

Search Tutorials

# Drools Tutorials- Drools Stateful Session + Integration with Spring Boot

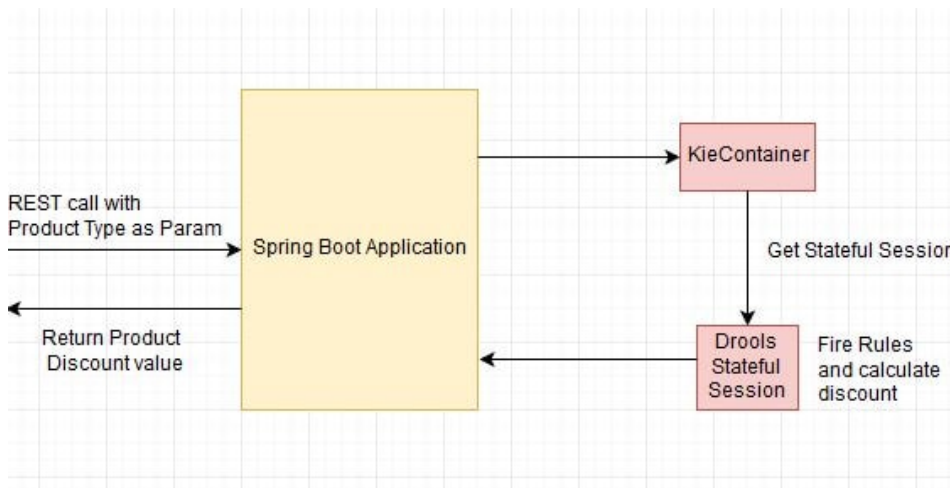


Introducing high-velocity I  
Powered by Jira Service Manage

## Overview

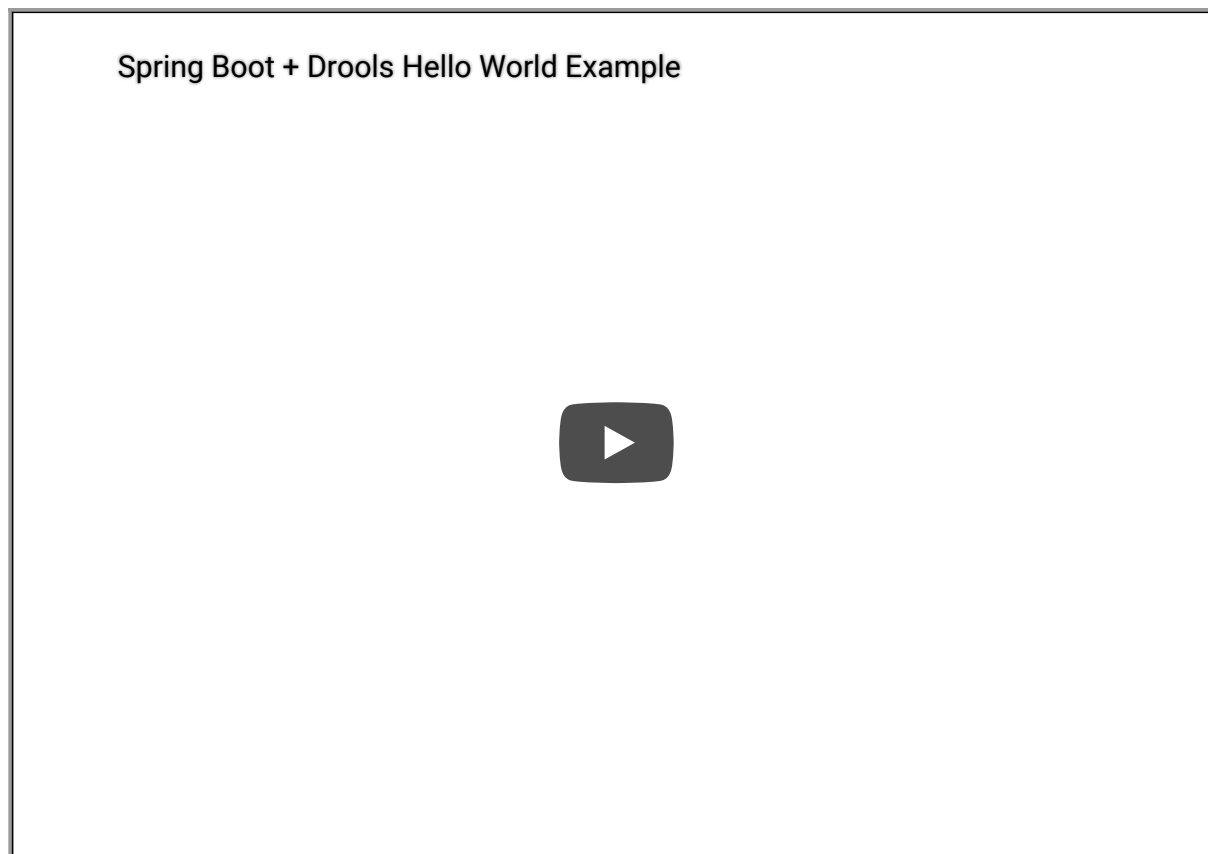
This site uses cookies to deliver our services and to show you relevant ads. By continuing to visit this site you agree to our use of cookies. Learn more (<http://www.javainuse.com/privacy>)

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. (/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.



