

# 扩展包开发入门示例

- 1 搭建开发环境
- 2 示例需求
  - 3 操作步骤
    - 3.1 创建插件包
    - 3.2 添加右键菜单扩展点
    - 3.3 创建 HelloPopupMenuHandler.js
    - 3.4 实现 HelloPopupMenuHandler.js
    - 3.5 功能测试和调试
    - 3.6 打包和部署插件包
    - 3.7 示例参考

## 1 搭建开发环境

工欲善其事必先利其器，做开发首先要有个好用的开发工具，并且配置好开发环境（开发环境搭建参考：[扩展包开发环境部署](#)）。

## 2 示例需求

**通过接口、扩展点插入右键菜单：**在Smartbi“分析展现”界面资源树的右键菜单中添加“Hello 163”菜单项，点击“Hello 163”菜单弹出163网站页面。



## 3 操作步骤

### 3.1 创建插件包

请参考wiki文档 [扩展包开发](#) 创建一个名为HelloDemo的插件包并导入到开发环境中。

### 3.2 添加右键菜单扩展点

修改插件包中的 ConfigurationPatch.js 声明文件，添加资源树右键菜单的声明。如下图所示，对 ConfigurationPatch.js 声明文件做修改，声明了HelloPopupMenuHandler 这个扩展点。

```

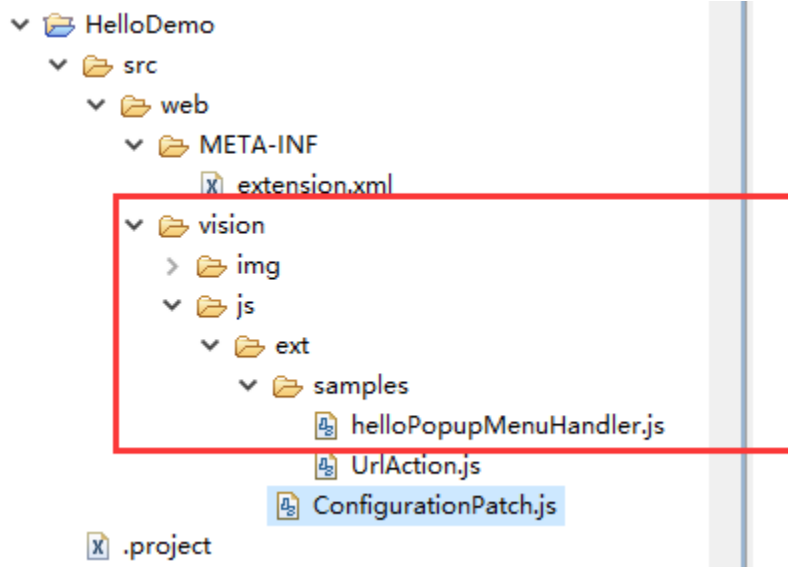
var ConfigurationPatch = {
  extensionPoints : {
    SuperviseTree: {/**定制管理的目录树*/
      handlers: [{
        className : 'ext.samples.HelloPopupMenuHandler' /** 绑定右键菜单事件对象 */
      }]
    }
  }
}

```

HelloPopuMenuHandler文件所在的目录结构，类似全类名

### 3.3 创建HelloPopupMenuHandler.js

根据上图声明，在web目录下创建：vision/js/ext/samples目录，然后创建HelloPopupMenuHandler.js 文件；配置文件中的路径是相对 vision/js 目录的。



### 3.4 实现 HelloPopupMenuHandler.js

核心代码如下：

```

2 * 构造函数
4 var HelloPopupMenuHandler = function(popupMenu) {
5     this.popupMenu = popupMenu;
6 };
7
8 /*
9  * 析构函数
10 */
11 HelloPopupMenuHandler.prototype.destroy = function() {
12 }
13
14 /*
15  * 初始化菜单，只会调用一次。一般在这里创建菜单项
16 */
17 HelloPopupMenuHandler.prototype.initMenu = function() {
18     //创建菜单项: 参数为【显示名称】、【功能编号（传递到doCmd方法中）】、【快捷键】、【图标】、【插入在哪个菜单项之前】
19     this.popupMenu.helloMenu = this.popupMenu.createMenuItem("hello 163", "HelloCommand", "H", null, "刷新");
20 };
21
22 /*
23  * 清空菜单状态，每次弹出前调用。一般应在这里隐藏本Handler所创建的所有菜单
24 */
25 HelloPopupMenuHandler.prototype.clearMenuState = function() {
26     this.popupMenu.helloMenu.setVisibility(false);
27 }
28

