

第二课：扩展包的创建、配置及打包

- 1 创建扩展包
 - 1.1 说明
- 2 开发调试新功能
 - 2.1 基于 SmartbiPluginApp开发
 - 2.2 创建扩展包开发
 - 2.2.1 配置扩展包
 - 2.2.2 断点调试
 - 2.2.3 启动 Smartbi应用
- 3 打包扩展包

1 创建扩展包

1.1 说明

因为 Smartbi 扩展包里面的目录结构还是有一定规范的，虽然可以手动使用Eclipse 开发工具，通过“File -> New -> project ...”创建一个扩展包项目工程，但还是得手动按照 Smartbi 的扩展包目录规范创建目录及文件，稍显麻烦，而且可能也不会记得要创建些什么文件以及文件的配置结构是什么。

所以Smartbi 提供了一个工具用来创建扩展包项目，自动将需要的文件及目录创建好，创建好后手动使用Eclipse中的“File ->Import -> Existing Projects Info Workspace ...”导入到开发环境中即可。

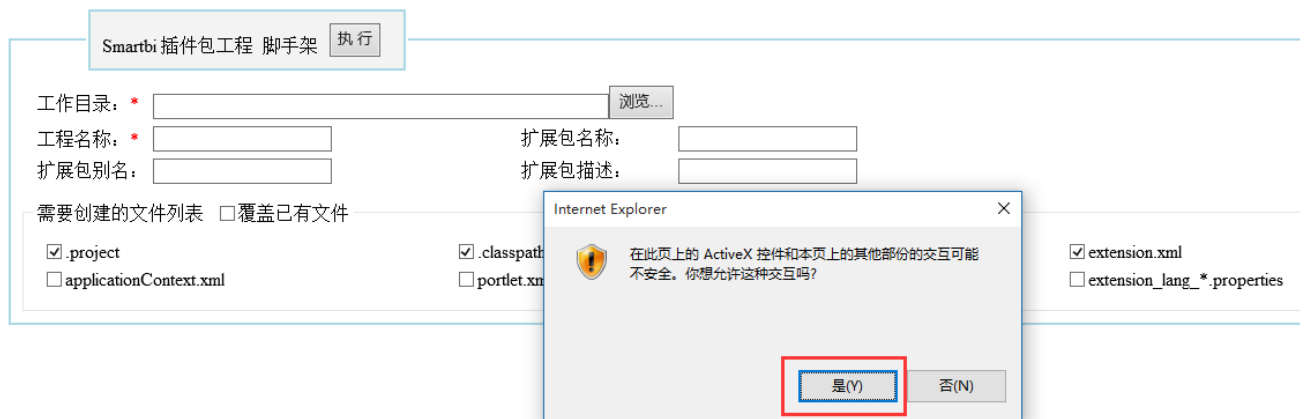
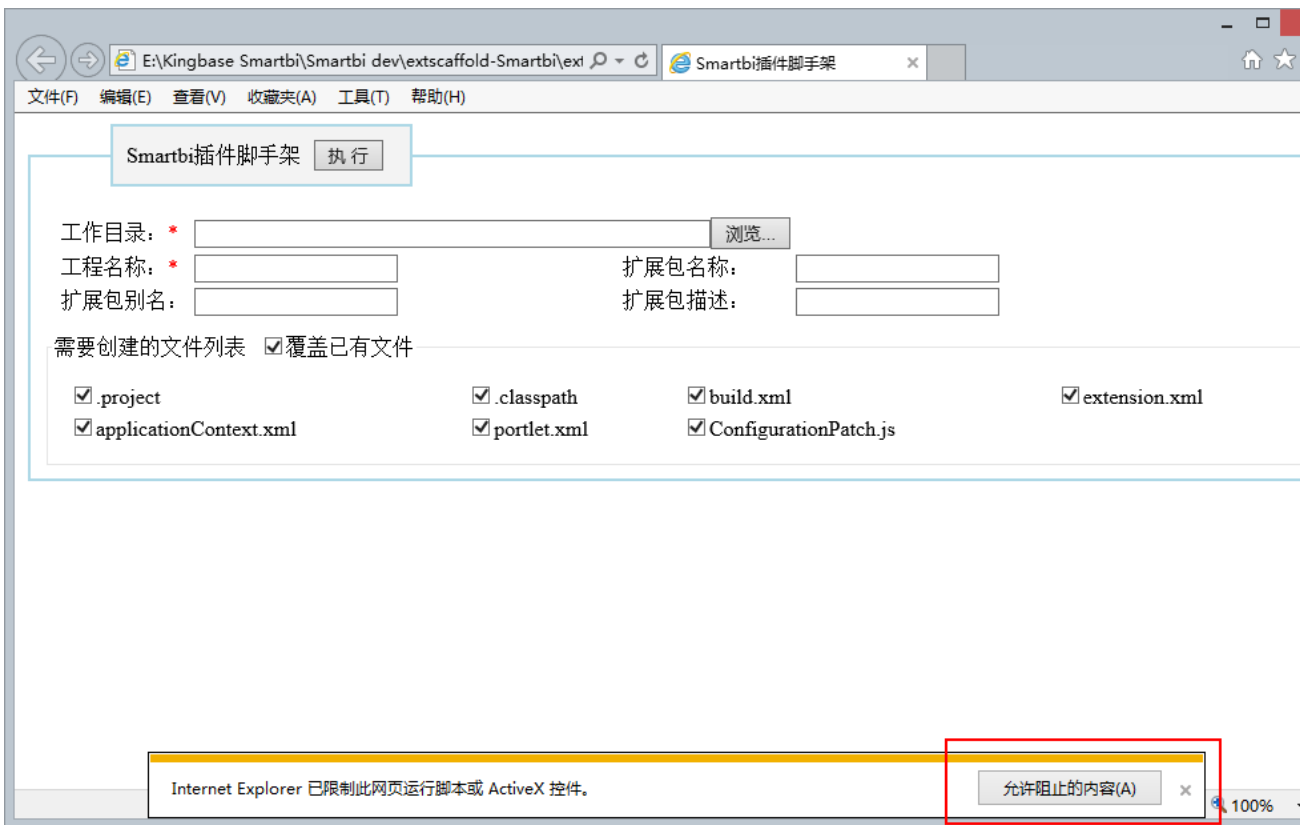
1.2 操作步骤

