

插件开发IDE

本文提供一个扩展包集成开发环境的下载及配置说明，内含Eclipse、Tomcat、Smartbi，便于大家快速开发 Smartbi 扩展包，尤其是用于调试后端 Java 代码。下载下来后需要参照下面配置说明替换成项目现场使用的smartbi环境（其实和[手动部署到tomcat](#)原理类似，只是非UI界面方式），譬如：修改知识库连接、替换license文件、替换smartbi war包为项目现场的。

如果您已经有了Eclipse工具，希望手动配置，请参考[手动配置插件开发环境](#)。

文档目录：

- [1、安装与启动IDE](#)
- [2、启动应用服务器](#)
 - [2.1 取得开发用的Smartbi-License文件](#)
 - [2.2 配置开发环境的知识库](#)
 - [2.3 启动 apache-tomcat-7.0.42](#)
 - [2.4 更新开发环境下smartbi版本](#)
- [3、开发新功能](#)
 - [3.1 检验新功能](#)
- [4、打包插件](#)
- [5、如何支持多扩展包开发](#)
- [6、解决扩展包修改后需要重启才生效的问题](#)

1、安装与启动IDE

请到以下地址下载IDE：链接：<https://pan.baidu.com/s/1LKRUFXcPScAVRabJIikCOQ> 密码：r58z

将下载回来的“SmartbiPluginIDE.zip”解压缩，进入解压后的目录执行“Startup.bat”即可启动IDE。启动后效果如下：

2、启动应用服务器

开发调试环境的原理是这样的：在 Eclipse 中默认加载了一个 SmartbiPluginApp 项目，该项目中包含了 apache-tomcat-7.0.42，在其下部署了 smartbi 应用。该 smartbi 应用在运行时，会加载 SmartbiPluginApp 扩展包项目，从而让开发人员对该项目下的代码进行调试。接下来我们需要进行配置，让该 smartbi 正常运行起来。

2.1 取得开发用的Smartbi-License文件

首先需要从[思迈特](#)处申请一个开发用的 Smartbi-License 文件。将取得的 Smartbi-License.xml 文件直接覆盖 /SmartbiPluginApp/apache-tomcat-7.0.42/bin/smartbi/ 目录下的同名文件。

2.2 配置开发环境的知识库

该开发环境下的 smartbi 需要有个知识库，修改 /SmartbiPluginApp/apache-tomcat-7.0.42/bin/smartbi/smartbi-config.xml 文件，设置知识库连接信息，主要修改<dao/>这一段。

```
<dao>
    <database-type>MYSQL</database-type>
    <server-name>localhost:6688</server-name>
    <database-name>smartbi-dev-ide</database-name>
    <username>admin</username>
    <pass>admin</pass>
    <encrypt-type>0</encrypt-type>
    <max-count>100</max-count>
    <init-count>2</init-count>
</dao>
```