```

### 3.5 功能测试和调试

启动应用后即可查看修改效果，如果是客户端的 JS 代码，可以在其中添加 debugger 断点，通过浏览器调试工具可以单步调试。

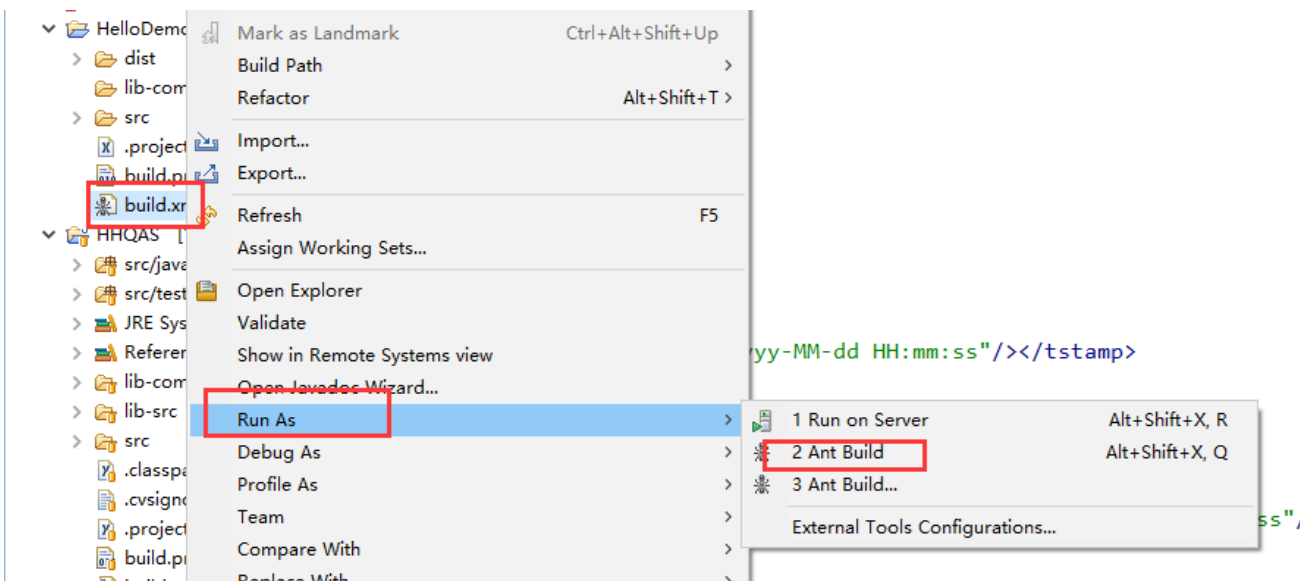
```

1
2 /** 把源js文件的方法从新命名，以便重用 */
3 SuperviseDefaultCommand.prototype.getItems_ext = SuperviseDefaultCommand.
4
5 /**
6  *
7  */
8 SuperviseDefaultCommand.prototype.getItems = function() {
9     debugger;
10     .....
11
12     具体实现什么功能
13
14
15
16
17
18
19
20 }

```

### 3.6 打包和部署插件包

功能开发及测试完成后，直接用 Ant 工具打包，会在 dist 目录下生成 HelloDemo.ext 插件包，参考“[插件包部署](#)”，把它部署到任意 Smartbi 服务器上。



### 3.7 示例参考

可下载插件包示例 [HelloDemo.rar](#) 。这是一个简单的应用开发，它演示了“如何在资源树的右键菜单添加第三方的功能菜单”的开发过程。