

# 自定义logo插件包

- 1. 需求背景说明
- 2. 功能说明
- 3. 操作说明
  - 3.1 部署开发插件包
  - 3.2 修改扩展包内容
  - 3.3 替换整个登录页面
  - 3.4 替换登陆页logo
  - 3.5 替换导航栏logo
  - 3.6 替换导航栏显示的样式
  - 3.7 修改关于页面信息
    - 3.7.1 修改关于页logo
    - 3.7.2 修改关于页文字信息
    - 3.7.3 删除关于页中的TAB页
  - 3.8 屏蔽右上角图标及下拉菜单
- 4. 注意事项

## 1. 需求背景说明

在客户的生产环境中部署 Smartbi 程序时，通常需要对其实做一些简单的客户化定制，比如替换登录界面、以及主界面顶部的LOGO和图片、或者修改程序名称标识等。

## 2. 功能说明

该插件包的主要功能有：替换登陆页整页内容；替换登陆页logo；替换导航栏logo；替换导航栏显示的样式；替换关于弹窗logo。

## 3. 操作说明

### 3.1 部署开发插件包

部署 [ChangeLogoV9.ext](#)，详细说明请参考Wiki链接：[部署扩展包](#)

### 3.2 修改扩展包内容

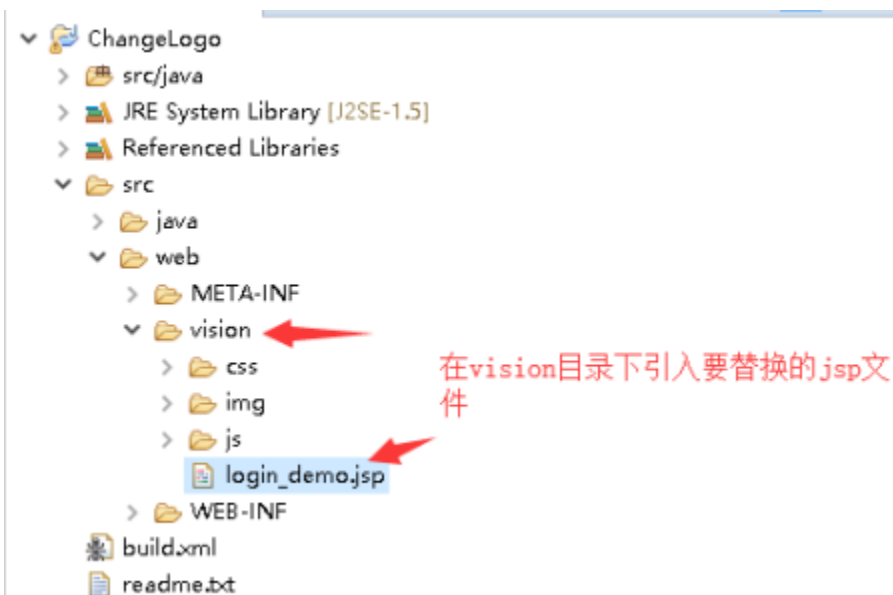
#### 1、修改插件包中某个文件

- (1) 首先用 WinRAR 之类的压缩工具，打开 ChangeLogo.ext 文件。
- (2) 然后将需要修改的文件从 WinRAR 窗口中拖出来，对其进行修改。
- (3) 接着将修改后的文件，拖放回 WinRAR 窗口中，覆盖以前的旧文件。
- (4) 关闭 WinRAR 窗口保存修改即可。

