## Demo入门

https://zhuanlan.zhihu.com/p/28528925

## 官网

https://docs.jboss.org/drools/release/6.5.0.Final/drools-docs/html\_single/

关注章节13 spring集成、章节8 Rule定义

**KIE:Knowledge Is Everything**

KIE 是一组项目的集合，这个集合用于提供业务的自动化和管理的解决方案。

1.Drools： 业务规则管理系统，具有基于前向链接和后向链接推理的规则引擎，允许快速可靠地评估业务规则和复杂的事件处理。规则引擎也是创建专家系统的基本构建块，该专家系统在人工智能中是模拟人类专家决策能力的计算机系统。

2.JBPM ： 一个灵活的业务流程管理套件，通过一个个流程步骤来实现你的业务目标。

3.OptaPlanner ：一种约束求解器，可优化员工排班，车辆路径，任务分配和云优化等用例。

4.Drools Workbench ： 一个功能齐全的Web应用程序，用于自定义业务规则(Drools)和流程(jBPM)的可视化组合。( Workbench是管理Drools和jBPM资产的Web应用程序和存储库)

5.UberFire：一个基于Web的工作台框架，受Eclipse富客户端平台的启发。

**依赖的JAR包**

The following is a description of the important libraries that make up JBoss Drools

**• knowledge-api.jar** - this provides the interfaces and factories. It also helps clearly show what is

intended as a user API and what is just an engine API.

**• knowledge-internal-api.jar** - this provides internal interfaces and factories.

**• drools-core.jar** - this is the core engine, runtime component. Contains both the RETE engine

and the LEAPS engine. This is the only runtime dependency if you are pre-compiling rules (and deploying via Package or RuleBase objects).

• **drools-compiler.jar** - this contains the compiler/builder components to take rule source, and build executable rule bases. This is often a runtime dependency of your application, but it need not be if you are pre-compiling your rules. This depends on drools-core.

• **drools-jsr94.jar** - this is the JSR-94 compliant implementation, this is essentially a layer over

the drools-compiler component. Note that due to the nature of the JSR-94 specification, not all features are easily exposed via this interface. In some cases, it will be easier to go direct to the Drools API, but in some environments the JSR-94 is mandated.

• **drools-decisiontables.jar** - this is the decision tables 'compiler' component, which uses the

drools-compiler component. This supports both excel and CSV input formats

**Kie Project**

A Kie Project has the structure of a normal Maven project with the only peculiarity of including

a **kmodule.xm**l file defining in a declaratively way the KieBases and KieSessions that can be

created from it.

This file has to be placed in the resources **/META-INF** folder of the Maven project

while all the other Kie artifacts, such as **DRL or a Excel files, must be stored in the resources**

**folder or in any other subfolder under it.**

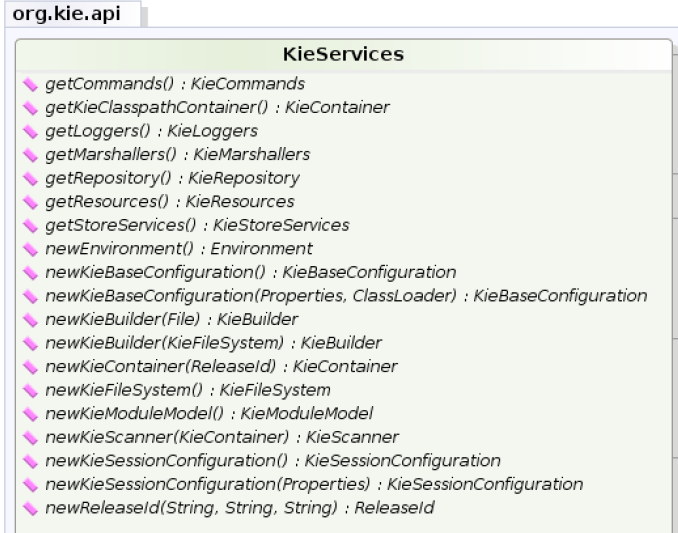
**KieServices** is the interface from where it possible to access all the Kie building and runtime

facilities:

