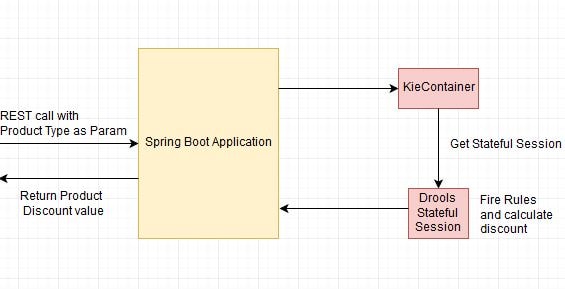
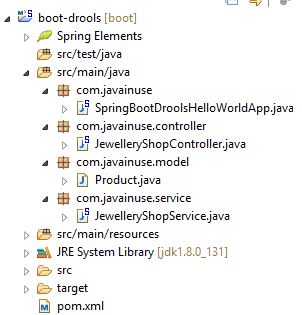
# Drools Drools Stateful Session + Integration with Spring Boot

# Overview

In this tutorial we will create a Spring Boot Application and integrate with JBoss Drools. Like previous drools examples we will create an application for a jewellery shop to calculate discount based on the type of jewellery. We have [previously integrated Drools with Spring MVC.](https://www.javainuse.com/drools/drools_spring) We will be using Stateful Drools session. We will be exposing a REST API which returns the discount based on the type of product value passed as request parameter.



The project structure is as follows-  


The pom.xml will be as follows-

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

<modelVersion>4.0.0</modelVersion>

<groupId>com.javainuse</groupId>

<artifactId>boot-drools</artifactId>

<version>1.0.0-SNAPSHOT</version>

<packaging>jar</packaging>

<name>boot-drools</name>

<description>Demo project for Spring Boot Drools</description>

<properties>

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

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

<kie.version>6.1.0.Final</kie.version>

</properties>

<dependencyManagement>

<dependencies>

<dependency>

<groupId>io.spring.platform</groupId>

<artifactId>platform-bom</artifactId>

<version>1.1.1.RELEASE</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

**<dependency>**

**<groupId>org.kie</groupId>**

**<artifactId>kie-ci</artifactId>**

**<version></version>**

**</dependency>**

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

Define the model class Product.java as follows-

package com.javainuse.model;

public class Product {

private String type;

private int discount;

public String getType() {

return type;

}

public void setType(String type) {

this.type = type;

}

public int getDiscount() {

return discount;

}

public void setDiscount(int discount) {

this.discount = discount;

}

}

Now define the rules in the drl file. We will use the type property of the Product class for defining the rules for defining what action needs to be taken if a particular condition is met. The .drl file should be place in resources/com.rule folder.

package rules

import com.javainuse.model.Product

rule "Offer for Diamond"

when

productObject: Product(type=="diamond")

then

productObject.setDiscount(15);

end

rule "Offer for Gold"

when

productObject: Product(type=="gold")

then

productObject.setDiscount(25);

end

The Drools 6.0 project consists of a meta data file META-INF/kmodule.xml. The file is located under the source folder as shown in below snapshot. A named session ksession-rule is created. The ksession rule is applicable for all drl file contained in the pacakge rules. Currently we have only one drl file of package rules.

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

<kmodule xmlns="http://jboss.org/kie/6.0.0/kmodule">

<kbase name="rules" packages="rules">

<ksession name="rulesSession"/>

</kbase>

</kmodule>

Define the JewelleryShopController as follows to expose the GET REST API.

package com.javainuse.controller;

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

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

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

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import com.javainuse.model.Product;

import com.javainuse.service.JewelleryShopService;

@RestController

public class JewelleryShopController {

private final JewelleryShopService jewelleryShopService;

@Autowired

public JewelleryShopController(JewelleryShopService jewelleryShopService) {

this.jewelleryShopService = jewelleryShopService;

}

**@RequestMapping(value = "/getDiscount", method = RequestMethod.GET, produces = "application/json")**

**public Product getQuestions(@RequestParam(required = true) String type)** {

Product product = new Product();

product.setType(type);

jewelleryShopService.getProductDiscount(product);

return product;

}

}

Define the JewelleryShopService as follows to get the Stateful session and execute the drools rules.

package com.javainuse.service;

import org.kie.api.runtime.KieContainer;

import org.kie.api.runtime.KieSession;

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

import org.springframework.stereotype.Service;

import com.javainuse.model.Product;

@Service

public class JewelleryShopService {

private final KieContainer kieContainer;

@Autowired

public JewelleryShopService(KieContainer kieContainer) {

this.kieContainer = kieContainer;

}

public Product getProductDiscount(Product product) {

**//get the stateful session**

KieSession kieSession = kieContainer.newKieSession("rulesSession");

kieSession.insert(product);

kieSession.fireAllRules();

kieSession.dispose();

return product;

}

}

Finally create the Main class with SpringBootApplication annotation. Also create a bean of type of KieContainer using JavaConfig

package com.javainuse;

import org.kie.api.KieServices;

import org.kie.api.runtime.KieContainer;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.Bean;

@SpringBootApplication

public class SpringBootDroolsHelloWorldApp {

public static void main(String[] args) {

SpringApplication.run(SpringBootDroolsHelloWorldApp.class, args);

}

**@Bean**

**public KieContainer kieContainer() {**

**return KieServices.Factory.get().getKieClasspathContainer();**

**}**

}

We are done with the code changes, now start the application. Open a browser and hit the url- http://localhost:8080/getDiscount?type=gold. We get the output with the discount value as follows-  