### 3.3 替换整个登录页面

- 引入替换的页面

将替换的页面引入到插件包的vision文件夹下



#### • 修改配置文件

修改插件包/ChangeLogo/src/web/vision/js/ext/路径下的ConfigurationPatch.js文件，配置项说明如下：

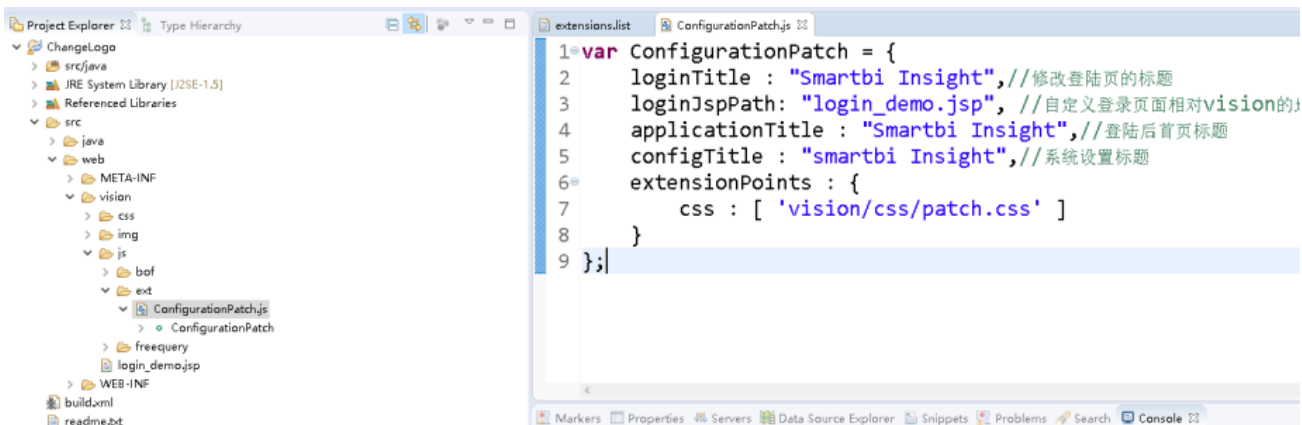
**loginTitle**——修改登陆页的标题

**loginJspPath**——自定义登录页面相对vision的地址

**applicationTitle**——登陆后首页标题

**configTitle**——系统设置标题

**extensionPoints**——扩展的资源信息，当前插件包可以用来引入样式信息



#### 注意点：

1) 若是引入的登陆页仍需要走smartbi原有的验证逻辑，需要在引入的登陆页中加入以下smartbi登陆验证的代码逻辑。

## Smartbi登录验证逻辑

```
var jsloader = new JSLoader({
    lang: '<%=smartbi.Bootstrap.getLocaleName(request)%>'
});
var util = jsloader.resolve("freequery.common.util");
document.onkeydown = keydown;

function init() {
    var userName = document.getElementById("user");
    if (userName)
        userName.focus();
}

function login() {
    var userName = document.getElementById("user"); //IDHTMLID
    var password = document.getElementById("password"); //IDHTMLID
    var ret = util.remoteInvokeEx("UserService", "login", [userName.value, password.value]); //Smartbi

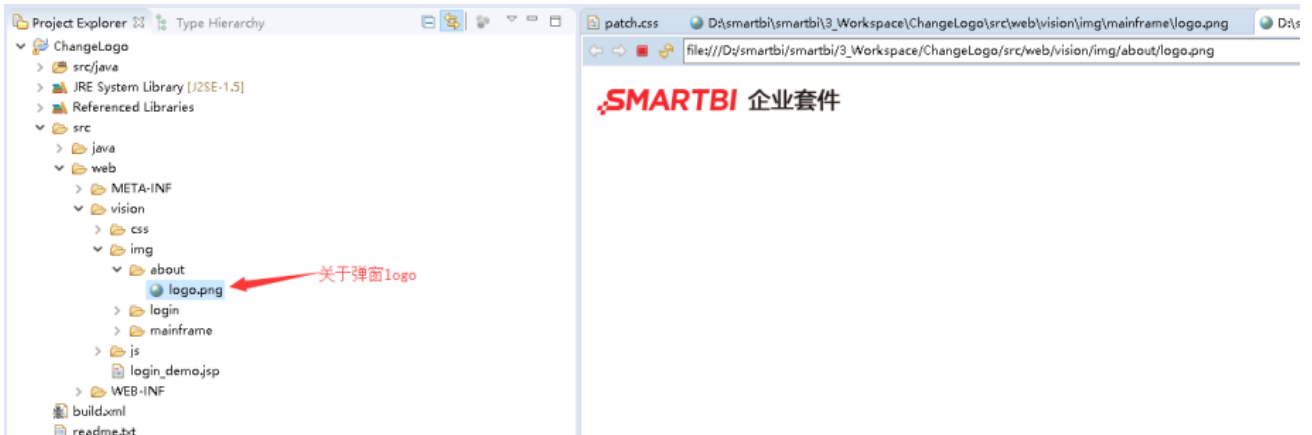
    if (!ret || !ret.result) {
        alert("!");
        return;
    } else {
        var url = window.location + '';
        var p = url.indexOf('login_demo.jsp');
        if (p > -1) {
            window.location = url.substring(0, p);
        } else {
            window.location.reload();
        }
    }
}

function keydown(e) {
    var theEvent = e || window.event;
    var code = theEvent.keyCode || theEvent.which || theEvent.charCode;
    if (code == 13) {
        login();
    }
}
```