2.3 启动 apache-tomcat-7.0.42

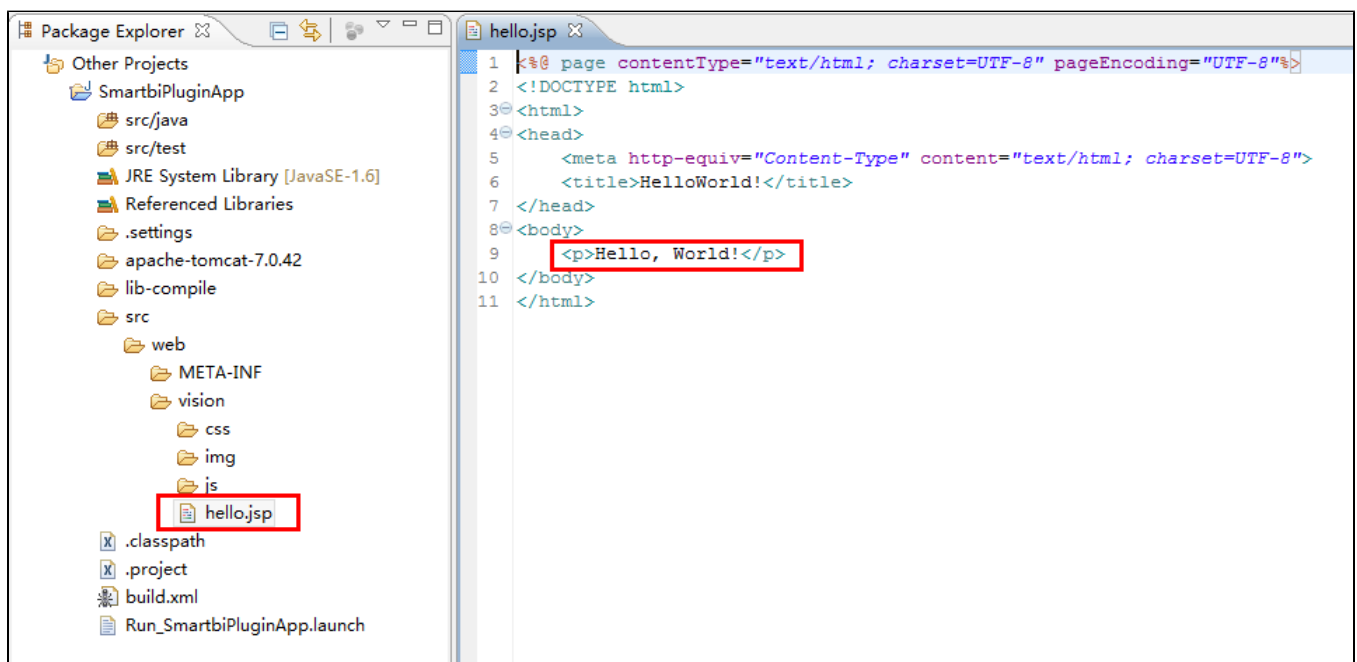
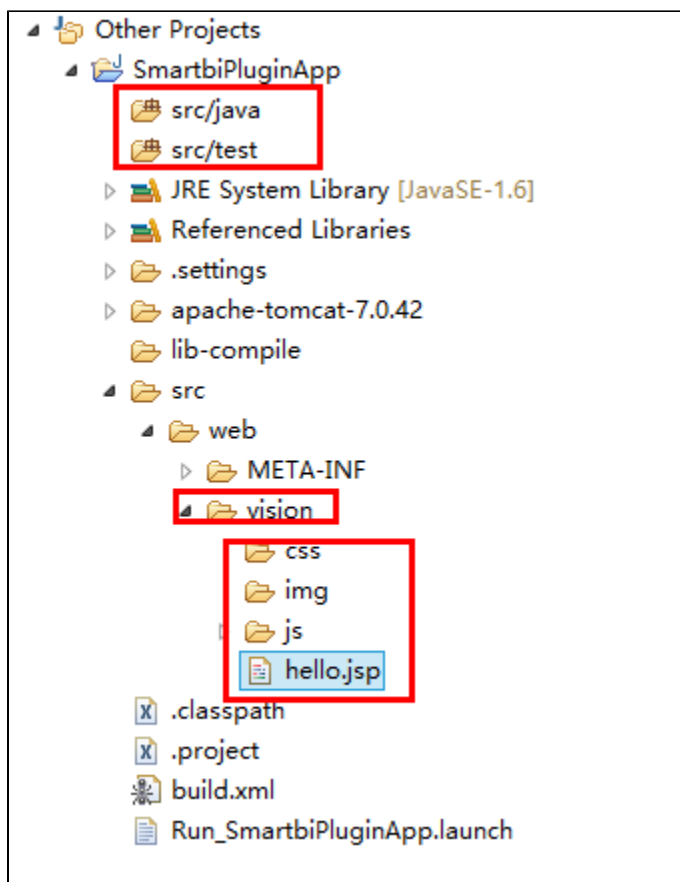
The screenshot shows an IDE with the 'Run' menu open. The 'Run_SmartbiPluginApp' option is highlighted. In the background, the 'smartbi-config.xml' file is visible, containing XML configuration for logging and database connections. The 'Run' menu options include 'New', 'Open', 'Open With', 'Show In', 'Copy', 'Copy Qualified Name', 'Paste', 'Delete', 'Remove from Context', 'Mark as Landmark', 'Build Path', 'Refactor', 'Import...', 'Export...', 'Refresh', 'Assign Working Sets...', 'Validate', 'Show in Remote Systems view', 'Run As', 'Debug As', 'Profile As', 'Team', 'Compare With', and 'Replace With'. The 'Run_SmartbiPluginApp' option is highlighted in the 'Run As' submenu.

