

AUTHINE 奥哲

深圳奥哲网络科技有限公司

# 二次开发指导

## V1.0

二〇一九年三月

版权所有(c)2010-2017深圳奥哲网络科技有限公司

## 目录

1 二次开发指导.....	3
1.1 目的.....	3
1.2 内容.....	3
1.3 Portal 端的后端代码开发.....	3
1.3.1 JDK 版本.....	3
1.3.2 Maven 版本.....	3
1.3.3 开发环境部署.....	3
1.3.4 更换或升级产品.....	5
1.3.5 新增类或者属性.....	5
1.3.6 新增 Controller 类.....	5
1.3.7 修改 Controller 类.....	6
1.4 引擎端事件开发.....	8

## 1 二次开发指导

---

### 1.1 目的

尽量避免在项目上进行二次开发中的定制化代码与产品的代码混合在一起，导致产品的 BUG 需要修复或者需要升级时不能做到无缝升级的问题。

### 1.2 内容

本文包括以下两大块：

- Portal 端的后端代码开发；
- 引擎端事件开发；

（开发过程涉及到 Spring、SpringMVC 和 Maven 等技术，请自行了解）

### 1.3 Portal 端的后端代码开发

#### 1.3.1 JDK 版本

请使用 JDK1.8 及以上的，建议 1.8；

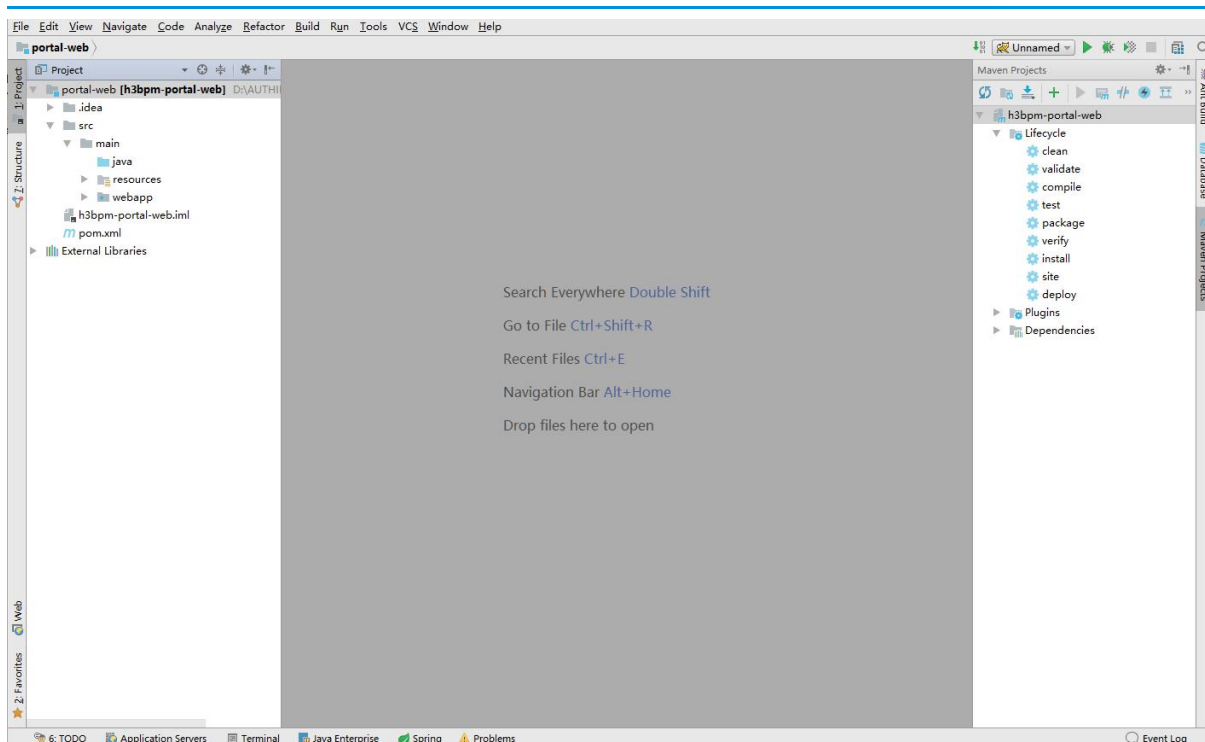
#### 1.3.2 Maven 版本

使用 3.3.9 及以上版本，相关配置请自行处理；

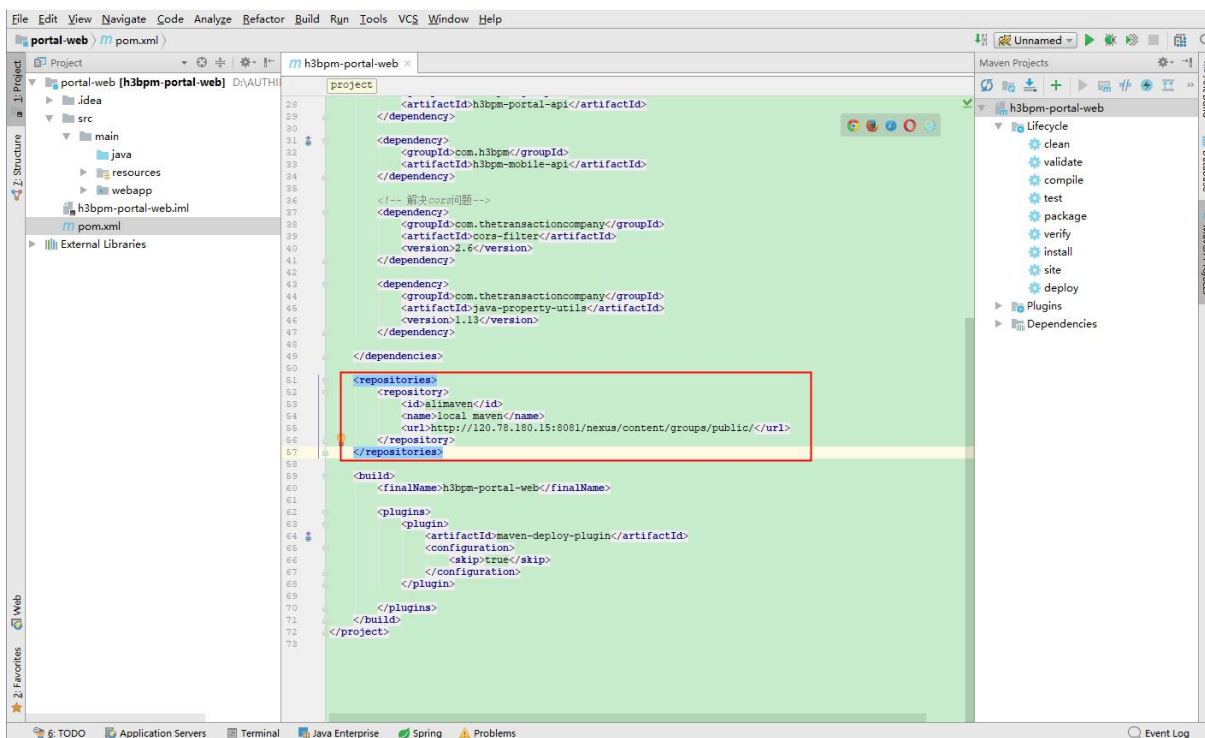
H3BPM 的 JAR 包仓库地址：<http://120.78.180.15:8081/nexus/content/groups/public/>