## Video

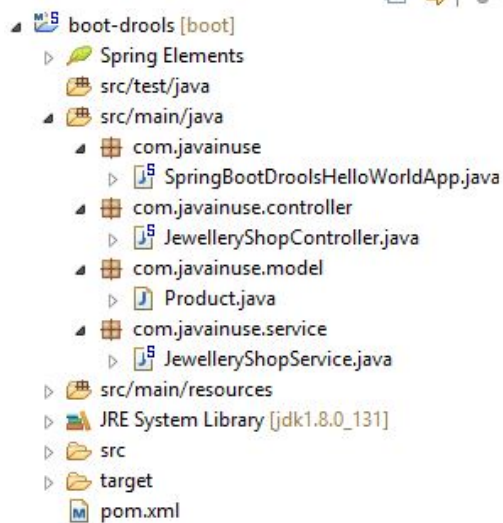
This tutorial is explained in the below Youtube Video.



This site uses cookies to deliver our services and to show you relevant ads. By continuing to visit this site you agree to our use of cookies. Learn more (<http://www.javainuse.com/privacy>)

# Lets Begin

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
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.a

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

```


```
</dependencies>
<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>

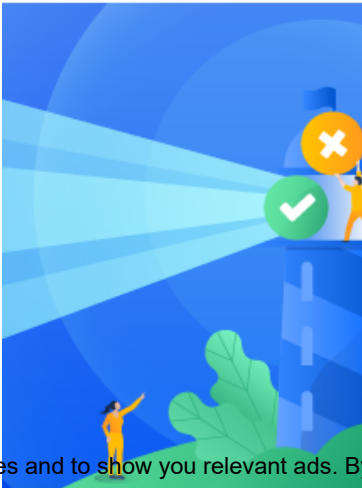
</project>
```



Share incident status in real time

Try Statuspage free





This site uses cookies to deliver our services and to show you relevant ads. By continuing to visit this site you agree to our use of cookies. Learn more (<http://www.javainuse.com/privacy>)

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.

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.

This site uses cookies to deliver our services and to show you relevant ads. By continuing to visit this site you agree to our use of cookies. Learn more (<http://www.javainuse.com/privacy>)

The `package rules` is applicable for all drl file contained in the package rules. Currently we have only one drl file of package rules.

```
import com.javainuse.model.Product
```

```
<?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>
```

```
end
```

```
rule "Offer for Gold"
```

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

```
when
    productObject: Product(type=="gold")
```

```
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,
    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("rulesSe
        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 {

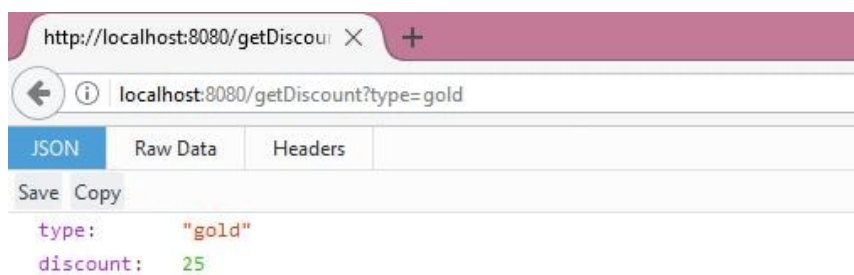
    public static void main(String[] args) {
        SpringApplication.run(SpringBootDroolsHelloWorldApp.class, args);
    }

    @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-



## Download Source Code

This site uses cookies to deliver our services and to show you relevant ads. By continuing to visit this site you agree to our use of cookies. [Learn more](#) [http://www.javainuse.com/spring/spring-boot-integration-example/zip/spring-boot-drools.rar](#)

## Popular Posts

- [Spring Boot Interview Questions \(/spring/SpringBootInterviewQuestions\)](/spring/SpringBootInterviewQuestions)
- [Implement Spring Boot Security and understand Spring Security Architecture \(/webseries/spring-security-jwt/chap3\)](/webseries/spring-security-jwt/chap3)
- [E-commerce Website - Online Book Store using Angular 8 + Spring Boot \(/fullstack/ecommerce\)](/fullstack/ecommerce)
- [Spring Boot + JSON Web Token\(JWT\) Hello World Example \(/spring/boot-jwt\)](/spring/boot-jwt)
- [Angular 7 + Spring Boot Application Hello World Example \(/spring/ang7-hello\)](/spring/ang7-hello)
- [Build a Real Time Chat Application using Spring Boot + WebSocket + RabbitMQ \(/spring/boot-websocket-chat\)](/spring/boot-websocket-chat)
- [Pivotal Cloud Foundry Tutorial - Deploy Spring Boot Application Hello World Example \(/pcf/pcf-hello\)](/pcf/pcf-hello)
- [Deploying Spring Based WAR Application to Docker \(/devOps/docker/docker-war\)](/devOps/docker/docker-war)
- [EIP patterns using Apache Camel \(/camel/camel\\_EIP\)](/camel/camel_EIP)
- [Spring Cloud- Netflix Eureka + Ribbon Simple Example \(/spring/spring\\_ribbon\)](/spring/spring_ribbon)
- [Spring Cloud- Netflix Hystrix Circuit Breaker Simple Example \(/spring/spring\\_hystrix\\_circuitbreaker\)](/spring/spring_hystrix_circuitbreaker)
- [Spring Boot + Swagger Example Hello World Example \(/spring/boot\\_swagger\)](/spring/boot_swagger)