下载回来的 IDE 中虽然自带了一个 smartbi，但是其版本比较早。在开发调试时，要求开发所基于的 smartbi 版本，应该与未来生产环境部署的 smartbi 版本一致。因此我们还需要升级 `/SmartbiPluginApp/apache-tomcat-7.0.42/webapps` 路径下的 smartbi，具体升级过程可以参考：[Smartbi 版本更新](#)

注意: /SmartbiPluginApp/apache-tomcat-7.0.42/webapps/smartbi/WEB-INF/extensions/extensions.list 文件在调试环境下需要修改, 以便让 smartbi 加载正在开发的代码。

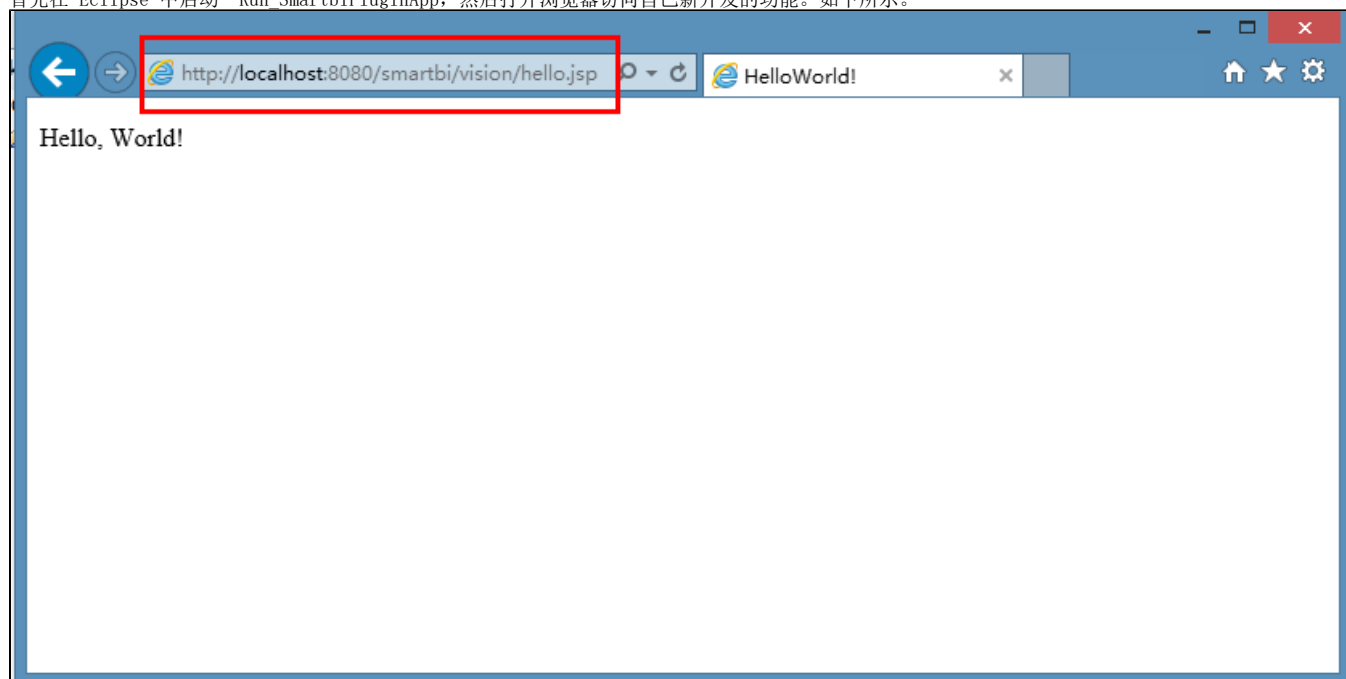
```
..\..\..\..\src\web
```

直接在 `SmartbiPluginApp` 项目中添加自己的功能，如果是后台代码添加到 `/SmartbiPluginApp/src/java` 目录下；如果是前端代码、图片、文件全部添加到 `/SmartbiPluginApp/src/web/vision` 路径下。