#### 1.3.3 开发环境部署

当拿到原始的开发项目（portal-web）后请自行解压然后导入到 IDE 中（以下均以 Ideal 为例）；如下图



配置 Maven 仓库地址，可以在 Maven 的配置文件中配置仓库地址也可以在 pom.xml 里进行单独配置，如下图

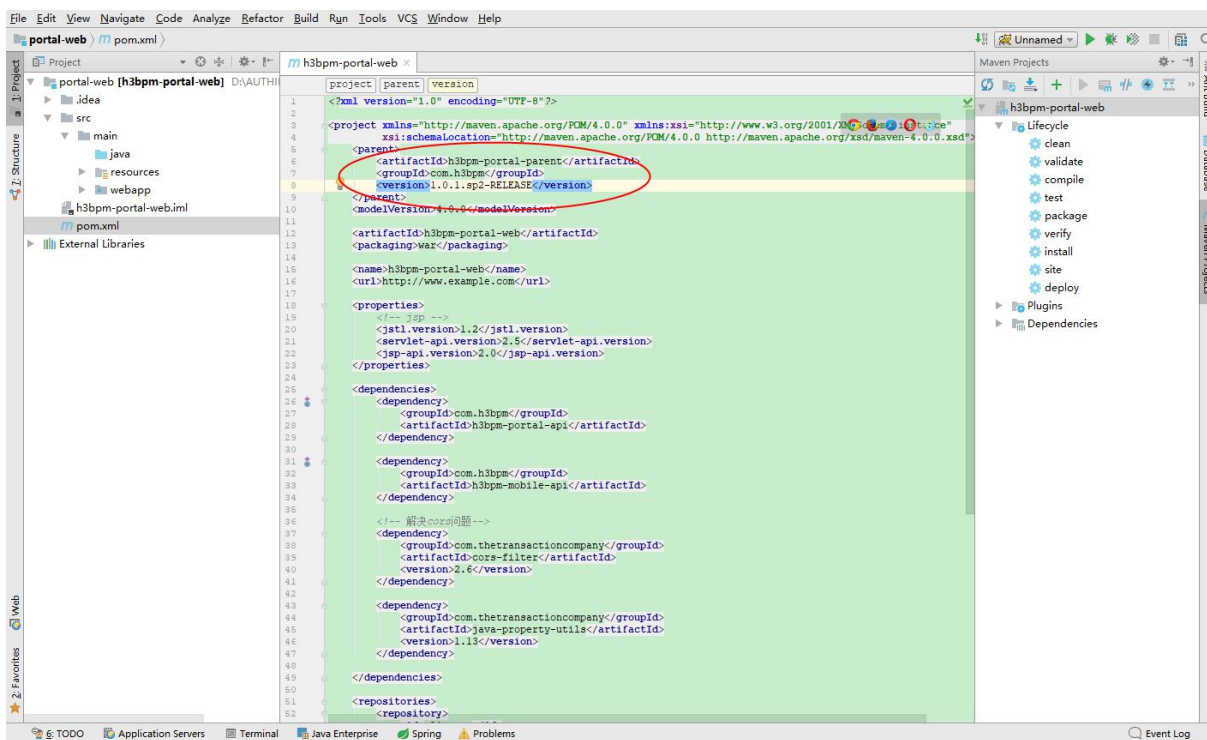


```
<repositories>
  <repository>
    <id>alimaven</id>
    <name>local maven</name>
    <url>http://120.78.180.15:8081/nexus/content/groups/public/</url>
  </repository>
</repositories>
```

```
</repository>
</repositories>
```

### 1.3.4 更换或升级产品

当需要更换或者升级产品时，只需要修改 pom.xml 文件更换引入的 JAR 包的版本号即可，如下图



```
<parent>
<artifactId>h3bpm-portal-parent</artifactId>
<groupId>com.h3bpm</groupId>
<version>1.0.1.sp2-RELEASE</version>
</parent>
```

