**Kie Stateful vs Stateless Session**

In Drools, KieSession is the primary interface for interacting with the Drools engine. It provides the environment for inserting facts, firing rules, and interacting with the rule base. There are two main types of sessions: Stateful and Stateless. Here’s a comparison of the two:

**1. Kie Stateful Session (KieSession)**

A Stateful session maintains state across multiple interactions, meaning that facts and rules are retained in the session until explicitly removed or the session is disposed of. This type of session is suitable for scenarios where you need to perform multiple operations or reason over the same set of data repeatedly.

**Characteristics:**

* **Stateful Memory:** Facts inserted into the session are retained and can be modified or removed later.
* **Multiple Fire Calls:** You can fire rules multiple times within the same session, and the session remembers the facts and their changes.
* **Long-Running Sessions:** Ideal for long-running processes where the same set of facts needs to be evaluated over time.
* **Handles Iterative Logic:** Rules can add or modify facts, which might trigger other rules in subsequent fireAllRules() calls.

**Usage Example:**

KieServices ks = KieServices.Factory.get();

KieContainer kContainer = ks.getKieClasspathContainer();

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

Order order = new Order(150.0);

kSession.insert(order);

kSession.fireAllRules();

order.setTotalAmount(200.0);

kSession.update(kSession.getFactHandle(order), order);

kSession.fireAllRules(); // Re-evaluates rules with updated facts

kSession.dispose(); // Clean up the session when done

**2. Kie Stateless Session (StatelessKieSession)**

A Stateless session, on the other hand, is a single-use, one-shot execution environment. It does not retain any state between invocations, meaning facts are discarded after the rules have been fired. This type of session is suitable for scenarios where you need to apply a set of rules to a set of data in a stateless manner, such as batch processing.

**Characteristics:**

* **Stateless Execution:** Once the rules are fired, all facts and changes are discarded. The session does not remember anything.
* **One-Shot Operation:** The session is typically used for single, isolated operations.
* **Simpler to Use:** No need to manage session disposal or fact handles, as everything is cleared after each invocation.
* **High Performance:** Suitable for high-throughput scenarios where state retention is unnecessary.

**Usage Example:**

KieServices ks = KieServices.Factory.get();

KieContainer kContainer = ks.getKieClasspathContainer();

StatelessKieSession kSession = kContainer.newStatelessKieSession("ksession-rules");

Order order = new Order(150.0);

kSession.execute(order); // Fire rules and process the order in a stateless manner

// No need to dispose of the session, as it is stateless

**Key Differences:**

You are processing a batch of data and do not need to maintain session state between executions.

You want to avoid managing session lifecycle explicitly.

By choosing the appropriate session type, you can optimize the performance and behavior of your Drools-based applications.