**KieModel, KieContainer and KieBase**

In Drools, KieModel and KieContainer are part of the Knowledge Is Everything (KIE) API, which is used to manage and execute Drools rules. These concepts are crucial for setting up the environment in which rules are executed. Let's explore each one and understand their roles, particularly focusing on KieBases.

**1. KieModel**

* **Definition:**
  + The KieModel represents the knowledge model in Drools. It is an internal representation of all the KIE modules and their resources (rules, processes, etc.) that can be deployed in a KIE project. It is generated during the build process by Drools and is typically not interacted with directly by developers.
* **Purpose:**
  + The KieModel provides a structured way to organize and manage various resources (like rules) in your Drools project. It is mainly used by the KIE runtime to locate and manage KieBases and KieSessions.
* **How it works:**
  + During the build process of a Drools project, the KieModel is generated by processing the kmodule.xml file and other resources. This model helps the KieContainer to load and create KieBases and KieSessions.

**2. KieContainer**

* **Definition:**
  + The KieContainer is a runtime representation of a KIE project. It is responsible for loading the KieModel and creating KieBases and KieSessions based on the configuration specified in the project (typically defined in kmodule.xml).
* **Purpose:**
  + The KieContainer is the main entry point for working with Drools rules at runtime. It manages the lifecycle of KieBases and KieSessions, ensuring that the correct resources are loaded and used during rule execution.
* **How it works:**
  + When the KieContainer is instantiated (usually via KieServices), it loads the KieModel, which includes all the rule resources and configurations. From the KieContainer, you can retrieve KieBases and KieSessions to execute rules.

**3. KieBase**

* **Definition:**
  + The KieBase (Knowledge Base) is a repository of all the knowledge definitions (rules, processes, functions, etc.) that Drools uses to perform reasoning. It is essentially a compiled representation of your rules that can be executed in a session.
* **Purpose:**
  + KieBases provide a mechanism to separate the definition of rules from their execution. You can think of KieBase as a container for all the compiled rules that are ready to be executed. Multiple KieSessions can be created from a single KieBase.
* **How it works:**
  + A KieBase is created within a KieContainer. The KieBase is initialized with all the rules and processes defined in the resources. Once a KieBase is created, it can be used to create KieSessions, which are then used to insert facts and fire rules.
* **Relationship with KieContainer:**
  + A KieContainer can contain multiple KieBases. Each KieBase can be configured differently, allowing you to have distinct sets of rules for different purposes within the same project.

**Example Workflow**

Here’s how these components typically interact in a Drools application:

1. **Setup and Build:**
   * During the build process, the KieModel is generated from the kmodule.xml and other resources. This model organizes all the rules, processes, and configurations.
2. **Load the KieContainer:**
   * At runtime, you create a KieContainer using KieServices. This container loads the KieModel and makes it possible to retrieve KieBases and KieSessions.

KieServices ks = KieServices.Factory.get();

KieContainer kContainer = ks.getKieClasspathContainer();

1. **Create a KieBase:**
   * The KieContainer can be used to get a KieBase, which contains the compiled rules.

KieBase kBase = kContainer.getKieBase("myKieBase");

1. **Create and Use a KieSession:**
   * From the KieBase, create a KieSession to insert facts and fire rules.

KieSession kSession = kBase.newKieSession();

kSession.insert(new MyFact());

kSession.fireAllRules();

kSession.dispose();

**Summary**

* **KieModel:** Represents the structured model of rules and resources, generated during the build.
* **KieContainer:** The runtime container that manages KieBases and KieSessions.
* **KieBase:** A repository of all compiled rules that can be executed within KieSessions.

These components work together to provide a flexible and powerful environment for managing and executing business rules in Drools.