2) 系统统一访问入口仍然为 <http://localhost:18080/smartbi/vision/index.jsp>；访问 Smartbi 时如果用户未登录，会自动跳转到 login\_demo.jsp 页面中，此登录界面的处理逻辑为：点击“登录”按钮时，调用 smartbi 的登录API进行登录，再重新刷新本页面

### 3.4 替换登陆页logo

将设计好的logo覆盖 /ChangeLogo/src/web/vision/img/login路径下的logo.png文件，即可替换原来的logo样式

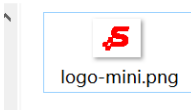




### 3.5 替换导航栏logo

将设计好的logo覆盖 /ChangeLogo/src/web/vision/img/baseframe路径下的logo-mini.png文件，即可替换原来的logo样式

示例库 > ChangeLogo > ChangeLogo\_修改登陆界面和LOGO > ChangeLogo-V856 > ChangeLogo-V856 > ChangeLogo > vision > img > baseframe



### 3.6 替换导航栏显示的样式

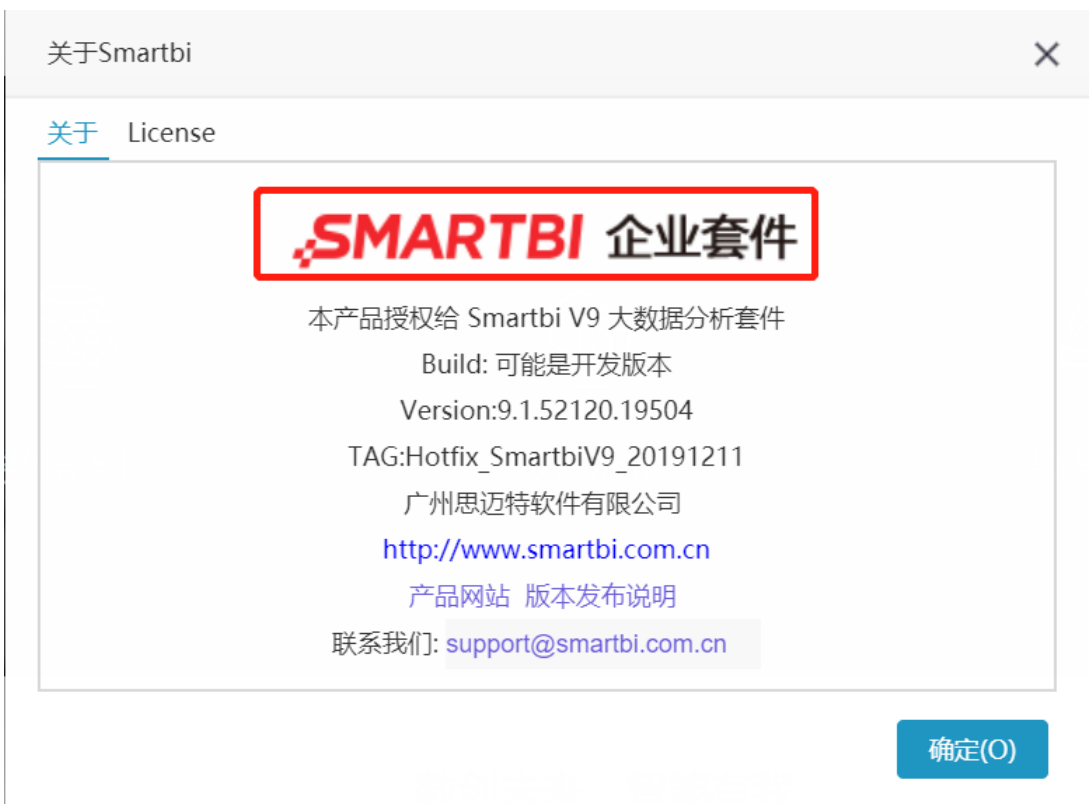
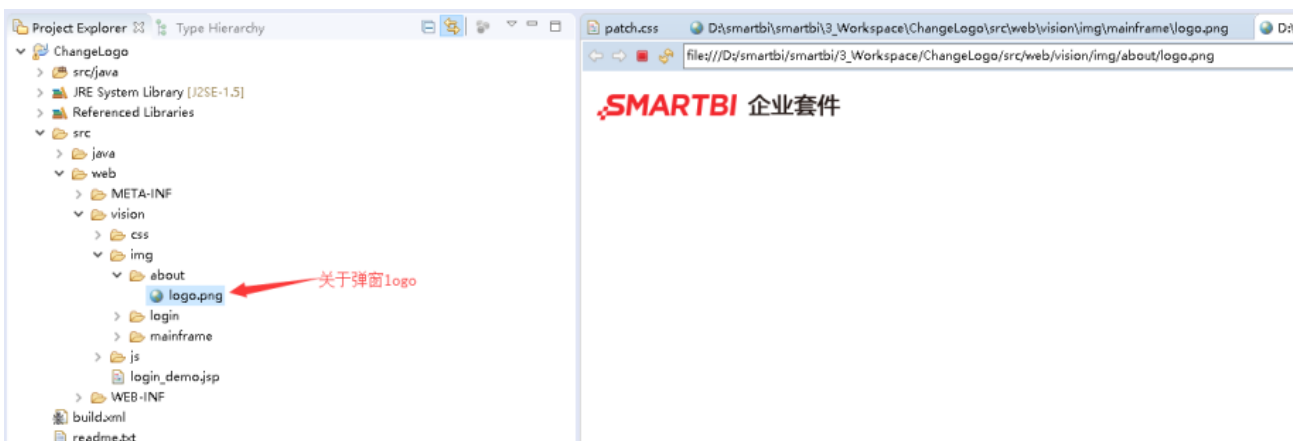
由于在2.2.2修改配置文件中，引入了css样式的文件的路径为/ChangeLogo/src/web/vision/css/patch.css，因此找到/ChangeLogo/src/web/vision/css下的patch.css文件，修改文件中.mainTopTd的样式。具体修改内容参考css文档的详解<http://css.doyoe.com/>



