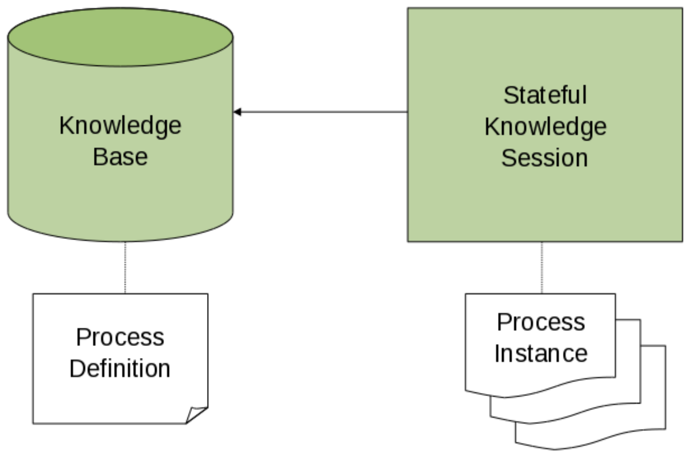
Understanding of Knowledge API



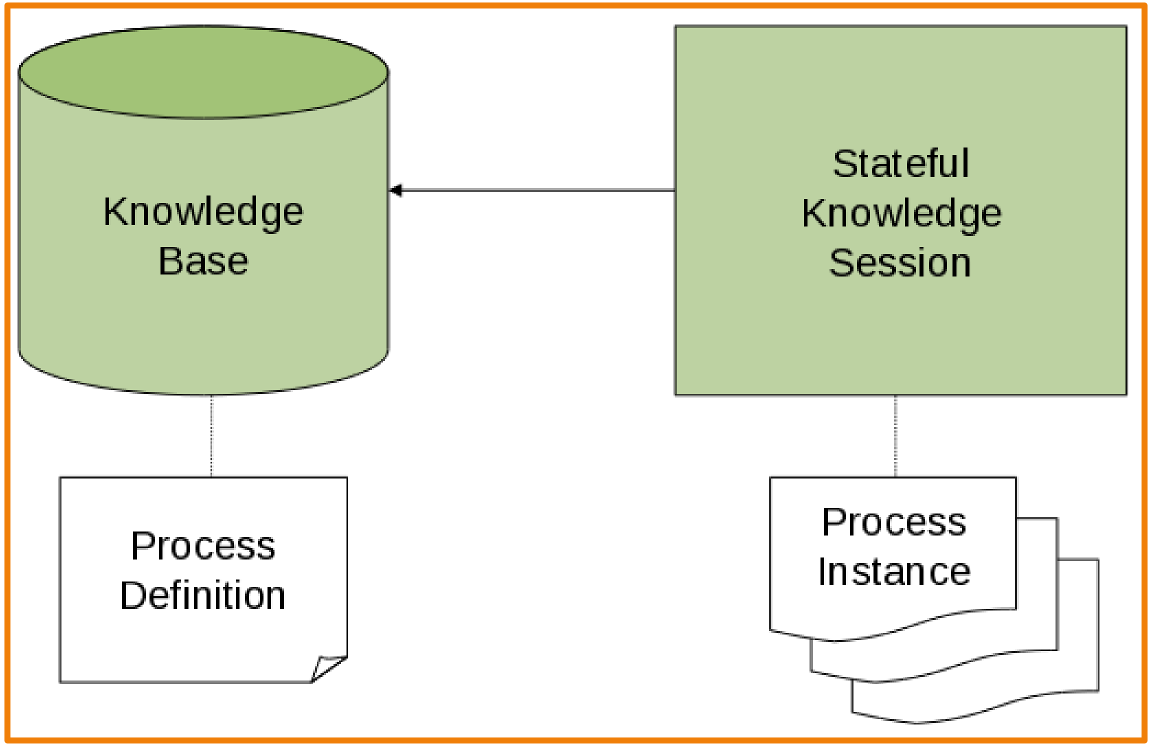
Agenda

Core Engine API

Core Engine API

- To interact with the jBPM engine (for example, to start a process), you need to set up a session. This session will be used to communicate with the jBPM engine. A session needs to have a reference to a KIE base, which contains a reference to all the relevant process definitions. This KIE base is used to look up the process definitions whenever necessary.
- To create a session, you first need to create a KIE base, load all the necessary process definitions (this can be from various sources, like from classpath, file system or process repository) and then instantiate a session.
- Once you have set up a session, you can use it to start executing processes. Whenever a process is started, a new process instance is created (for that process definition) that maintains the state of that specific instance of the process.

Core Engine API



KieBase

- The jBPM API allows you to first create a KIE base. This KIE base should include all your process definitions that might need to be executed by that session.
- To create a KIE base, use a KieHelper to load processes from various resources (for example from the classpath or from the file system), and then create a new KIE base from that helper.

