# Drools Hello World-Stateful Knowledge Session using KieSession

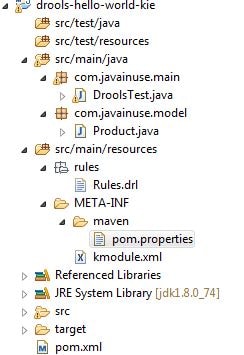
Overview

In previous chapter we implemented a simple drools project to execute simple rule. From Drools 6.0 onwards a new approach is used to create a Knowledge Base and a Knowledge Session. Knowledge base is an interface that manages a set of rules and processes. The main task of the knowledge base is to store and re-use rules because creation of rules is very expensive.Rules are contained inside the package org.drools.KnowledgeBase. These are commonly referred to as knowledge definitions or knowledge. Knowledge base provides methods for creating a Session.  
The knowledge session can be of two types:

* Stateless Knowledge Session
* Stateful Knowledge Session

In this chapter we make use of KieSession to get stateful session. Stateful sessions are longer lived and allow iterative changes over time.

Lets Begin

Using drools we have to define the discounts offered on various jewellery products depending on the type. For example if the product is jewellery item offer a discount of 25% and so on. We will create Eclipse Maven project as follows-  


The POM defined is as follows- Only a single dependency og Drools-compiler is required.

<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>com.javainuse</groupId>

<artifactId>drools-hello-world</artifactId>

<version>0.0.1-SNAPSHOT</version>

<properties>

<drools.version>6.2.0.Final</drools.version>

</properties>

<dependencies>

<dependency>

<groupId>org.kie</groupId>

<artifactId>kie-api</artifactId>

<version></version>

</dependency>

<dependency>

<groupId>org.drools</groupId>

<artifactId>drools-core</artifactId>

<version></version>

</dependency>

<dependency>

<groupId>org.drools</groupId>

<artifactId>drools-compiler</artifactId>

<version></version>

</dependency>

<dependency>

<groupId>org.kie</groupId>

<artifactId>kie-ci</artifactId>

<version></version>

</dependency>

</dependencies>

</project>

Next define the Model class Product which defines the type of jewellery item.

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.

ackage 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);

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="ksession-rule"/>

</kbase>

</kmodule>

Also drools kie-api/internal library execution looks for mandatory values under src\main\resources\META-INF\maven\pom.properties file for the drools eclipse project. If this file is not specified then the nullpointer exception is thrown.

groupId=com.javainuse

artifactId=drools-hello-world

version=0.0.1-SNAPSHOT

Finally we define DroolsTest class. Here load the facts and the rules in the drools working memory and firing all the rules.

package com.javainuse.main;

import org.kie.api.KieServices;

import org.kie.api.runtime.KieContainer;

import org.kie.api.runtime.KieSession;

import org.kie.api.runtime.rule.FactHandle;

import com.javainuse.model.Product;

public class DroolsTest {

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

try {

KieServices ks = KieServices.Factory.get();

KieContainer kContainer = ks.getKieClasspathContainer();

**//Get the session named kseesion-rule that we defined in kmodule.xml above.**

**//Also by default the session returned is always stateful.**

KieSession kSession = kContainer.newKieSession("ksession-rule");

Product product = new Product();

product.setType("gold");

FactHandle fact1;

fact1 = kSession.insert(product);

kSession.fireAllRules();

System.out.println("The discount for the jewellery product "

+ product.getType() + " is " + product.getDiscount());

} catch (Throwable t) {

t.printStackTrace();

}

}

}

On running get the output as-  