### 3.7 修改关于页面信息

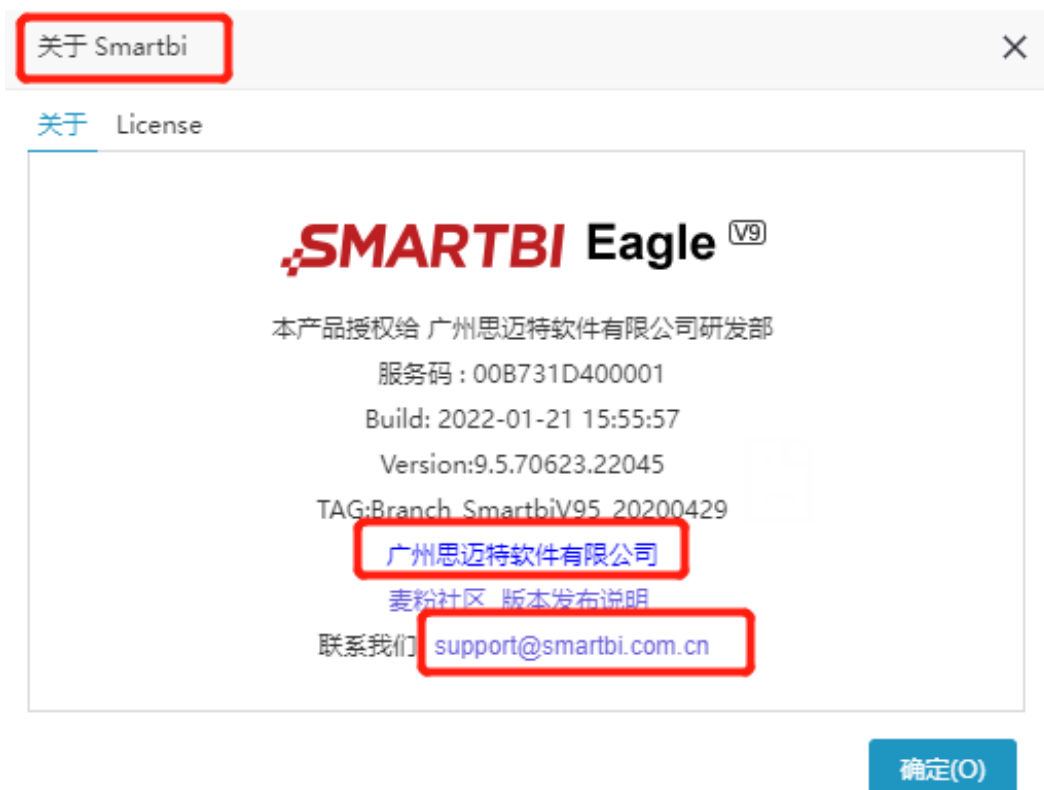
#### 3.7.1 修改关于页logo

将设计好的logo覆盖 /ChangeLogo/src/web/vision/img/about 路径下的logo.png文件，即可替换原来的logo样式



### 3.7.2 修改关于页文字信息

弹窗标题、公司名称、网址和邮箱等信息，修改对应文件：ChangeLogo.ext\vision\js\bof\baseajax\common\BannerView.js.patch



```
BannerView.prototype.showAbout = function() {  
    var data = [ registry.get('CompanyName'), registry.get('WebAddress'),  
                registry.get('MailAddr'), window ];  
    var dialogConfig = {  
        title : '${About}',  
        size : DialogFactory.getInstance().size.MIDDLE,  
        fullName : 'freequery.main.AboutDialog'  
    };  
    DialogFactory.getInstance().showDialog(dialogConfig, data);  
};
```

公司名称、网站、  
邮箱

弹框标题

- 上图data可参考下面代码修改：

```
var data = [ "公司名称", "公司网址", "mailto:公司邮箱", window ];
```

### 3.7.3 删除关于页中的TAB页

修改对应文件：ChangeLogo.ext\vision\js\freequery\main\AboutDialog.js.patch，屏蔽对应代码，则会删除对应的tab页（注意不可同时屏蔽两个tab），修改如下图所示：

- 修改前

```

AboutDialog.prototype.initTab = function() {
    this.initLicenseInfo(this.elemLicenseInfoDiv);
    this.pagecontrol = new PagePanel(this.elemTabPanel);
    this.pagecontrol.contentBanner.style.backgroundColor = '#fff';

    this.aboutTab = this.pagecontrol.appendTab();
    this.aboutTab.setCaption("${About}");
    this.aboutTab.appendItem(this.elemAboutDiv);