KieBase

- The following code snippet shows how to create a KIE base consisting of only one process definition (using in this case a resource from the classpath).

```
KieHelper kieHelper = new KieHelper();
KieBase kieBase = kieHelper
                  .addResource(ResourceFactory.newClassPathResource("MyProcess.bpmn"))
                  .build();
```

KieSession

- Once you’ve loaded your KIE base, you should create a session to interact with the engine. This session can then be used to start new processes, signal events, etc. The following code snippet shows how easy it is to create a session based on the previously created KIE base, and to start a process (by id).

```
KieSession ksession = kbase.newKieSession();
ProcessInstance processInstance = ksession.startProcess("com.sample.MyProcess");
```

RuntimeManager

RuntimeManager will ensure that regardless of the strategy it will provide same capabilities when it comes to initialization and configuration of the RuntimeEngine. That means

- KieSession will be loaded with same factories (either in memory or JPA based)
- WorkItemHandlers will be registered on every KieSession (either loaded from db or newly created)
- Event listeners (Process, Agenda, WorkingMemory) will be registered on every KieSession (either loaded from db or newly created)
- TaskService will be configured with:
  - JTA transaction manager
  - same entity manager factory as for the KieSession
  - UserGroupCallback from environment

On the other hand, RuntimeManager maintains the engine disposal as well by providing dedicated methods to dispose RuntimeEngine when it's no more needed to release any resources it might have acquired.

Services

On top of RuntimeManager API a set of high level services has been provided from jBPM version 6.2. These services are meant to be the easiest way to embed (j)BPM capabilities into custom application. A complete set of modules are delivered as part of these services. They are partitioned into several modules to ease thier adoptions in various environments.

- **jbpm-services-api** - contains only api classes and interfaces
- **jbpm-kie-services** - rewritten code implementation of services api - pure java, no framework dependencies
- **jbpm-services-cdi** - CDI wrapper on top of core services implementation
- **jbpm-services-ejb-api** - extension to services api for ejb needs
- **jbpm-services-ejb-impl** -EJB wrappers on top of core services implementation
- **jbpm-services-ejb-timer** - scheduler service based on EJB TimerService to support time based operations e.g. timer events, deadlines, etc
- **jbpm-services-ejb-client** - EJB remote client implementation - currently only for JBoss Service modules are grouped with its framework dependencies, so developers are free to choose which one is suitable for them and use only that.

Configuration

- There are several control parameters available to alter engine default behavior. This allows to fine tune the execution for the environment needs and actual requirements.
- All of these parameters are set as JVM system properties, usually with -D when starting program e.g. application server.

Runtime Management

Runtime Management

- **Deployments**
- **Process Deployments**

KIE REST API

KIE REST API

- jBPM provides a KIE Server REST API that you can use to interact with your KIE containers and business assets (such as business rules, processes, and solvers) in jBPM without using the jBPM Workbench user interface. This API support enables you to maintain your jBPM resources more efficiently and optimize your integration and development with jBPM.
- With the KIE Server REST API, you can perform the following actions:
  - Deploy or dispose KIE containers
  - Retrieve and update KIE container information
  - Return KIE Server status and basic information
  - Retrieve and update business asset information
  - Execute business assets (such as rules and processes)

JAVA CLIENT API

Java Client API

- jBPM provides a KIE Server Java client API that enables you to connect to KIE Server using REST protocol from your Java client application. You can use the KIE Server Java client API as an alternative to the KIE Server REST API to interact with your KIE containers and business assets (such as business rules, processes, and solvers) in jBPM without using the jBPM Workbench user interface.
- This API support enables you to maintain your jBPM resources more efficiently and optimize your integration and development with jBPM.
- With the KIE Server Java client API, you can perform the following actions also supported by the KIE Server REST API:
  - Deploy or dispose KIE containers
  - Retrieve and update KIE container information
  - Return KIE Server status and basic information
  - Retrieve and update business asset information
  - Execute business assets (such as rules and processes)

Quiz

KIE API

Quiz

You can connect jBPM in loosely fashion using

a) REST API

b) Java Client API

c) Connectors

d) Java Integration

Answers

You can connect jBPM in loosely fashion using

[✓ a) REST API]

b) Java Client API

c) Connectors

d) Java Integration

Quiz

jbpm-services-api contains only api classes and interfaces

a) True

b) False

Answers

jbpm-services-api contains only api classes and interfaces

a) True

b) False

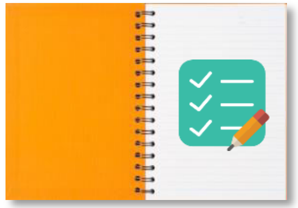
Sample jBPM Project

Conclusion

Summary

In this session, you have learnt:

- Core Engine API
- Runtime Management
- KIE REST API
- Java Client API
- Sample jBPM Project



India : +91-7847955955

US : 1-800-216-8930 (TOLL FREE)

support@intellipaat.com

24X7 Chat with our Course Advisor