**Creating a KieContainer from the classpath**

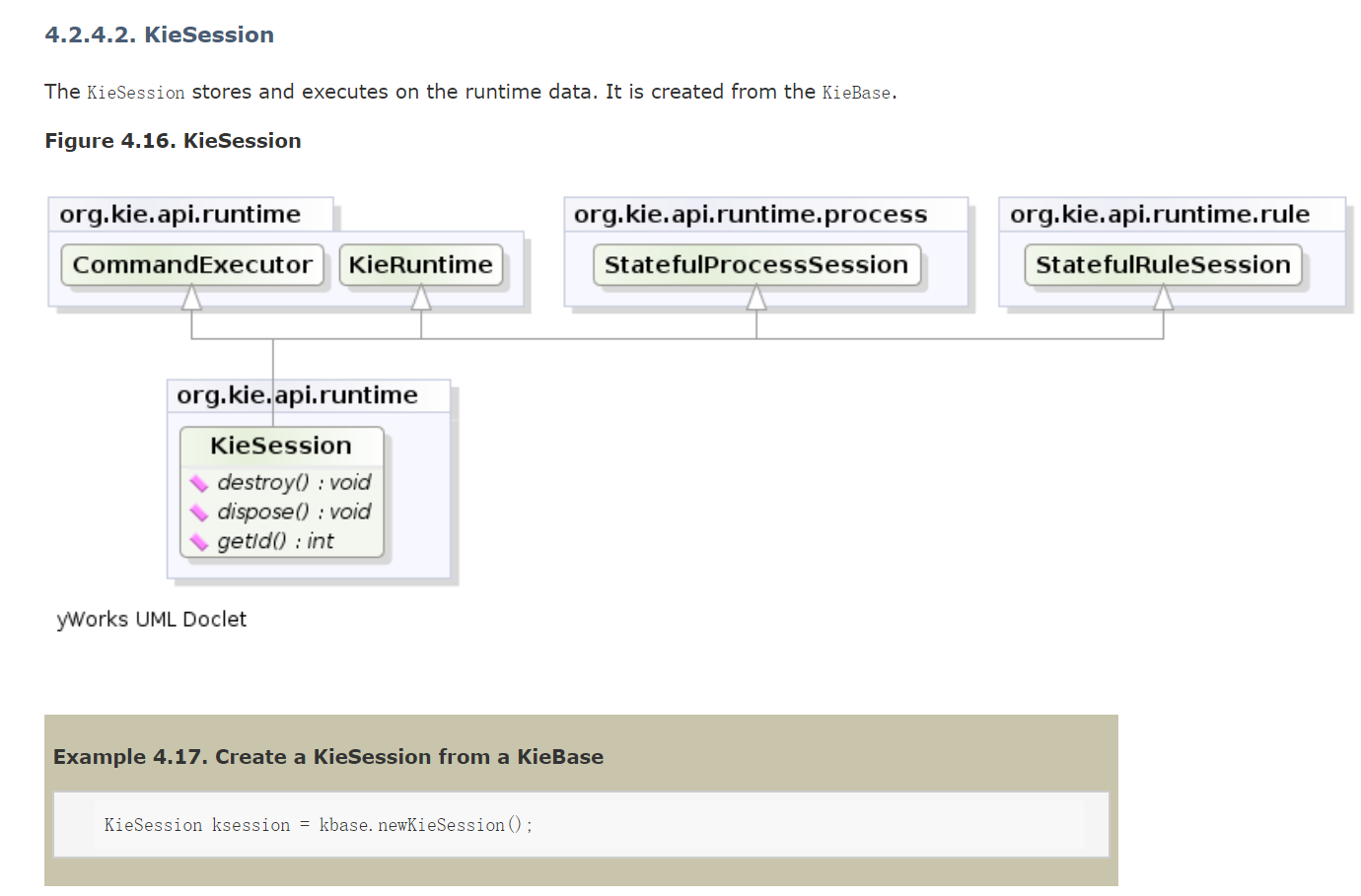
KieServices kieServices = KieServices.Factory.get();

KieContainer kContainer = kieServices.getKieClasspathContainer();