    if (this.aboutTab.itemParent) {
        this.aboutTab.itemParent.style.verticalAlign = 'middle';
    }
    this.elemAboutDiv.style.display='none';

    this.licenseInfoTab = this.pagecontrol.appendTab();
    this.licenseInfoTab.setCaption("License");
    this.licenseInfoTab.appendItem(this.elemLicenseInfoDiv);
    this.elemLicenseInfoDiv.style.display='none';

    this.pagecontrol.tabs[0].setActive();
    this.pagecontrol.autoFitResize();
}

```

关于

license

- 屏蔽license页

```

AboutDialog.prototype.initTab = function() {
    this.initLicenseInfo(this.elemLicenseInfoDiv);
    this.pagecontrol = new PagePanel(this.elemTabPanel);
    this.pagecontrol.contentBanner.style.backgroundColor = '#fff';

    this.aboutTab = this.pagecontrol.appendTab();
    this.aboutTab.setCaption("${About}");
    this.aboutTab.appendItem(this.elemAboutDiv);
    if (this.aboutTab.itemParent) {
        this.aboutTab.itemParent.style.verticalAlign = 'middle';
    }
    //this.elemAboutDiv.style.display='none';

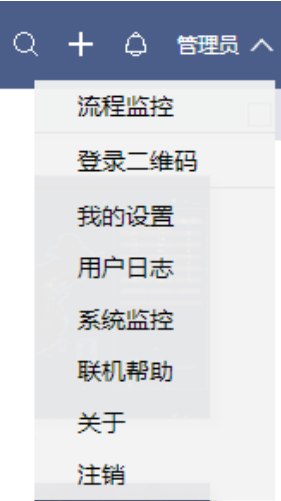
    // this.licenseInfoTab = this.pagecontrol.appendTab();
    // this.licenseInfoTab.setCaption("License");
    // this.licenseInfoTab.appendItem(this.elemLicenseInfoDiv);
    this.elemLicenseInfoDiv.style.display='none';

    this.pagecontrol.tabs[0].setActive();
    this.pagecontrol.autoFitResize();
}