当产品有升级时只要修改此处版本号，然后重新编译打包就即可。

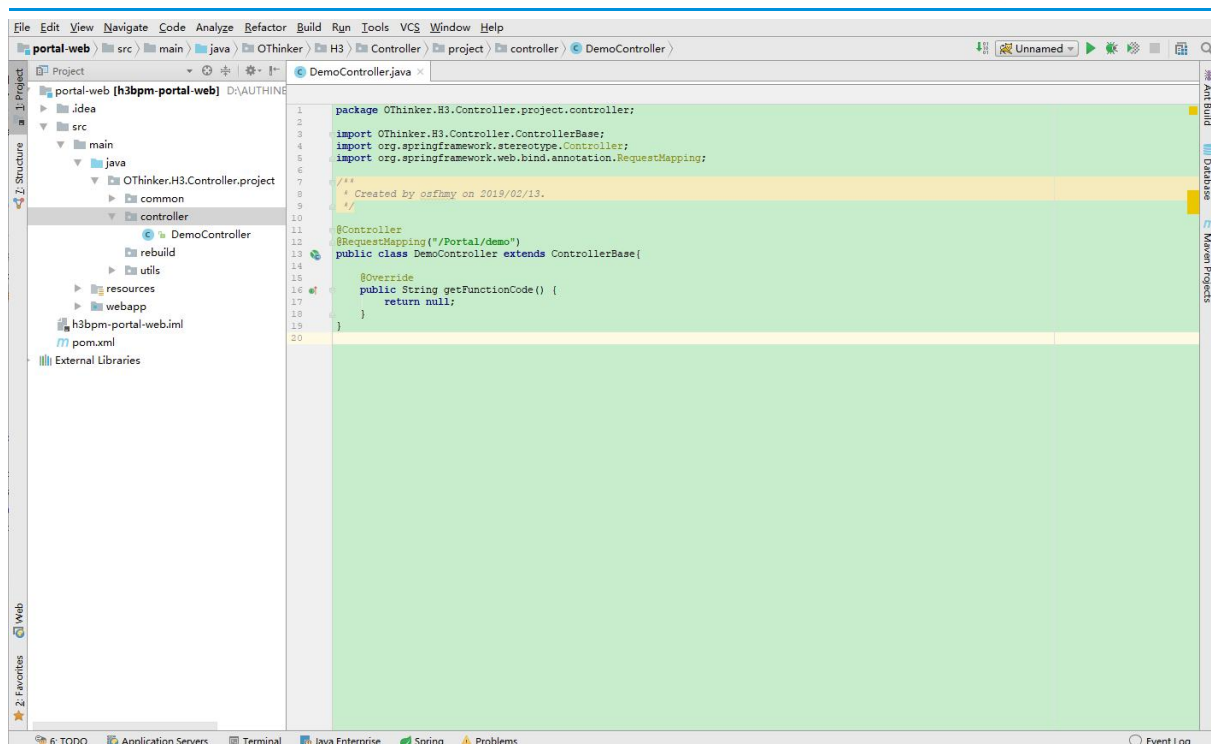
### 1.3.5 新增类或者属性

当产品中有些类不满足当前项目上需求时需要增加属性时，可以采用继承或者重写都是可以的。

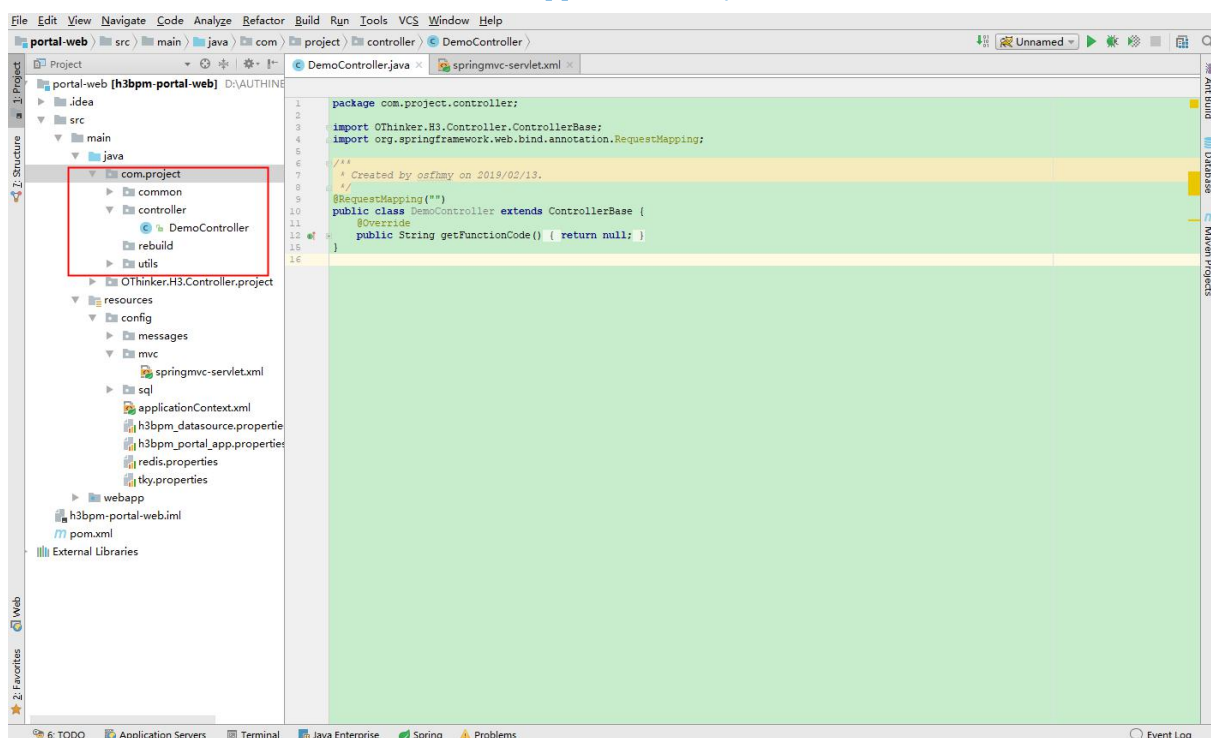
### 1.3.6 新增 Controller 类

有两种方式，第一种方式不需要修改 Spring 的配置文件，此方式比较简单只需要遵循路径的规则就可以；

新增代码按照如下路径进行管理 `0Thinker.H3.Controller.project.controller` 其中 project 按照项目名称自行定义作为整个新项目的根路径，效果如下图：