KieServices kieServices = KieServices.Factory.get();

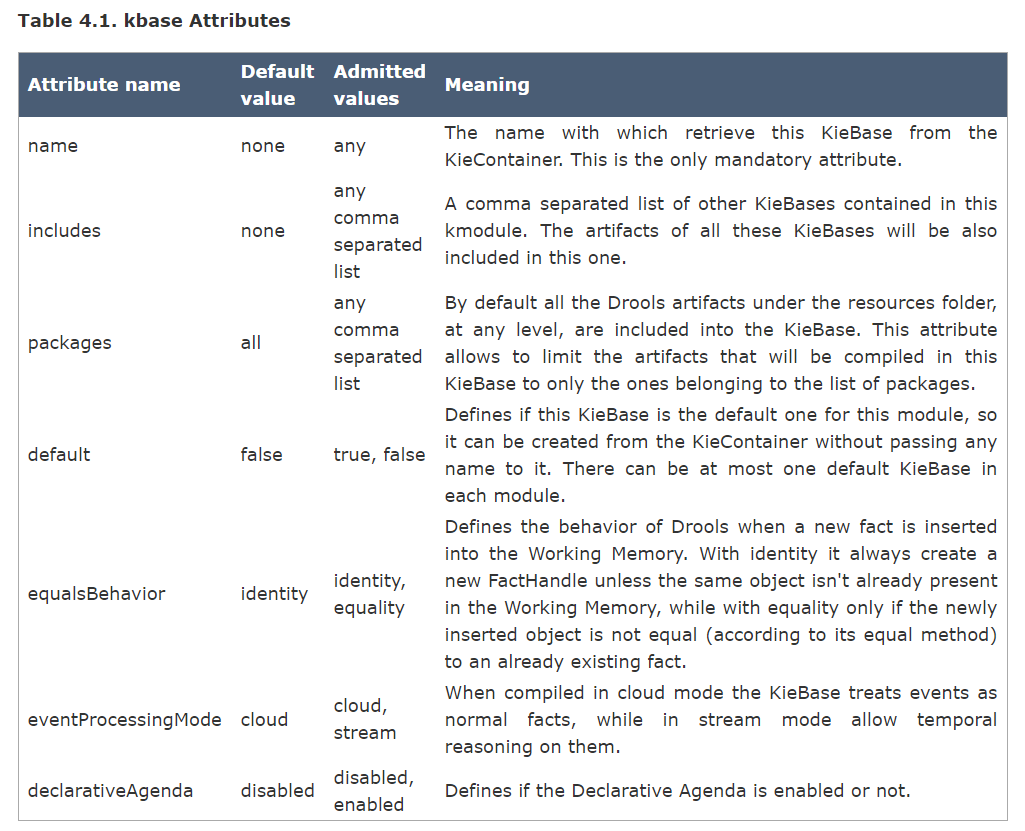
KieContainer kContainer = kieServices.getKieClasspathContainer();

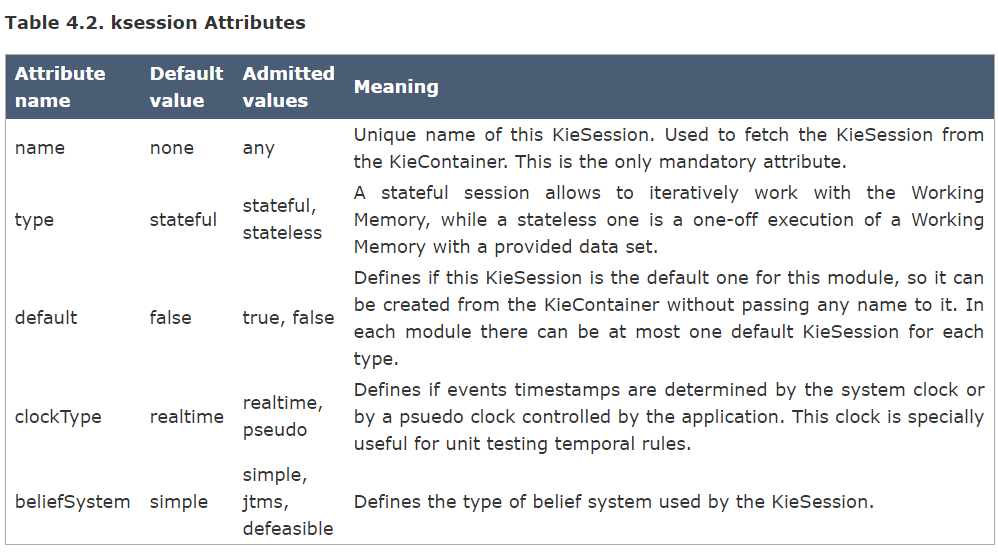
KieBase kBase1 = kContainer.getKieBase("KBase1");