该框架小工具是 Smartbi 扩展包的模版项目，用来生成Smartbi扩展包的框架代码。

1、首先下载 [extscaffold-Smartbi.html](#) 或 [extscaffold-Smartbi.cmd](#) 工具，用来生成扩展包项目的框架代码。

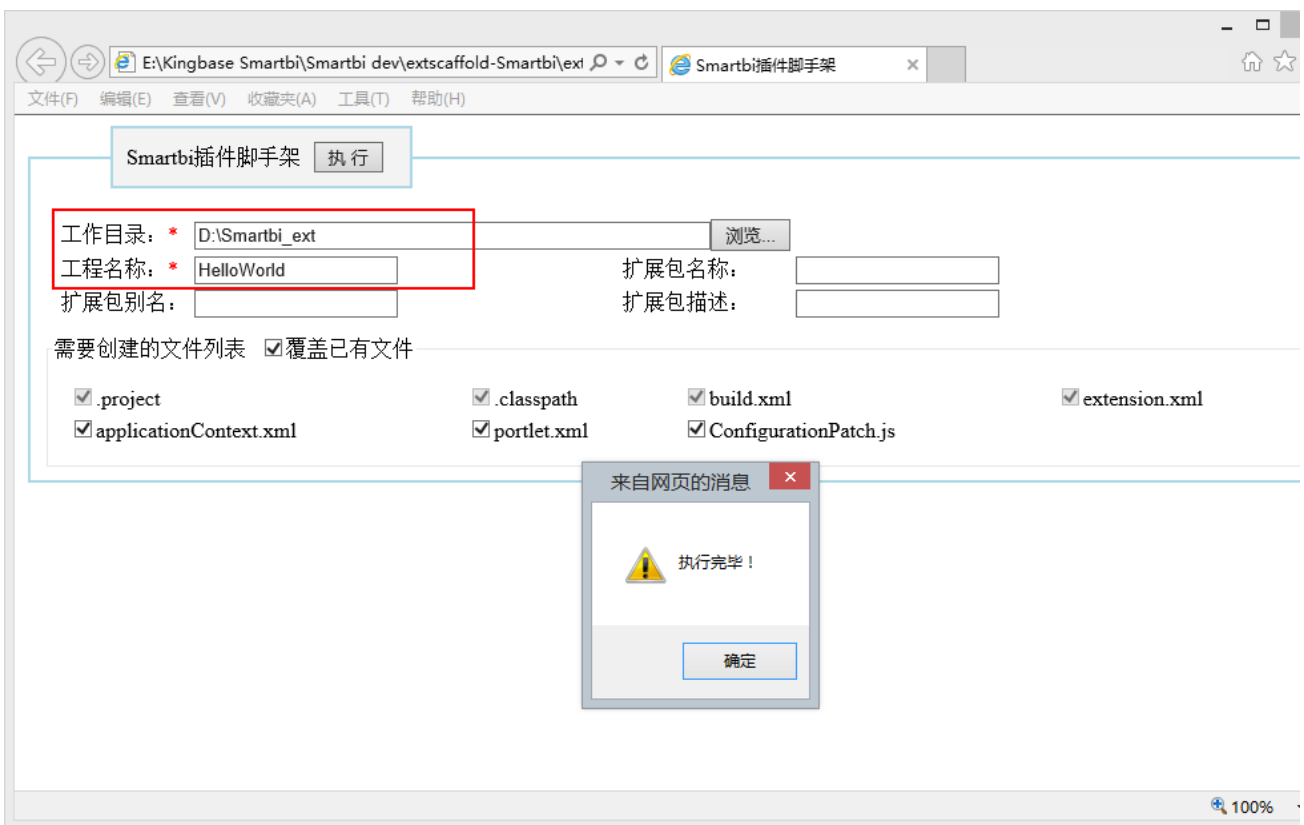
2、在IE 10或以上的机器中，[extscaffold-Smartbi.html](#) 有可能无法正常工作，则可以使用上面的 [extscaffold-Smartbi.cmd](#)，双击即可运行。