另外一种方式需要修改配置文件 `application.yml`，在配置文件中增加扫描路径，



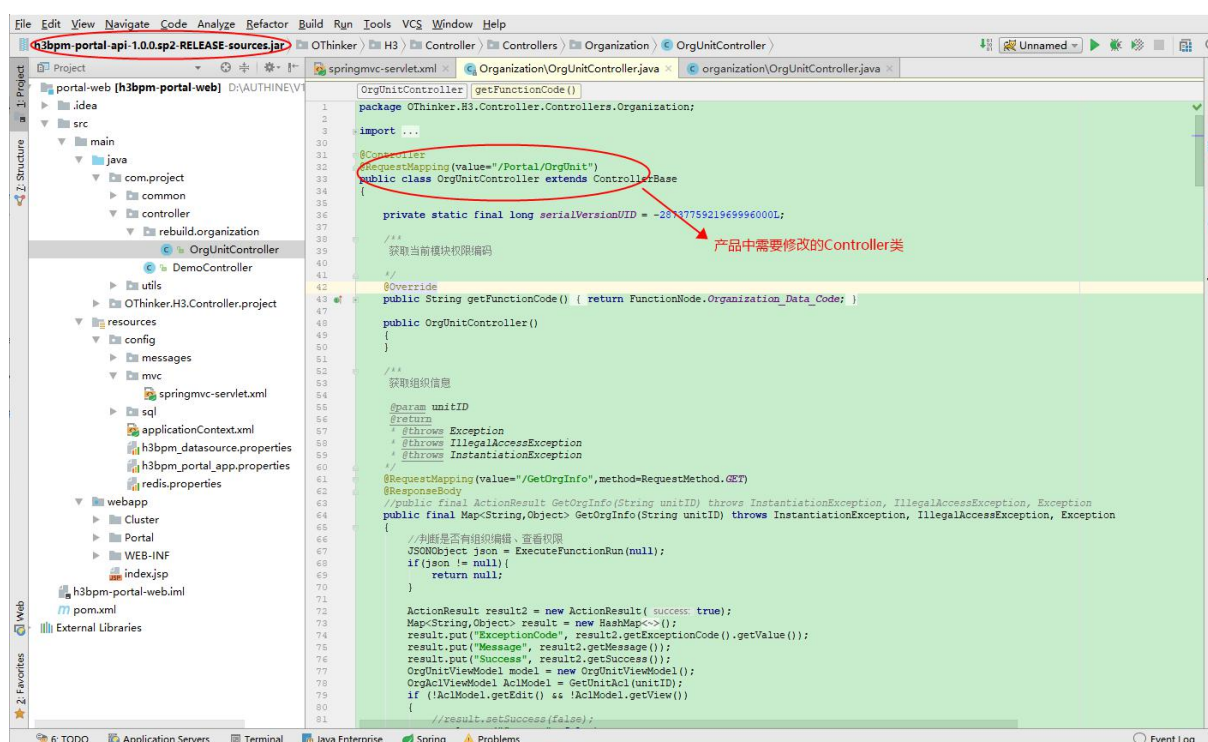
### 1.3.7 修改 Controller 类

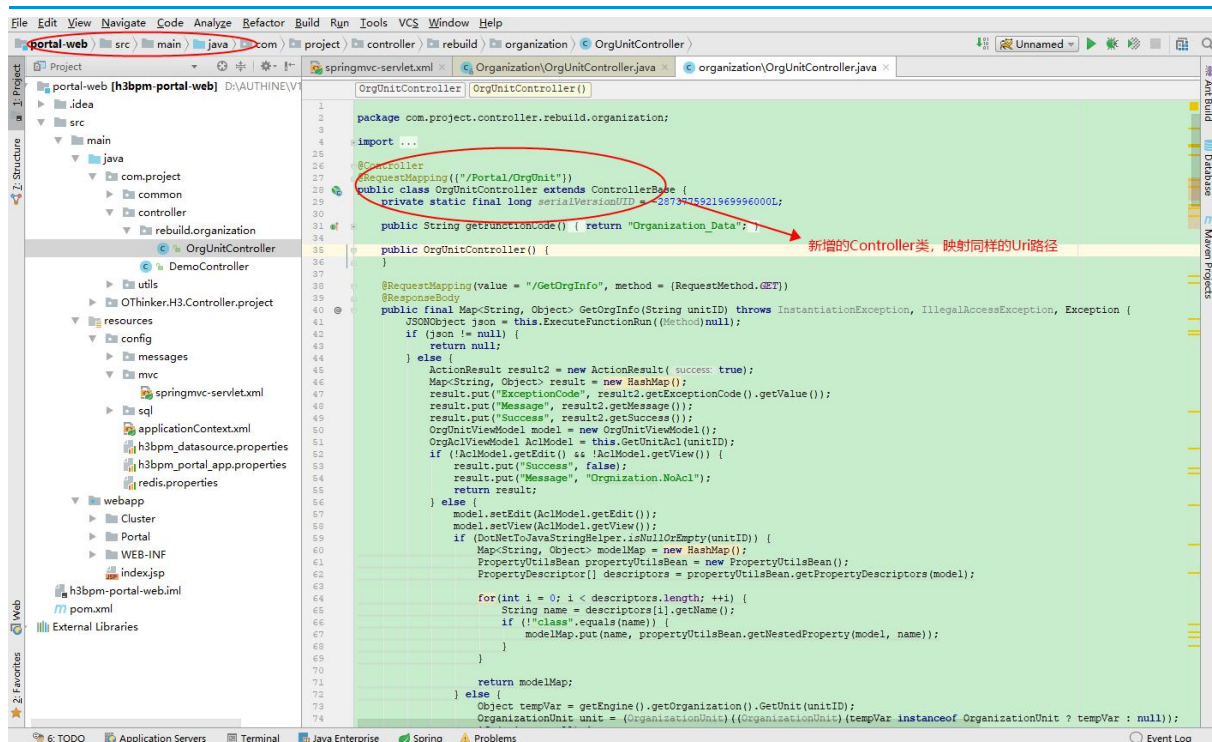
当需要修改产品的 Controller 类时会比较麻烦，因为此处会涉及到产品的升级和更换带来的问题，同时目前产品不直接提供源码的情况下做修改也时比较麻烦，针对这种情况，目前有两种方式可以参考。



