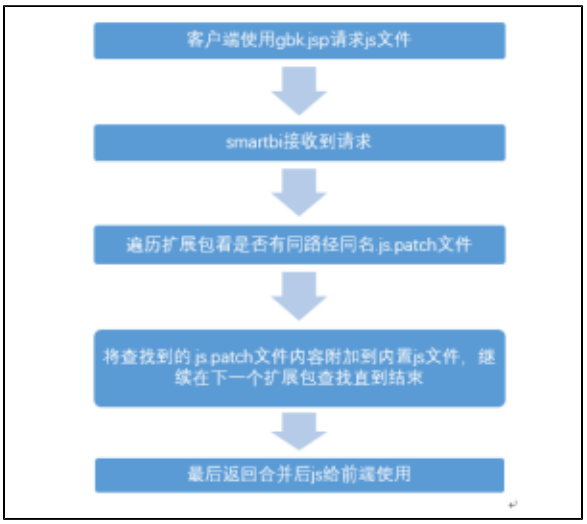
第四课：修改Smartbi JS文件

1. 说明

在插件开发过程中经常遇到要修改Smartbi原有的JavaScript文件的情况。可以通过下面两种方法去修改：

1. 直接替换JS文件；
2. 自定义Patch文件进行修改（推荐）。



2. 整个JS文件替换

当需要替换原有的JavaScript文件时，在插件中建立与该JavaScript文件一致的目录结构，在该目录中建立同名的JavaScript文件。Smartbi会优先加载插件中的JavaScript文件。请参照"替换Smartbi文件"。

3. 通过Patch文件修改

在需要修改原有JavaScript的部分功能时，在插件中建立与该JavaScript文件一致的目录结构，建立【原有JavaScript文件名】+【.patch】命名的文件，例如在在插件中建立文

件"src/web/vision/js/bof/baseajax/common/LoginView.js.patch"，该文件的作用在于声明对原有JavaScript文件的修改。

一般修改的方式会有以下几种：

1. 替换原有方法：在Patch文件中重新声明该方法的实现。
2. 在原有方法调用前或调用后增加特殊代码：在Patch文件中先保留原有方法实现并重新声明该方法实现，在新的实现中可以调用保留的方法。
3. 修改原有JavaScript对象的构造函数：该方式请谨慎使用！

> ⚠ 自定义Patch文件方式，利用同一个js文件中多个相同的同名方法，最后一个会生效的原理，smartbi在加载js脚本时，会遍历所有插件包，然后将插件包中同路径及同名的.js.patch文件内容附加到产品内置js的后面，最后返回的是合并后的js，意味着这种方式是部分修改产品js，风险相对会小，升级产品的话也只是修改的那个方法可能产生问题。
>
> 所以在一般情况下最好使用Patch文件修改方式，这样可以降低将来Smartbi升级造成的风险。

3.1替换原有方法

在Patch文件中重新声明该方法的实现。

```
LoginView.prototype.elemPassword_focus_handler = function(ev) {
        alert("Before Password Select");
        this.elemPassword.select();
};
```

3.2原有方法调用前或调用后增加特殊代码

在Patch文件中先保留原有方法实现并重新声明该方法实现，在新的实现中可以调用保留的方法。

```
//
LoginView.prototype.__elemLogin_click_handler = LoginView.prototype.elemLogin_click_handler;
LoginView.prototype.elemLogin_click_handler = function(ev) {
        alert('Before login');
        //
        this.__elemLogin_click_handler(ev);
        alert('After login');
}
```

3.3修改原有JavaScript对象的构造函数(该方式请谨慎使用！)

```
// JavaScript
var LoginViewPatch = function(container) {
        alert('Before constructor');
        //
        LoginViewPatch.superclass.constructor.call(this, container);
        alert('After constructor');
}
// JavaScript
lang.patch(LoginViewPatch, LoginView);
```

4.示例工程

1、示例工程代码 ModifyJSFile.rar

2、效果如下：