Custom Roles in Spring Security

# **Maven Configuration(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/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>spring-custom-role2</groupId>

<artifactId>spring-custom-role2</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>spring-custom-role1</name>

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

<properties>

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

<spring.version>3.2.8.RELEASE</spring.version>

<spring.security.version>3.2.3.RELEASE</spring.security.version>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</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-jdbc</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-web</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-taglibs</artifactId>

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

</dependency>

</dependencies>

</project>

# **Spring Security Configuration(beans.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:aop=*"http://www.springframework.org/schema/aop"*

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

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

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

xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.1.xsd*

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

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

**<bean id=*"roleHierarchy"* class=*"org.springframework.security.access.hierarchicalroles.RoleHierarchyImpl"*>**

**<property name=*"hierarchy"*>**

**<value>**

**APP\_ADMIN > APP\_MANAGER**

**APP\_MANAGER > APP\_TELLER**

**APP\_TELLER > APP\_USER**

**APP\_GOVT**

**</value>**

**</property>**

**</bean>**

**<bean id=*"defaultMethodSecurityExpressionHandler"***

**class=*"org.springframework.security.access.expression.method.DefaultMethodSecurityExpressionHandler"*>**

**<property name=*"roleHierarchy"* ref=*"roleHierarchy"* />**

**</bean>**

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

**<security:expression-handler ref=*"defaultMethodSecurityExpressionHandler"* />**

**</security:global-method-security>**

<bean id=*"bankTxn"* class=*"com.ddlab.rnd.spring.security.BankTransactionImpl"* />

</beans>

**Note : In case of custom role where role prefix does not start with “ROLE\_”, @Secured annotation will not work.**

# **Java Source Code**

**BankTransaction.java**

**package** com.ddlab.rnd.spring.security;

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

**public** **interface** BankTransaction {

/\*\*

\* Can be done by Manager and Admin

\*/

@PreAuthorize ("hasRole('APP\_MANAGER')")

**public** **void** approveLoan(String actNo);

/\*\*

\* Can be done only by USER

\*/

// @Secured("USER") ~~ @Secured annotation does not work with Custom Role and Role Prefix

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

**public** **void** openAccount(String customerName);

/\*\*

\* Can be done by Manager and Admin

\*/

@PreAuthorize ("hasRole('APP\_ADMIN')")

**public** **void** resetNetBankingPassword(String customerName);

/\*\*

\* Can be done by Teller, Manager and Admin

\*/

@PreAuthorize ("hasRole('APP\_TELLER')")

**public** **void** checkAccountStatus(String actNo);

/\*\*

\* Can be done by anybody in bank other than Normal user or customer

\*/

@PreAuthorize("hasAnyRole('APP\_ADMIN', 'APP\_MANAGER' , 'APP\_TELLER')")

**public** **void** suggestCreditCard(String customerName);

/\*\*

\* Can be approved only by Manager and not by any other persons

\*/

@PreAuthorize("hasAuthority('APP\_MANAGER')")

**public** **void** approveOverDraftFacility(String customerName);

//Only GOVT can close this Bank, No one can do this

@PreAuthorize("hasRole('APP\_GOVT')")

**public** **void** shutDownBank();

}

**BankTransactionImpl.java**

**package** com.ddlab.rnd.spring.security;

**public** **class** BankTransactionImpl **implements** BankTransaction {

**public** **void** approveLoan(String actNo) {

System.***out***.println("The loan has been approved for the account no : "+actNo);

}

**public** **void** openAccount(String customerName) {

System.***out***.println("Hi "+customerName+" , your account will be activated within 24 hours");

}

**public** **void** resetNetBankingPassword(String customerName) {

System.***out***.println("Hi "+customerName+", Administrator has reset your internet banking password "

+ "and an email has been sent you");

}

**public** **void** checkAccountStatus(String actNo) {

System.***out***.println("You available balance is 2000 INR");

}

**public** **void** suggestCreditCard(String customerName) {

System.***out***.println("Hi "+customerName+" , a life time free Titanium credit card is available for you");

}

**public** **void** approveOverDraftFacility(String customerName) {

System.***out***.println("Hi "+customerName+" , a overdraft facility is approved for you");

}

**public** **void** shutDownBank() {

System.***out***.println("Bank has been shutdown by Government");

}

}

**RoleType.java**

**package** com.ddlab.rnd.spring.security;

**public** **enum** ROLETYPE {

***GOVT***("APP\_GOVT") , ***USER***("APP\_USER") , ***ADMIN***("APP\_ADMIN") , ***MANAGER***("APP\_MANAGER") , ***TELLER***("APP\_TELLER");

**private** String role;

**private** ROLETYPE(String role) {

**this**.role = role;

}

@Override

**public** String toString() {

**return** role;

}

}

**SecurityUtil.java**

**package** com.ddlab.rnd.spring.security;

**import** java.util.List;

**import** org.springframework.security.authentication.UsernamePasswordAuthenticationToken;

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

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

**import** org.springframework.security.core.context.SecurityContextHolder;

**import** org.springframework.security.core.userdetails.User;

**public** **class** SecurityUtil {

**public** **static** **void** loginAs(String userName, String password, ROLETYPE roleType) {

*setSecurity*(userName, password, roleType.toString());

}

**public** **static** **void** logout() {

// SecurityContextHolder.getContext().setAuthentication(null);

// SecurityContextHolder.getContext().getAuthentication().setAuthenticated(false);

SecurityContextHolder.*clearContext*();

