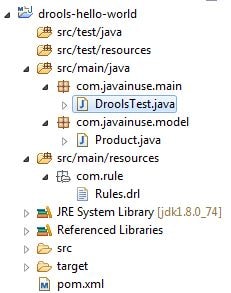
# Drools Hello World

Overview

Consider a Jewellery shop which needs to maintain business logic of calculating discount based on the type of Jewellery. This can be done in our Java Code. But such rules may change daily and need to be updated regularly in our code which is not good. Also we will always require a Developer for making such changes. So its a good practice to define this logic as rules in a Business Rule Management System. If tomorrow the discount is to be changed this can be done even by a non technical person.

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>

<dependencies>

**<dependency>**

**<groupId>org.drools</groupId>**

**<artifactId>drools-compiler</artifactId>**

**<version>6.0.1.Final</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.

package com.rule

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

Drools employs a concept called Working Memory. Finally we define DroolsTest class to load the facts and the rules in the drools working memory and firing all the rules.

package com.javainuse.main;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.Reader;

import org.drools.compiler.compiler.DroolsParserException;

import org.drools.compiler.compiler.PackageBuilder;

import org.drools.core.RuleBase;

import org.drools.core.RuleBaseFactory;

import org.drools.core.WorkingMemory;

import com.javainuse.model.Product;

public class DroolsTest {

public static void main(String[] args) throws DroolsParserException,

IOException {

DroolsTest droolsTest = new DroolsTest();

droolsTest.executeDrools();

}

public void executeDrools() throws DroolsParserException, IOException {

PackageBuilder packageBuilder = new PackageBuilder();

String ruleFile = "/com/rule/Rules.drl";

InputStream resourceAsStream = getClass().getResourceAsStream(ruleFile);

Reader reader = new InputStreamReader(resourceAsStream);

packageBuilder.addPackageFromDrl(reader);

org.drools.core.rule.Package rulesPackage = packageBuilder.getPackage();

RuleBase ruleBase = RuleBaseFactory.newRuleBase();

ruleBase.addPackage(rulesPackage);

WorkingMemory workingMemory = ruleBase.newStatefulSession();

Product product = new Product();

product.setType("gold");

workingMemory.insert(product);

workingMemory.fireAllRules();

System.out.println("The discount for the product " + product.getType()

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

}

}

On running get the output as-  
