**Lab Exercise 1- Create Simple Drool Project**

Creating a simple Drools project as a lab exercise can help you understand the basic concepts of business rules, rule engines, and how they interact with Java applications. Below is a step-by-step guide to creating a simple Drools project using Maven, which is a common build tool for Java projects.

**Objective:**

* To create a basic Drools project that evaluates a simple business rule. The rule will check if a customer is eligible for a discount based on their membership status.

**Prerequisites:**

* Basic understanding of Java programming.
* Maven installed on your machine.
* An Integrated Development Environment (IDE) like IntelliJ IDEA, Eclipse, or VS Code.

**Step 1: Set Up Maven Project**

* Create a New Maven Project:
* Open your IDE.

Create a new Maven project with the following structure:

- src

- main

- java

- resources

- test

- java

**Update the pom.xml File:**

Add the necessary Drools dependencies to your pom.xml file:

<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.example</groupId>

<artifactId>drools-example</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<!-- Drools dependencies -->

<dependency>

<groupId>org.kie</groupId>

<artifactId>kie-api</artifactId>

<version>7.74.0.Final</version>

</dependency>

<dependency>

<groupId>org.kie</groupId>

<artifactId>kie-dmn-api</artifactId>

<version>7.74.0.Final</version>

</dependency>

<dependency>

<groupId>org.drools</groupId>

<artifactId>drools-core</artifactId>

<version>7.74.0.Final</version>

</dependency>

<dependency>

<groupId>org.drools</groupId>

<artifactId>drools-compiler</artifactId>

<version>7.74.0.Final</version>

</dependency>

<dependency>

<groupId>org.drools</groupId>

<artifactId>drools-decisiontables</artifactId>

<version>7.74.0.Final</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-simple</artifactId>

<version>1.7.30</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

**Step 2: Define a Java Model**

**Create a Customer Class:**

In the src/main/java/com/example/model directory, create a Customer.java file with the following content:

package com.example.model;

public class Customer {

private String name;

private String membership;

private int discount;

public Customer(String name, String membership) {

this.name = name;

this.membership = membership;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getMembership() {

return membership;

}

public void setMembership(String membership) {

this.membership = membership;

}

public int getDiscount() {

return discount;

}

public void setDiscount(int discount) {

this.discount = discount;

}

@Override

public String toString() {

return "Customer{name='" + name + "', membership='" + membership + "', discount=" + discount + '}';

}

}

**Step 3: Create a DRL File**

**Create the Rule File:**

In the src/main/resources directory, create a file named **discount.drl** with the following content:

package com.example.rules

rule "Apply Discount to Premium Members"

when

$customer : com.example.model.Customer(membership == "Premium")

then

$customer.setDiscount(10);

System.out.println("Applied 10% discount to premium customer: " + $customer.getName());

end

**Step 4: Write a Main Application**

**Create a Main Class:**

In the src/main/java/com/example directory, create a DroolsTest.java file with the following content:

package com.example;

import com.example.model.Customer;

org.kie.api.KieServices;

org.kie.api.runtime.KieContainer;

org.kie.api.runtime.KieSession;

public class DroolsTest {

public static void main(String[] args) {

// Load the knowledge base

KieServices ks = KieServices.Factory.get();

KieContainer kContainer = ks.getKieClasspathContainer();

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

// Create a new customer

Customer customer = new Customer("John Doe", "Premium");

// Insert the customer into the session

kSession.insert(customer);

// Fire all rules

kSession.fireAllRules();

// Print the customer details

System.out.println(customer);

// Dispose the session

kSession.dispose();

}

}

**Configure kmodule.xml:**

In the src/main/resources directory, create a META-INF/kmodule.xml file with the following content:

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

<kmodule xmlns="http://www.drools.org/xsd/kmodule">

<kbase name="kbase" packages="com.example.rules">

<ksession name="ksession-rules"/>

</kbase>

</kmodule>

**Step 5: Run the Application**

**Compile and Run:**

Use your IDE to compile and run the DroolsTest class. Alternatively, you can use

Maven to build and run the project:

mvn clean install

mvn exec:java -Dexec.mainClass="com.example.DroolsTest"

**Expected Output:**

The console should display the following output:

Applied 10% discount to premium customer: John Doe

Customer{name='John Doe', membership='Premium', discount=10}

This simple exercise demonstrates how to set up a basic Drools project, define business rules, and execute them in a Java application. You can expand on this example by adding more rules, integrating decision tables, or creating more complex business logic.