```

### 3.8 屏蔽右上角图标及下拉菜单

系统右上角的功能图标入口及用户的下拉菜单选项，用户可以根据需求进行屏蔽和释放，通过修改扩展包文件“BannerView.js.patch”（路径：ChangeLogo.ext\vision\js\bof\baseajax\common）中的代码实现。



具体修改及影响的对象如下所示：

修改内容	修改详情
搜索图标 (  )	<p>找到如下代码：</p> <pre>this.elemSearchText.style.display='none';</pre> <p>将这段代码的注释取消则屏蔽了“搜索”图标。</p>
新建图标 (  )	<p>找到如下代码：</p> <pre>this.elemCreateAnalysis.style.display='none';</pre> <p>将这段代码的注释取消则屏蔽了“新建”图标。</p>
我的任务图标 (  )	<p>添加如下代码：</p> <pre>this.bannertask.style.display='none';</pre> <p>添加位置如下图所示：</p> <pre>//this.elemSearchText.style.display='none';    //搜索 //this.elemCreateAnalysis.style.display='none';  //新建 //this.elemItemDropdown.style.display='none';  //管理员 this.bannertask.style.display='none'; //我的任务</pre>
管理员	<p>找到如下代码：</p> <pre>this.elemItemDropdown.style.display='none';</pre> <p>将这段代码的注释取消则屏蔽了“管理员”。</p>



管理员下拉菜单	<div>流程监控、登录二维码、用户日志、系统监控</div> <div>流程监控、登录二维码、用户日志、系统监控这四个菜单的处理方法一致，在文档最后添加如下代码：</div> <div><pre>BannerView.prototype._changeLogo_addMenu = BannerView.prototype.addMenu; BannerView.prototype.addMenu = function(text, cmd, parentElem) {   if ( cmd == "MY_INSTANCE"    cmd == "QRCode"    cmd == "USER_LOGS"    cmd == "SYS_MONITOR") {     return ;   }   this._changeLogo_addMenu(text, cmd, parentElem); };</pre></div> <div>该代码中，对象关系如下：</div> <div><ul style="list-style-type: none"><li>• MY_INSTANCE：流程监控</li><li>• QRCode：登录二维码</li><li>• USER_LOGS：用户日志</li><li>• SYS_MONITOR：系统监控</li></ul></div> <div>若想屏蔽对应的菜单，只需要在if()语句中将菜单对应的对象加上即可。如以上代码中，表示将“流程监控、登录二维码、用户日志、系统监控”都屏蔽，效果如下：</div> <div><table><tr><th>屏蔽前</th><th>屏蔽后</th></tr><tr><td></td><td></td></tr></table></div>	屏蔽前	屏蔽后		
屏蔽前	屏蔽后				
我的设置	<div>找到如下代码：</div> <div><pre>this.addMenu('\${My}\${Settings}', 'MY_SETTINGS', parentElem);</pre></div> <div>将这段代码注释即屏蔽了“我的设置”。</div>				
联机帮助	<div>找到如下代码：</div> <div><pre>this.addMenu('\${Onlinehelp}', 'ONLINE_HELP', parentElem);</pre></div> <div>将这段代码注释即屏蔽了“联机帮助”。</div>				
关于	<div>找到如下代码：</div> <div><pre>this.addMenu('\${About}', 'ABOUT_SMARTBI', parentElem);</pre></div> <div>将这段代码注释即屏蔽了“关于”。</div>				
注销	<div>找到如下代码：</div> <div><pre>this.addMenu('\${Logout}', 'LOGOUT', parentElem);</pre></div> <div>将这段代码注释即屏蔽了“注销”。</div>				

#### 4. 注意事项

- 1、修改完插件包的文件之后，需要保存
- 2、修改的位置需正确，否则会导致正常功能失效
- 3、若需要替换更多的图片，需先找到该图片所在目录，找目录请参考 [前端定位技巧](#)。