3、解压后用 [IE](#) 浏览器打开 [extscaffold-Smartbi.html](#) 文件，浏览器会在页面顶部的信息栏提示“已限制此网页运行可以访问计算机的脚本和ActiveX控件”，点击信息栏，在出现的快捷菜单上点击“允许阻止的内容”。最新版本的 IE 会在页面底部提示，直接点击“允许阻止的内容”。



注意：这个工具生成的applicationContext.xml, extension.xml等xml文件，需要另存为UTF-8的格式，否则可能报错。

4、在页面上设置“工作目录”和“工程名称”，然后点击“执行”按钮（如下图，会在你指定的“D:\Smartbi_ext\”目录下生成 HelloWorld 项目的全部框架代码）。

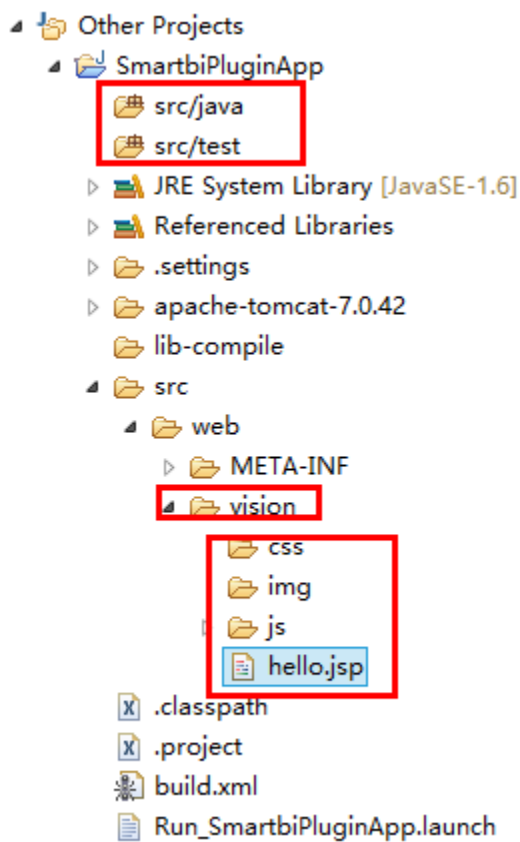


5、然后在 Eclipse 中 Import 该项目到开发环境中即可。

2 开发调试新功能

2.1 基于SmartbiPluginApp开发

