class RestAuthenticationEntryPoint implements AuthenticationEntryPoint {

@Override

public void commence(final HttpServletRequest request, final HttpServletResponse response, final AuthenticationException authException) throws IOException {

response.sendError(HttpServletResponse.SC\_UNAUTHORIZED, "Unauthorized");

}

}

# How to test

1. First make a rest using post request in rest client in firefox.

**POST**

<http://localhost:8080/cookiebasedspringrestauth2/j_spring_security_check?j_username=d&j_password=d>

1. Then make subsequent call using the following url.

**GET**

<http://localhost:8080/cookiebasedspringrestauth2/api/bankapi/withdraw/12345/500>

1. Login as admin and reset the password. To login change the first url with admin username and password.

**PUT**

<http://localhost:8080/cookiebasedspringrestauth2/api/bankapi/reset>

#### Request Payload

{

"accountNo": "12345",

"userName": "Deb",

"userId":"222"

}

## Logout

**GET :** <http://localhost:8080/cookiebasedspringrestauth2/j_spring_security_logout>

To login in the main browser, use the following url.

[**http://localhost:8080/cookiebasedspringrestauth2/**](http://localhost:8080/cookiebasedspringrestauth2/)

## Test using HttpClient

You can use http client to test the cookie based authentication.

Use the following maven dependency for apache http client.

<dependency>

<groupId>org.apache.httpcomponents</groupId>

<artifactId>httpclient</artifactId>

<version>4.3.6</version>

</dependency>

### Java code to test

**package** cookiebasedspringrestauth2;

**import** org.apache.http.Header;

**import** org.apache.http.HttpEntity;

**import** org.apache.http.HttpResponse;

**import** org.apache.http.client.HttpClient;

**import** org.apache.http.client.methods.HttpGet;

**import** org.apache.http.client.methods.HttpPost;

**import** org.apache.http.impl.client.HttpClientBuilder;

**import** org.apache.http.util.EntityUtils;

**public** **class** TestUsingHttpClient {

**public** **static** **void** main(String[] args) **throws** Exception {

String url = "http://localhost:8080/cookiebasedspringrestauth2/j\_spring\_security\_check?j\_username=d&j\_password=d";

HttpPost post = **new** HttpPost(url);

HttpClient client = HttpClientBuilder.*create*().build();

HttpResponse response = client.execute(post);

Header[] headers = response.getHeaders("Set-Cookie");

**for** (Header h : headers) {

System.***out***.println(h.getValue().toString());

}

System.***out***.println("---------Handshake complete----------");

url = "http://localhost:8080/cookiebasedspringrestauth2/api/bankapi/withdraw/12345/500";

HttpGet get = **new** HttpGet(url);

HttpResponse response1 = client.execute(get);

System.***out***.println("Response :::"+response1);

HttpEntity entity = response1.getEntity();

String responseString = EntityUtils.*toString*(entity, "UTF-8");

System.***out***.println("===========>"+responseString);

}

}