**Agenda-Group in Drools**

**Overview:**

The **agenda-group** attribute in Drools is used to control the order in which rules are executed by grouping them into different sets called "agenda groups." Each agenda group can be given focus explicitly, meaning only the rules within that group will be considered for execution at a given time. This feature is particularly useful in complex rule-based systems where you need to sequence the execution of rules according to a specific workflow or priority.

**How Agenda Groups Work**

* **Defining an Agenda Group:**
  + You assign a rule to an agenda group by using the agenda-group attribute in the rule definition.

rule "MyRule"

agenda-group "my-group"

when

// conditions

then

// actions

end

* **Activating an Agenda Group:**
  + The execution of rules in a specific agenda group is controlled by setting the focus to that group. You do this using the KieSession.getAgenda().getAgendaGroup("group\_name").setFocus(); method.

ksession.getAgenda().getAgendaGroup("my-group").setFocus();

ksession.fireAllRules();

* **Default Agenda Group:**
  + If no agenda-group is specified, a rule belongs to the default agenda group. Rules in the default group will be executed if no other group has focus.

**Example Use Case**

Consider a scenario where an e-commerce order processing system is divided into different stages: validation, processing, and notification. Each stage can be represented as an agenda group.

**1. Example Rules File (orderRules.drl)**

package com.example.rules;

import com.example.model.Order;

import com.example.model.Payment;

import com.example.model.Customer;

rule "Validate Order Amount"

agenda-group "validation"

when

$order : Order( totalAmount > 0 )

then

$order.setValid(true);

System.out.println("Order " + $order.getId() + " is valid.");

update($order);

end

rule "Process Payment"

agenda-group "processing"

when

$order : Order( isValid == true )

$payment : Payment( orderId == $order.getId(), isPaid == false )

then

$payment.setPaid(true);

$order.setProcessed(true);

System.out.println("Payment processed for order " + $order.getId());

update($payment);

update($order);

end

rule "Notify Customer"

agenda-group "notification"

when

$order : Order( isProcessed == true )

$customer : Customer( isNotified == false )

then

$customer.setNotified(true);

System.out.println("Customer " + $customer.getName() + " notified for order " + $order.getId());

update($customer);

end

**2. Setting Agenda Group Focus in the Application**

package com.example;

import com.example.model.Order;

import com.example.model.Payment;

import com.example.model.Customer;

import org.kie.api.KieServices;

import org.kie.api.runtime.KieContainer;

import org.kie.api.runtime.KieSession;

public class OrderProcessingApp {

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 facts

Order order = new Order("Order-001", 150.0);

Payment payment = new Payment(order.getId());

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

// Insert facts into the session

kSession.insert(order);

kSession.insert(payment);

kSession.insert(customer);

// Focus on the "validation" agenda group and fire rules

kSession.getAgenda().getAgendaGroup("validation").setFocus();

kSession.fireAllRules();

// Focus on the "processing" agenda group and fire rules

kSession.getAgenda().getAgendaGroup("processing").setFocus();

kSession.fireAllRules();

// Focus on the "notification" agenda group and fire rules

kSession.getAgenda().getAgendaGroup("notification").setFocus();

kSession.fireAllRules();

// Dispose of the session

kSession.dispose();

}

}

**When to Use Agenda Groups**

* **Controlled Execution Flow:** Agenda groups allow you to control the sequence in which rules are fired, which is particularly useful in multi-stage processes.
* **Prioritizing Rule Execution:** By setting focus on different agenda groups, you can prioritize certain operations over others within your rule base.
* **Complex Workflows:** When dealing with complex business processes, where different steps must be executed in a specific order, agenda groups provide a clear and manageable way to enforce this sequence.

**Important Notes**

* **Salience and Agenda Groups:** You can combine agenda-group with the salience attribute to further control the order of rule execution within a group.
* **Agenda Group Activation:** Only one agenda group can be active (in focus) at any time. Once a group is exhausted, you can switch focus to another group.
* **Default Group Behavior:** If no agenda group is explicitly given focus, rules in the default agenda group may fire.

**Conclusion**

The agenda-group attribute in Drools is a powerful feature for managing the flow of rule execution in complex scenarios. It allows you to organize rules into groups and control their execution sequence, ensuring that your business logic is applied in a structured and predictable manner.