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。



2.2 创建扩展包开发

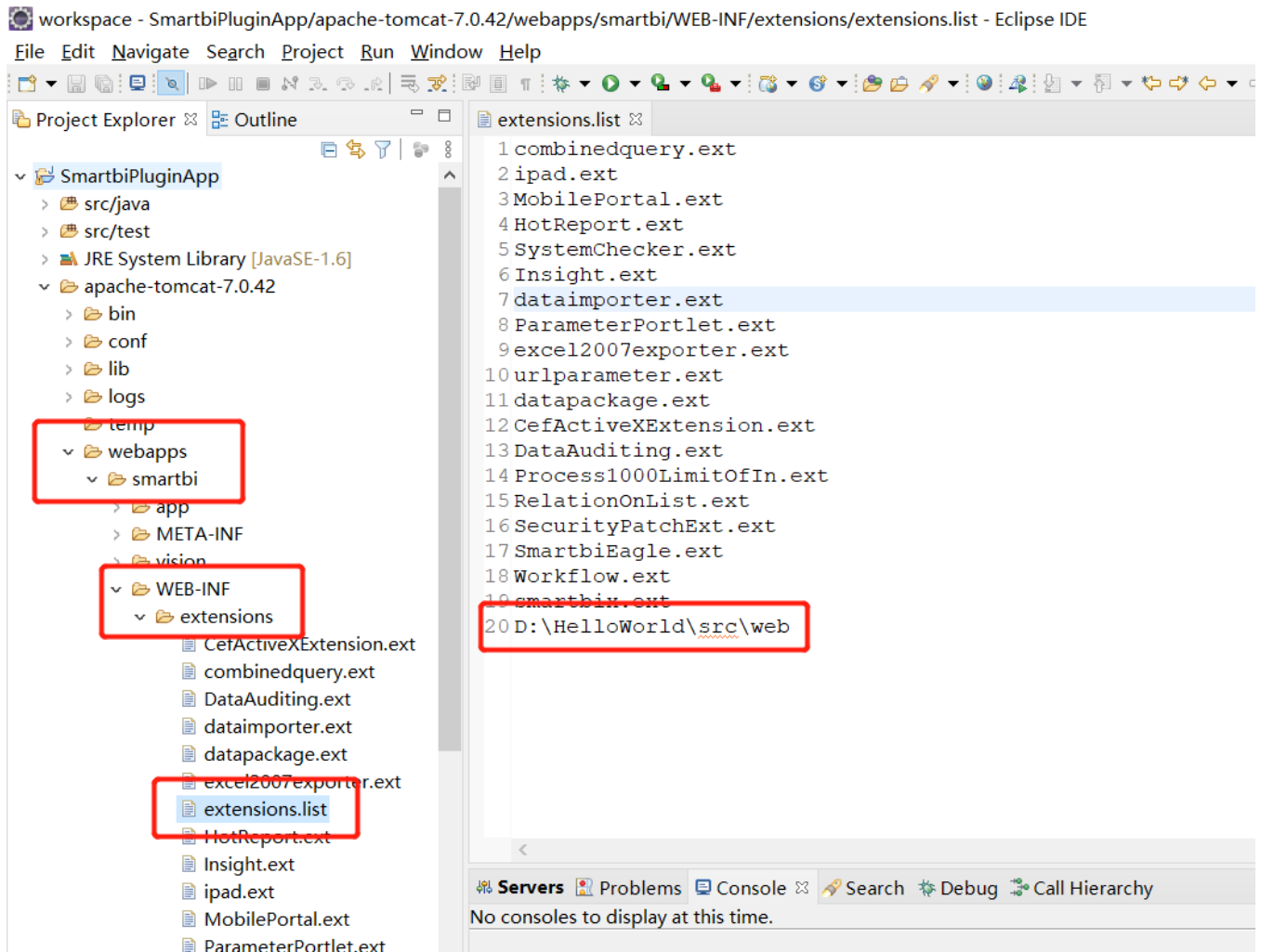
2.2.1 配置扩展包

方式一：在 Eclipse 中启动 Run_SmartbiPluginApp，打开浏览器访问 config.jsp 页面，配置 smartbi 扩展包加载路径指向创建的扩展包目录（如：/HelloWorld/src/web），该方式具体可见：[部署扩展包](#)。

方式二：通过下面方式添加扩展包路径到smartbi项目

1) 环境修改：/SmartbiPluginApp/apache-tomcat-7.0.42/webapps/smartbi/WEB-INF/extensions/extensions.list 文件，以便让 smartbi 加载正在开发的代码。请确认该文件中是否存在下面这一行，如果没有则添加

