开发standard-lesson项目中获取系统信息页面即SystemInfo页面。该页面在com.bstek.dorado.sample.standardlessn.junior.system目录下。

需求：

1）点击[获取Dorado版本信息]按钮后对后台进行ajax请求，通过按钮下方的四个标签展示返回信息。

2）点击[清空]按钮，清空标签信息。

效果如下：



## 1、准备工作

由于有前后台交互功能，需要在Spring上下文中注册一个用于提供服务的bean，对于这个bean使用Spring提供的@Component注解，如果需要使用该注解，则要在项目中app-content.xml文件中增加如下配置：

<context:component-scan base-package="com.bstek.dorado.sample" />

这样系统默认就会自动扫描com.bstek.dorado.sample及其子包中配置有@Component注解的类，并将其注册到Spring的BeanFactory中。

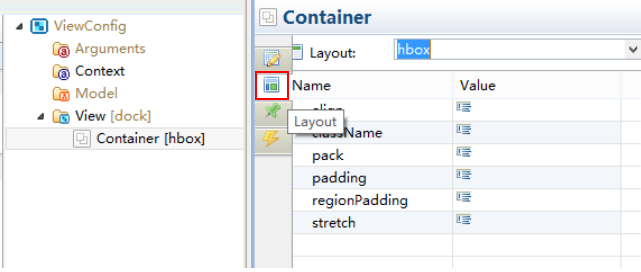
## 2、开发步骤

### 2.1创建视图并添加控件

1）新建SystemInfo.view.xml文件

2）添加并配置Container控件

将右侧工作区切换到Layout属性设置区中，如下图所示：



将Container控件的Layout属性设置为hbox（hbox为横向布局即加入的子控件将依次横向排列）。

注：Container是一个容器控件，作为容器型的控件，我们可以在内部加入多个子控件。控件之间的布局关系由Container本身的Layout特性决定。

3）在容器中添加并配置Button控件

向Container中添加两个按钮：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| id | buttonGetSysInfo | 控件id |
| caption | 获取版本信息 | 按钮显示名称 |

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| id | buttonClear | 控件id |
| caption | 清除 | 按钮显示名称 |

4）再添加一个Container控件

将其Layout设置为vbox（vbox为纵向布局即加入的子控件将依次纵向排列）。

5）向第二个Container中添加4个Label控件

设置属性：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| id | labelProduct | 控件id |
| text | 产品： | 控件显示的文本 |

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| id | labelVendor | 控件id |
| text | 供应商： | 控件显示的文本 |

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| id | labelVersion | 控件id |
| caption | 版本： | 控件显示的文本 |

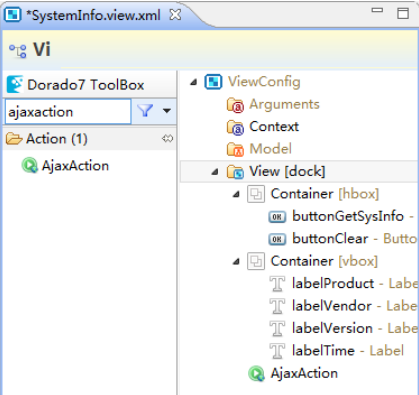
|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| id | labelTime | 控件id |
| caption | 当前系统时间： | 控件显示的文本 |

6）设置四个标签的布局，使之美观

设置labelProduct、labelVender、labelVersion、labelTime的padding属性为10。

7）添加AjaxAction控件，用于前后台交互

在[View]节点下添加AjaxAction控件：



AjaxAction是用于执行后台方法的动作控件。设置AjaxAction的属性如下：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| id | ajaxactionGetSysInfo | 控件id |
| service | systemInfoService#getSystemInfo | 用于调用后台服务，systemInfoService为注册到beanFactory中的beanId，getSystemInfo为方法名 |
| executingMessage | 正在获取Dorado版本信息... | ajaxaction执行中提示信息 |
| successMessage | 获取Dorado版本信息成功 | ajaxaction执行成功提示信息 |