第二种方式是修改配置文件 `application.yml`，对扫描的类进行过滤，新增一个 `Controller` 类映射同样的 `URI` 地址然后让 `SpringMVC` 在启动的时候过滤原来的 `Controller` 就可以，这样的方式有个问题就是如果需要对多个 `Controller` 进行修改需要添加多个过滤类。

此处以第二种方式来处理，比如需要对 `OThinker.H3.Controller.Controllers.Organization.OrgUnitController` 类做修改，则我们可以在新项目的新建一个类 `om.project.rebuild.organization.OrgUnitController`（类名可以随便取或者和原来一致），然后把 `OThinker.H3.Controller.Controllers.Organization.OrgUnitController` 的内容全部拷贝到新的类中，然后修改我们需要修改的方法即可，如下各图所示：





新增类使用了rebuild作为开始路径可以明确在这个路径下的类都是对原产品的类做修改的。

**备注：**如果找不到源码，请查看对应的 IDE 的 maven 帮助或者直接使用 maven 的命令 `dependency:sources` 来下载。

## 1.4 引擎端事件开发

引擎提供了大概 9 个事件的接口

### ✧ IEngineEventHandler

引擎级对象的事件处理接口，比如：新建 WorkItem 事件等。注意：这里的事件都是滞后的事件，并不是实时的事件，比如：流程实例创建事件，是在新建的流程实例保存进数据库中才被触发的，而且是异步触发的。如果事件处理程序耗用大量的 CPU，那么会影响引擎的性能；如果事件处理程序抛出异常，那么引擎会忽略掉这个异常。

### ✧ ICirculateItemEventHandler

传阅项的事件处理接口，注意：这里的事件都是滞后的事件，并不是实时的事件，比如：工作项创建事件，是在新建的工作项保存进数据库中才被触发的，而且是异步触发的。如果事件处理程序耗用大量的 CPU，那么会影响引擎的性能；如果事件处理程序抛出异常，那么引擎会忽略掉这个异常。

### ✧ ICirculateItemFinishedEventHandler

传阅项的事件处理接口，注意：这里的事件都是滞后的事件，并不是实时的事件，比如：工作项创建事件，是在新建的工作项保存进数据库中才被触发的，而且是异步触发的。如果事件处理程序耗用大量的 CPU，那么会影响引擎的性能；如果事件处理程序



抛出异常，那么引擎会忽略掉这个异常。

#### ✧ IInstanceEventHandler

流程实例的事件处理接口，注意：这里的事件都是滞后的事件，并不是实时的事件，比如：流程实例创建事件，是在新建的流程实例保存进数据库中才被触发的，而且是异步触发的。如果事件处理程序耗用大量的 CPU，那么会影响引擎的性能；如果事件处理程序抛出异常，那么引擎会忽略掉这个异常。

#### ✧ IMessageEventHandler

流程引擎 Message 处理事件接口

#### ✧ INotificationEventHandler

消息通知 EventHandler 接口

#### ✧ IOrganizationEventHandler

组织的事件处理接口，注意：这里的事件都是滞后的事件，并不是实时的事件，比如：组织创建事件，是在新建的组织保存进数据库中才被触发的，而且是异步触发的。如果事件处理程序耗用大量的 CPU，那么会影响引擎的性能；如果事件处理程序抛出异常，那么引擎会忽略掉这个异常。