2) 将扩展包路径（例如：D:\HelloWorld\src\web），添加到extensions.list，如下图：

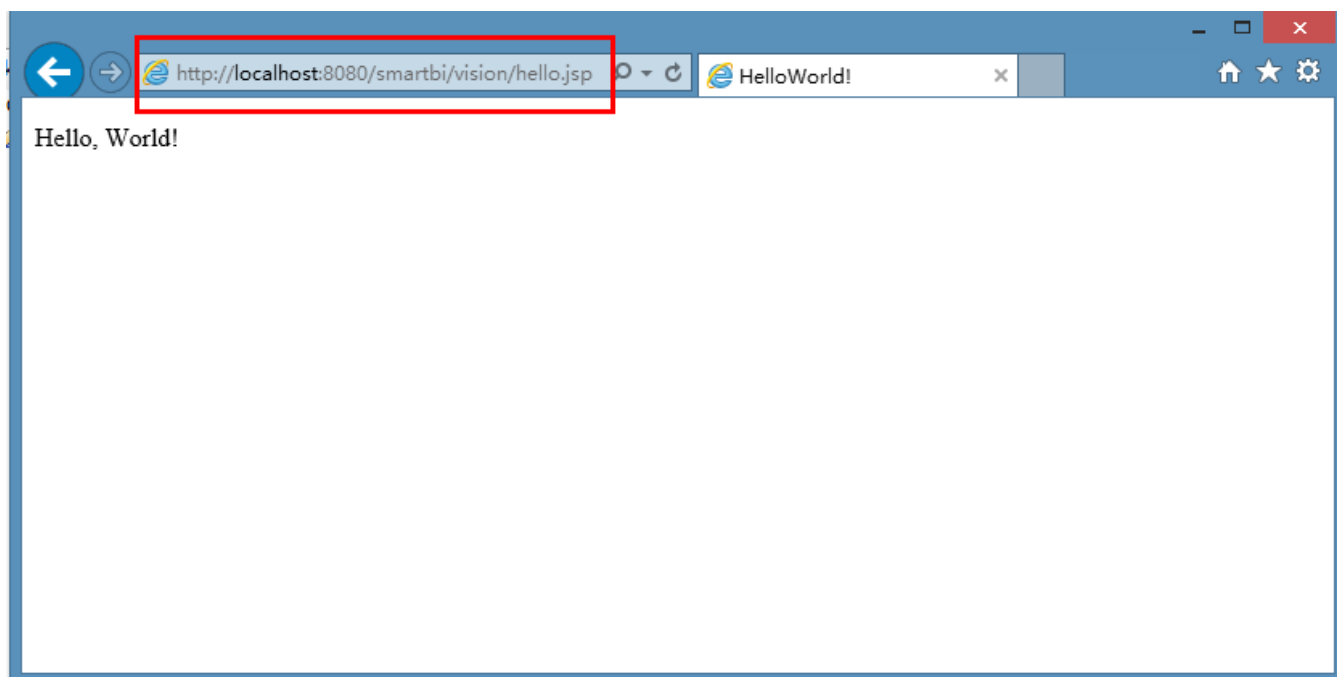


2.2.2 断点调试

若需要则在扩展包代码中设置断点，若是js文件可在需要的地方写`debugger`语句调试。

2.2.3 启动Smartbi应用

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：



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

- 1、使用上文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。
- 4、重新启动服务器，这样这个新扩展包也加入到smartbi 中了，并且可以调试它。

3 打包扩展包

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。

Package Explorer

- Other Projects
- SmartbiPluginApp
 - src/java
 - src/test
 - JRE System Library [JavaSE-1.6]
 - Referenced Libraries
 - .settings
 - apache-tomcat-7.0.42
 - lib-compile
 - src
 - .classpath
 - .project
 - build.xml
 - Run_SmartbiPluginApp.launch

smartbi-config.xml

```
49 <echo file="${basedir}/src/web/META-INF/version.txt" message="${today}" />
50 <jar destfile="${basedir}/dist/${ext_name}.ext" duplicate="preserve">
51 <fileset dir="${basedir}/src/web">
52 <exclude name="**/.cvsignore" />
53 </fileset>
54 </jar>
55 <!-- <delete file="${basedir}/src/web/META-INF/version.txt"/> -->
56 </target>
57
58 <target name="dist" depends="init">
59 <echo>=====</echo>
60 <echo>${ant.project.name}:
61 <echo>=====</echo>
62 <antcall target="compile" />
63 <antcall target="jar" />
64 </target>
65 </project>
```

Outline

- SmartbiPluginApp
 - ext_name
 - base.classpath
 - clean
 - init
 - compile
 - jar
 - dist (default)

Run As

- Run As
- Debug As
- Profile As
- Validate
- Open Javadoc Wizard...
- Team
- Compare With
- Replace With
- Remove from Context

Console

```
<terminated> SmartbiPluginApp build.xml [Ant Build] D:\support\SmartbiPluginIDE\1_JDK\bin\javaw.exe (2014-3-1)
[echo] SmartbiPluginApp: dist
[echo]
compile:
[echo]
[echo] SmartbiPluginApp: compile
[echo]
jar:
[jar] Building jar: D:\support\SmartbiPluginIDE\3_Workspace\SmartbiPluginApp\dist\smartbipluginapp.ext
BUILD SUCCESSFUL
Total time: 302 milliseconds
```