设置Button的action属性表示，点击Button按钮就触发AjaxAction动作，执行后台方法。

### 2.2准备后台服务

1）新建com.bstek.dorado.sample.standardlesson.service包，用于集中放置项目中需要的service。

2）新建SystemInfoService.java

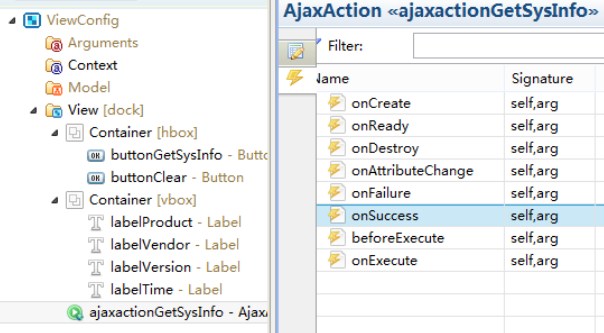


注：@Expose是Dorado7专门提供的注解，用于定义可暴露服务，根据这个规则Dorado7会将这个方法自动注册在ExposedServiceManager中，其中服务名为：systemInfoService，服务表达式为：systemInfoService#getSystemInfo。服务名就是SystemInfoService.java在BeanFactory中注册的bean的id，annotation机制会自动将Ajax的S自动转换为小写字母。

### 2.3给控件绑定事件

AjaxAction执行成功后，应该获取后台方法的返回值并对前台控件赋值，因此需要对ajaxactionGetSysInfo控件绑定onSuccess事件。

1）选中ajaxactionGetSysInfo控件，在右侧工作区中，点击第二个tab页，切换到ClientEvents事件列表：



2）双击选择onSuccess事件，主工作区将由下方的Properties切换到Events编辑模式，在事件编辑区输入代码：

var info = self.get("returnValue");

view.set("#labelProduct.text","产品：" + info.product);

view.set("#labelVendor.text","提供商：" + info.vendor);

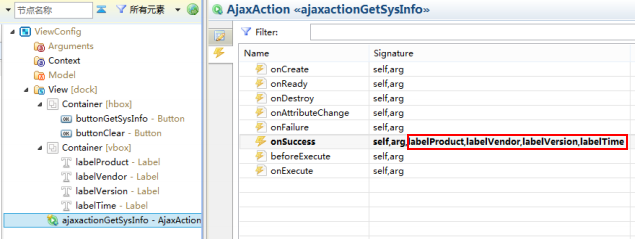
view.set("#labelVersion.text","版本：" + info.version);

view.set("#labelTime.text","当前系统时间：" + info.time);

注：self表示ajaxactionGetSysInfo这个控件自身，returnValue为AjaxAction控件的属性字段，表示Ajax调用结束后Java层返回的结果。

#labelProduct表示当前页面的labelProduct控件。

也可以将labelProduct，labelVendor，labelVersion，labelTime这四个控件的id作为参数传递到onSuccess方法中，然后在时间中通过控件Id直接引用控件对象。



//优化后onSuccess事件的代码

var info = self.get("returnValue");

labelProduct.set("text","产品：" + info.product);

labelVendor.set("text","提供商：" + info.vendor);

labelVersion.set("text","版本：" + info.version);

labelTime.set("text","当前系统时间：" + info.time);

3）点击[清空]按钮时，清空页面4个标签显示的信息。鼠标选中buttonClear控件，给按钮绑定事件onClick事件，将labelProduct，labelVendor，labelVersion，labelTime这四个控件的Id作为参数传递到onClick事件，通过控件Id直接引用控件对象：

labelProduct.set("text","产品：...");

labelVendor.set("text","提供商：...");

labelVersion.set("text","版本：...");

labelTime.set("text","当前系统时间：...");

### 2.4查看运行效果



### 2.5利用Chrome调试工具

通常在项目中我们对js脚本进行调试有以下2种方式：

1）alert调试法（不推荐）

首先是最原始也是最简单的使用alert，在页面中需要输出的变量的地方加上alert函数，将变量弹出显示：

var info = self.get("returnValue");