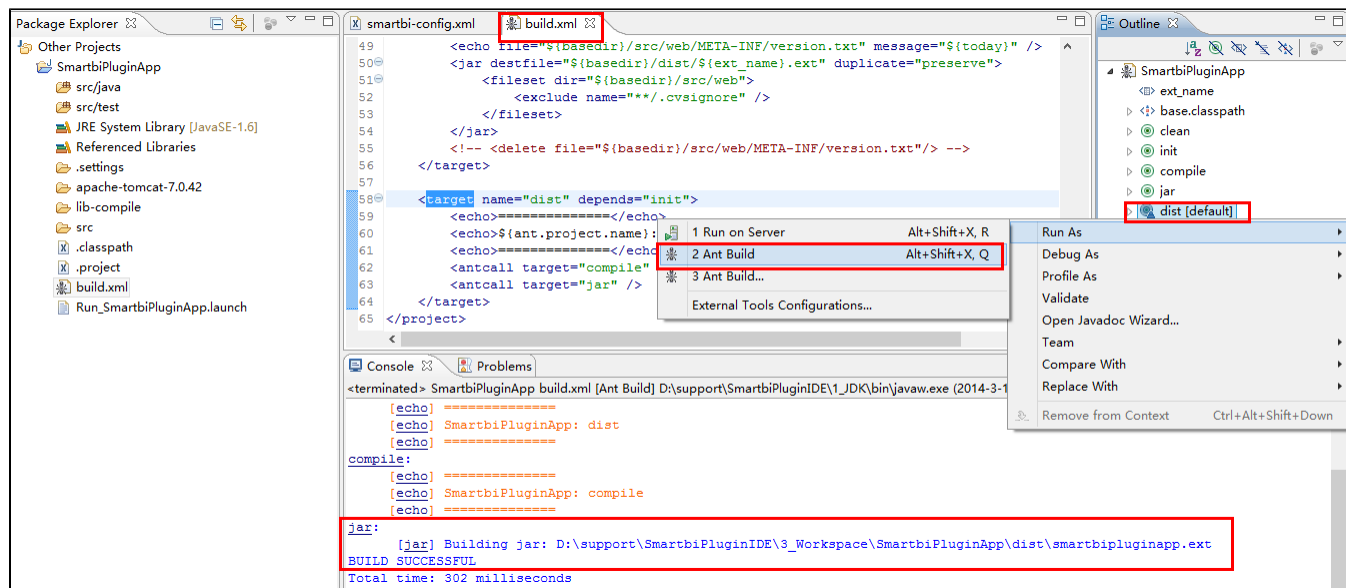
3.1 检验新功能

首先在 Eclipse 中启动 Run_SmartbiPluginApp, 然后打开浏览器访问自己新开发的功能。如下所示。



4、打包插件

开发调试全部完成后, 如何将所开发的成果部署到生产环境上。利用 Ant Build 配置文件 /SmartbiPluginApp/build.xml, 将整个项目的代码打包成一个 ext 扩展包。然后参考“[扩展包部署](#)”文档, 将打包后的扩展包部署到生产环境中。



5、如何支持多扩展包开发

如果项目现场还想多创建一个扩展包, 如何加入到这个调试环境中呢? 具体操作方法如下:

- 1、使用“[插件开发框架](#)”中提供的工具创建扩展包。比如, 创建的扩展包名为SecondPlugin, 存放在: D:/ext/ 目录下。
- 2、在上面的 Eclipse 开发环境中, 点击“File -> Import -> Existing Projects into Workspace” (或者右击Package Explorer -> Import), 将上面扩展包项目加入到Eclipse 环境中。
- 3、加载扩展包, 修改/SmartbiPluginApp/apache-tomcat-7.0.42/webapps/smartbi/WEB-INF/extensions/extensions.list 文件, 在其中加入一行 D:/ext/SecondPlugin/src/web。

```
..\..\..\..\src\web  
D:/ext/SecondPlugin/src/web
```

- 4、参照上面“2.3 启动 apache-tomcat-7.0.42”，这样这个新扩展包也加入到smartbi 中了，并且可以调试它。

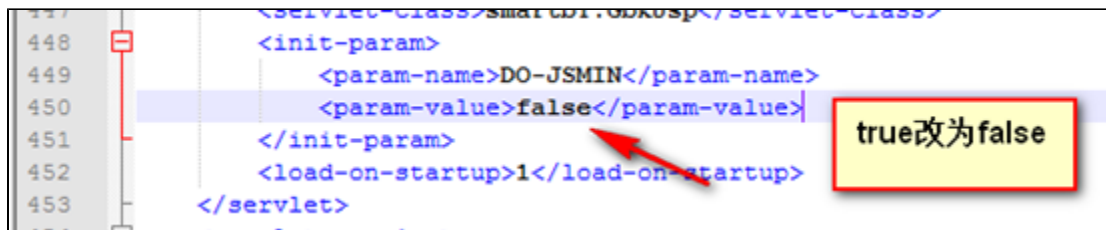
6、解决扩展包修改后需要重启才生效的问题

配置Smartbi，让其处在开发模式下即可。

- 修改/SmartbiPluginApp/apache-tomcat-7.0.42/webapps/smartbi/WEB-INF下的web.xml文件，在文件加入 DevEnv 对应的四行代码。在文件中找到 ProductName 这个 <context-param>，在其下面添加 DevEnv 对应的四行<context-param>，其值为 true。

```
<context-param>  
    <param-name>ProductName</param-name>  
    <param-value>Smartbi</param-value>  
</context-param>  
<context-param>  
    <param-name>DevEnv</param-name>  
    <param-value>true</param-value>  
</context-param>
```

- 并且修改如下图所示位置，配置 DO-JSMIN 参数，其值为 false。



至此，插件开发的IDE环境，全部配置完成了。