This site uses cookies to deliver our services and to show you relevant ads. By continuing to visit this site you agree to our use of cookies. Learn more (<http://www.javainuse.com/privacy>)

- [Spring Boot Batch Simple example \(/spring/bootbatch\)](/spring/bootbatch)

- [Spring Boot + Apache Kafka Example \(/spring/spring-boot-apache-kafka-hello-world\)](#)
- [Spring Boot Admin Simple Example \(/spring/boot-admin\)](#)
- [Spring Boot Security - Introduction to OAuth \(/spring/spring-boot-oauth-introduction\)](#)
- [Spring Boot OAuth2 Part 1 - Getting The Authorization Code \(/spring/spring-boot-oauth-authorization-code\)](#)
- [Spring Boot OAuth2 Part 2 - Getting The Access Token And Using it to Fetch Data. \(/spring/spring-boot-oauth-access-token\)](#)
- [JBoss Drools Hello World-Stateful Knowledge Session using KieSession \(/drools\\_hello\\_kie\)](#)
- [Understand Drools Stateful vs Stateless Knowledge Session \(/drools\\_states\)](#)
- [JBoss Drools- Understanding Drools Decision Table using Simple Example \(/drools/drools\\_decision\)](#)

## See Also

- [Spring Batch Interview Questions \(/spring/sprbatch\\_interview\)](#)
- [Spring AOP Interview Questions \(/spring/spring-AOP-interview-questions\)](#)
- [Angular 2 Interview Questions \(/angular/ang2\\_intvw\)](#)
- [Apache Camel Interview Questions \(/camel/Apache\\_Camel\\_Questions\)](#)
- [JBoss Fuse Interview Questions \(/camel/JBoss\\_Fuse\\_Questions\)](#)
- [Drools Interview Questions \(/drools/drools\\_intvw\)](#)
- [Java 8 Interview Questions \(/java/java8\\_intvw\)](#)

This site uses cookies to deliver our services and to show you relevant ads. By continuing to visit this site you agree to our use of cookies. Learn more (<http://www.javainuse.com/privacy>)

- [Spring Cloud Interview Questions \(/spring/spring-cloud-interview-questions\)](#)
- [Microservices Interview Questions \(/spring/microservices-interview-questions\)](#)
- [Java HashMap and ConcurrentHashMap Interview Questions \(/java/java\\_map\\_intvw\)](#)
- [Snowflake frequently asked interview questions \(/prep/snowflake\)](#)
- [SAP FI - Accounts Receivable frequently asked interview questions \(/prep/sap1\)](#)
- [Top SAP ALV Interview Questions \(/prep/sap3\)](#)
- [Top SAP Business Objects Administration Interview Questions \(/prep/sap2\)](#)
- [EC2 frequently asked interview questions \(/prep/ec2\)](#)
- [Mule ESB frequently asked interview questions \(/misc/muleintvw\)](#)
- [Apache Kafka Interview Questions \(/misc/apache-kafka-interview-questions\)](#)
- [Tosca Testing Tool Interview Questions \(/misc/tosca-testing-tool-interview-questions\)](#)
- [Top Maven Build Tool Interview Questions \(/misc/maven-interview-questions\)](#)
- [Top Gradle Build Tool Interview Questions \(/misc/gradle-interview-questions\)](#)
- [Miscellaneous Topics \(/misc\)](#)



**Build software  
smarter**

Ad PieceX

**JBoss Drools Hello  
World-Stateful  
Knowledge Session  
using KieSession**

[javainuse.com](http://javainuse.com)

**usb java fingerprint  
scanner**

Ad

**Drools Spring  
Integration Example**

[javainuse.com](http://javainuse.com)

**Laraship Elite - #1  
Laravel Modular  
Platform**

Ad [laraship.com](http://laraship.com)

**JBoss Drools Stateful  
vs Stateless  
Knowledge Session**

[javainuse.com](http://javainuse.com)

**Getting started with  
Drools Decision Table  
using Drools 6 Simple  
example**

[javainuse.com](http://javainuse.com)

**Drools Tutori**

[javainuse.com](http://javainuse.com)