//alert("product=" + info.product);

labelProduct.set("text","产品：" + info.product);

labelVendor.set("text","提供商：" + info.vendor);

labelVersion.set("text","版本：" + info.version);

labelTime.set("text","当前系统时间：" + info.time);

alert方式虽然简单，但是调试繁琐，加入需要监视变量过多的话，会弹出大量的弹出框，用户体验相当不好。

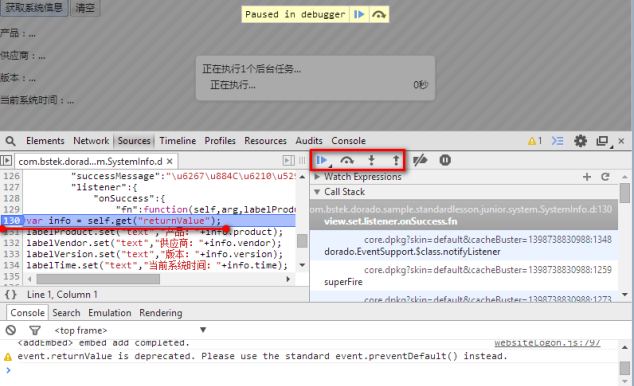
2）浏览器调试工具

使用浏览器的调试工具，Chrome、IE以及FireFox都自带浏览器调试工具，特别是FireFox还有调试插件FireBug，下面以Chrome浏览器的调试工具为例对js进行调试。

打开浏览器输入访问的页面：

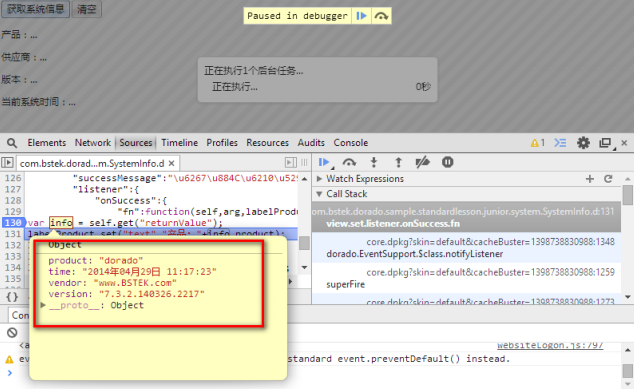
F12进入调试界面

切换到[Sources]页，在130行用[鼠标左键]打上断点，然后点击页面中的[获取系统信息]按钮后，页面执行到断点处暂停，如下图所示：

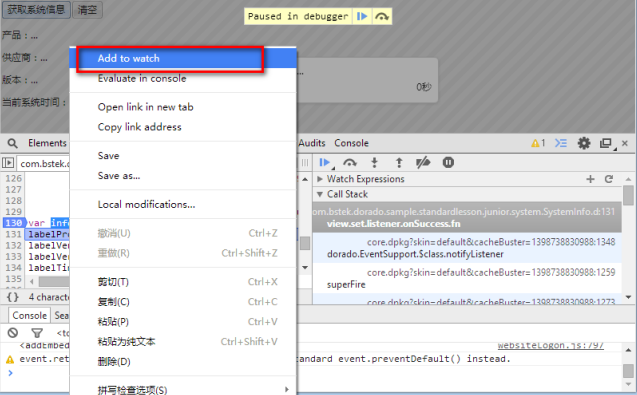


可以选择使用快捷键[F8]继续执行js代码或者跳到下一个断点处，也可以使用快捷键[F10]逐过程（即跳过该语句中的方法、表达式等），快捷键[F11]逐语句测试（即单步测试，会跳入方法、表达式，进行逐句的跟踪调试。）。

点击[F10]，执行下一句js代码，鼠标双击选中info并悬停片刻，可以看到info中包含的变量：



可以看到3个变量，如果需要添加监视，点击鼠标右键在弹出菜中选择[Add to Watch]，添加监视：



在右侧是[Watch Expressions]监视区中可以监视res中的变量。

（强烈推荐）Debugger关键字

方便快捷，不需要在View输出的页面寻找需要调试的js，只要在需要调试的js中加入debugger关键字。代码如下：

var info = self.get("returnValue");

//需要js执行停止处

debugger;

labelProduct.set("text","产品："+info.product);

labelVendor.set("text","提供商："+info.vendor);

labelVersion.set("text","版本："+info.version);

labelTime.set("text","当前系统时间："+info.time);

此调试方法的好处在于不用在页面上打断点，且更加方便快捷。

## 3、小结

* Container容器控件的使用
* Hbox与Vbox布局管理器的使用
* AjaxAction前后台事件响应。