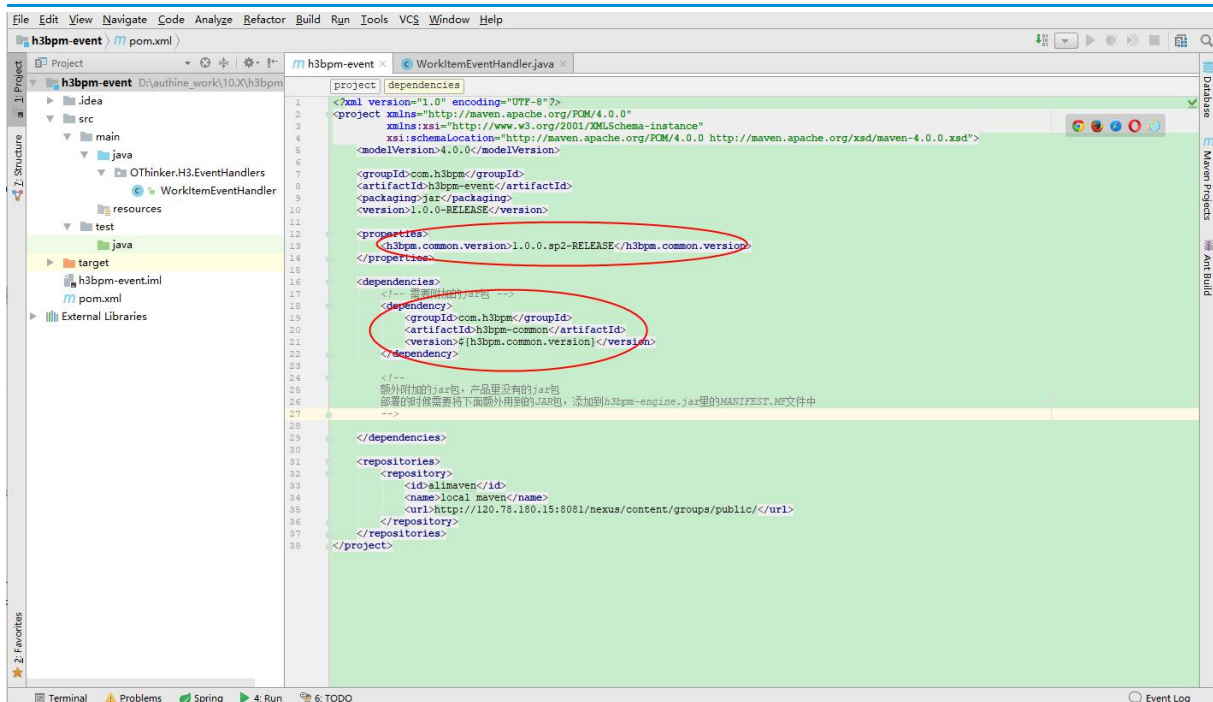
#### ✧ IWorkItemEventHandler

工作项的事件处理接口，注意：这里的事件都是滞后的事件，并不是实时的事件，比如：工作项创建事件，是在新建的工作项保存进数据库中才被触发的，而且是异步触发的。如果事件处理程序耗用大量的 CPU，那么会影响引擎的性能；如果事件处理程序抛出异常，那么引擎会忽略掉这个异常。

#### ✧ IWorkItemFinishedEventHandler

工作项的事件处理接口，注意：这里的事件都是滞后的事件，并不是实时的事件，比如：工作项创建事件，是在新建的工作项保存进数据库中才被触发的，而且是异步触发的。如果事件处理程序耗用大量的 CPU，那么会影响引擎的性能；如果事件处理程序抛出异常，那么引擎会忽略掉这个异常。

此处以工作项事件作为示例，先自行新建一个 java 工程，此处以 maven 工程为例，在 pom.xml 中添加 h3bpm-common 包和对应的版本。如下图所示：



当在开发过程中如果有新的 JAR 包需要引入到开发中，部署的时候需要将额外用到的 JAR 包，添加到 h3bpm-engine.jar 里的 MANIFEST.MF 文件中。

实现事件接口的类必须在 `OThinker.H3.EventHandlers` 路径下，开发完成后打包成 Jar 包同时将 class 文件放到引擎的固定目录 `h3bpm-engine\OThinker\H3\EventHandlers` 下即可，如下图