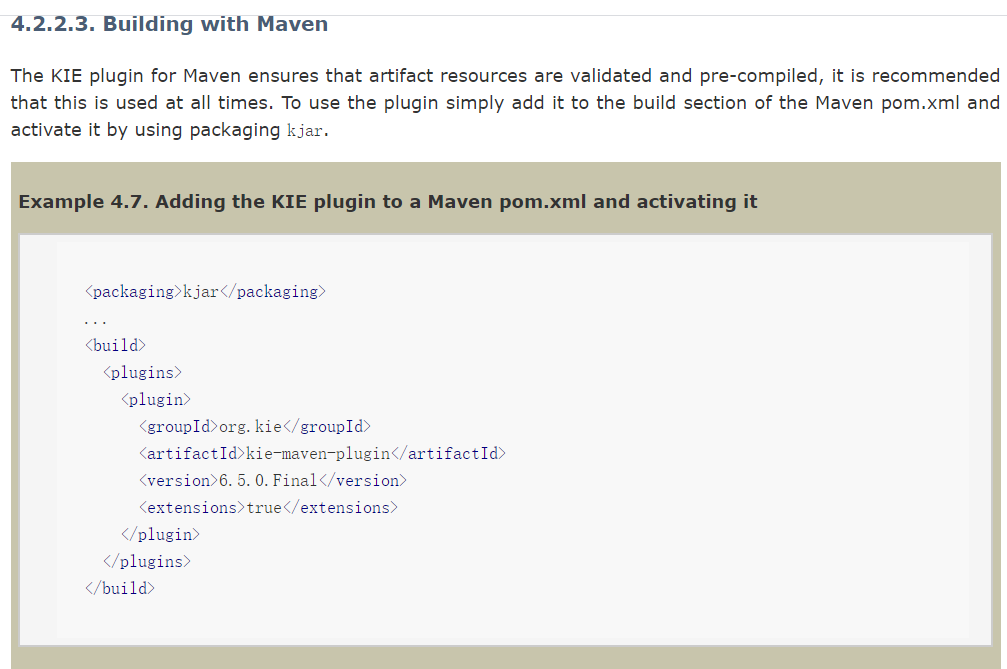
KieSession kieSession1 = kContainer.newKieSession("KSession2\_1");

StatelessKieSession kieSession2 = kContainer.newStatelessKieSession("KSession2\_2");









### Event Listener

The event package provides means to be notified of rule engine events, including rules firing, objects being asserted, etc. This allows separation of logging and auditing activities from the main part of your application (and the rules).

The KieRuntimeEventManager interface is implemented by the KieRuntime which provides two interfaces, RuleRuntimeEventManager and ProcessEventManager. We will only cover the RuleRuntimeEventManager here.

ksession.addEventListener( **new** DefaultAgendaEventListener() {

**public** **void** **afterMatchFired**(AfterMatchFiredEvent event) {

**super**.afterMatchFired( event );

System.out.println( event );

}

});

The events currently supported are:

MatchCreatedEvent

MatchCancelledEvent

BeforeMatchFiredEvent

AfterMatchFiredEvent

AgendaGroupPushedEvent

AgendaGroupPoppedEvent

ObjectInsertEvent

ObjectDeletedEvent

ObjectUpdatedEvent

ProcessCompletedEvent

ProcessNodeLeftEvent

ProcessNodeTriggeredEvent

ProcessStartEvent

### 持久化

4.2.4.9章节