}

**private** **static** **void** setSecurity(String userName, String password,String roleType) {

List<SimpleGrantedAuthority> auths = **new** java.util.ArrayList<SimpleGrantedAuthority>();

auths.add(**new** SimpleGrantedAuthority(roleType));

User user = **new** User(userName, password, auths);

Authentication authToken = **new** UsernamePasswordAuthenticationToken (user.getUsername(), user.getPassword(), user.getAuthorities());

SecurityContextHolder.*getContext*().setAuthentication(authToken);

}

}

**Test.java**

**package** com.ddlab.rnd.spring.security;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public** **class** Test {

//Person having designation more than MANAGER

**public** **static** **void** approveLoan(BankTransaction bankTxn) {

SecurityUtil.*loginAs*("Deb", "pqrs", ROLETYPE.***MANAGER***);

// SecurityUtil.loginAs("Deb", "pqrs", ROLETYPE.ADMIN);

bankTxn.approveLoan("12345");

}

**public** **static** **void** openAccount(BankTransaction bankTxn) {

// SecurityUtil.loginAs("Deb", "pqrs", ROLETYPE.USER);

SecurityUtil.*loginAs*("Deb", "pqrs", ROLETYPE.***USER***);

bankTxn.openAccount("Deb");

}

**public** **static** **void** restPassword(BankTransaction bankTxn) {

SecurityUtil.*loginAs*("Deb", "pqrs", ROLETYPE.***ADMIN***);

bankTxn.resetNetBankingPassword("customerName");

}

**public** **static** **void** checkAccount(BankTransaction bankTxn) {

SecurityUtil.*loginAs*("Deb", "pqrs", ROLETYPE.***TELLER***);

bankTxn.checkAccountStatus("111");

}

**public** **static** **void** suggestCard(BankTransaction bankTxn) {

SecurityUtil.*loginAs*("Deb", "pqrs", ROLETYPE.***MANAGER***);

bankTxn.suggestCreditCard("Deb");

}

**public** **static** **void** approveOverDraftFacility(BankTransaction bankTxn) {

SecurityUtil.*loginAs*("Deb", "pqrs", ROLETYPE.***MANAGER***);

bankTxn.approveOverDraftFacility("Deb");

}

**public** **static** **void** closeBank(BankTransaction bankTxn) {

SecurityUtil.*loginAs*("Deb", "pqrs", ROLETYPE.***GOVT***);

bankTxn.shutDownBank();

}

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

ApplicationContext applicationContext = **new** ClassPathXmlApplicationContext("beans.xml");

BankTransaction bankTxn = (BankTransaction) applicationContext.getBean("bankTxn");

*approveLoan*(bankTxn);

*openAccount*(bankTxn);

*restPassword*(bankTxn);

*checkAccount*(bankTxn);

*suggestCard*(bankTxn);

*approveOverDraftFacility*(bankTxn);

//Only Govt can close the Bank

*closeBank*(bankTxn);

}

}

Custom Role with Prefix “ROLE\_” in Spring Security secured @Secured annotation

In this case, the example is similar which is given above, the following changes have been mentioned.

# **Spring Security Configuration(beans.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:aop=*"http://www.springframework.org/schema/aop"*

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

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

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

xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.1.xsd*

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

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

<bean id=*"roleHierarchy"* class=*"org.springframework.security.access.hierarchicalroles.RoleHierarchyImpl"*>

<property name=*"hierarchy"*>

<value>

ROLE\_ADMIN > ROLE\_MANAGER

ROLE\_MANAGER > ROLE\_TELLER

ROLE\_TELLER > ROLE\_USER

ROLE\_GOVT

</value>

</property>

</bean>

<bean id=*"defaultMethodSecurityExpressionHandler"*

class=*"org.springframework.security.access.expression.method.DefaultMethodSecurityExpressionHandler"*>

<property name=*"roleHierarchy"* ref=*"roleHierarchy"* />

</bean>

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

<security:expression-handler ref=*"defaultMethodSecurityExpressionHandler"* />

</security:global-method-security>

<bean id=*"bankTxn"* class=*"com.ddlab.rnd.spring.security.BankTransactionImpl"* />

</beans>

# **Java Source Code(BankTransaction.java)**

**package** com.ddlab.rnd.spring.security;

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

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

**public** **interface** BankTransaction {

/\*\*

\* Can be done by Manager and Admin

\*/

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

**public** **void** approveLoan(String actNo);

/\*\*

\* Can be done only by USER

\*/

// @Secured("USER") ~~ @Secured annotation does not work with Custom Role and Role Prefix

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

**public** **void** openAccount(String customerName);

/\*\*

\* Can be done by Manager and Admin

\*/

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

**public** **void** resetNetBankingPassword(String customerName);

/\*\*

\* Can be done by Teller, Manager and Admin

\*/

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

**public** **void** checkAccountStatus(String actNo);

/\*\*

\* Can be done by anybody in bank other than Normal user or customer

\*/

@Secured({"ROLE\_ADMIN", "ROLE\_MANAGER" , "ROLE\_TELLER"})

**public** **void** suggestCreditCard(String customerName);

/\*\*

\* Can be approved only by Manager and not by any other persons

\*/

@PreAuthorize("hasAuthority('ROLE\_MANAGER')")

**public** **void** approveOverDraftFacility(String customerName);

//Only GOVT can close this Bank, No one can do this

// @PreAuthorize("hasRole('ROLE\_GOVT')")

@Secured("ROLE\_GOVT")

**public** **void** shutDownBank();

}