ExtJs自学之路

### 前言

* Web程序的革命
* CGI——基于传统的服务器端模式（可视化界面）
* Java Applet——基于本地虚拟机的运行模式（小程序）
* Flash——基于浏览器插件的运行模式
* Web Start——在线安装运行模式
* Ajax——基于浏览器的运行模式
* RIA——基于Flex、Sliverlight桌面程序系统

### 选择AJAX理由

AJAX虽然不是最新的WEB开发技术，但是其实最后一个支持浏览器的WEB开发技术，因为无任Sliverlight，还是Flex其原理已经不再需要浏览器支持了，而浏览器技术发展到现在，不可能在短期放弃AJAX大都是用Javascript，JavaScript在已经十年发展，已经拥有大量的应用队伍，其继续发展的可能性是肯定存在的

### EXTJS属于Ajax技术

* 基于JavaScript语言
* 基于Java Swing的MVC架构
* 支持组件化、模块化设计
* 提供“本地数据源”的支持
* 完善与服务端的交互机制
* 是最有可能拥有大规模可视化集成开发环境Ajax技术

### 学习EXTjs

#### 学习ExtJs前提

JavaScript JSON java C# Web Services HTML Dom Ajax 基于类化思维JavaScript编程

#### ExtJs基本概念



##### ExtJs——组件 component

能够以图形化形式呈现界面的类，其中还可以分为容器组件与元件组件

容器组件：能够包含其他容器组件或元件组件的类，其是进行单元化组件的开发基础

元件组件：能图形化形式表现一个片面功能的组件，不仅在实现原油的传统的WEB程序的现有组件，而且扩展了好多实用的组件，让刻度器、日历、树形例表

##### EXTJS——类 class

* 提供功能的非图形可型的类，他们为图形类提供了有支持
* 按功能可分为：**数据支持类（data）、拖放支持类（dd）、布局支持类（layout）、本地状态存储支持类（state）、实用工具类（util）**
* 密封类：不能被扩展的类
* 原型类：扩展了javaScript标准类库中类

##### EXTJS——方法 method

作为类的功能体现，能够产生该表对象本身产生变化的直接因素

方法按访问形式可分为公有方法和私有方法，但是由于javascript从原理上根本不支持这中结构，所以在EXTJS中，私有与公有方法完全靠用户自觉，没有像java那用强制

##### EXTJS——事件event

由类定义的，并且可以在类对象自身状态发生改变的触发

只有被订阅的事件才会有效

如果不需要此事件，应该进行退订，增强程序的执行效率

##### EXTJS——配置选项 config option

用以初始化一个EXTJS类对象的手段

注意，配置选项并不一定就是属性，就算是属性，也可能出现属性返回类型与当初指定配置类型不一致情况

##### ExtJs 在面向对象所出的努力

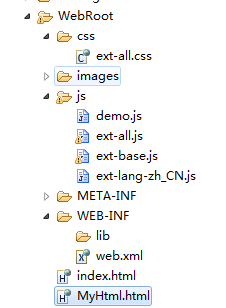
* 支持命名空间
* 类实例属性
* 类实例方法 类静态方法 构造方法
* 类继承
* 类实例方法重写
* 命名空间别名
* 类别名
* 事件队列

### 实例测试：

命名空间：是java中包名（Ext.namespace(“Ext.wuliang”)）

#### 第一例子：

1. 学js在MyEclips中没有自动提示功能，比较麻烦，所以需配置Spket，才会有提示功能 <http://wangke0611.iteye.com/blog/550257> 这个就有spket破解版和如何配置spket，这里就多说了
2. 下载ExtJs破解版，网上了解，4版本以上比较难理解，还是下容易的在继续<http://www.sencha.com/products/extjs/download/> 官网
3. 现在开始做：



必要的最小集合是这样的： ext-3.4.0/ext-all.js

ext-3.4.0\adapter\ext\ext-base.js

ext-3.4.0\resources\css\ext-all.css 所有css

ext-3.4.0\resources\images 所有图片

**ext-3.4.0\src\locale\** *ext-lang-zh\_CN.js显示出中文*

*注意导入顺序，切忌顺序错误*

html：

<script type=*"text/javascript"* src=*"./js/ext-base.js"*></script>

<script type=*"text/javascript"* src=*"./js/ext-all.js"*></script>

<script type=*"text/javascript"* src=*"./js/ext-lang-zh\_CN.js"*></script>

<script type=*"text/javascript"* src=*"./js/demo.js"*></script>

<link rel=*"stylesheet"* type=*"text/css"* href=*"./css/ext-all.css"*>

</head>

<body>

</body>

Demo

Ext.onReady(**function**(){

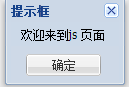
Ext.MessageBox.alert("提示框","欢迎来到js 页面");

});

出现错误：ext is not defined

**注意：base.js与all.js上下顺序，而且all.js不能再其他js下，所以js中，base与all必须在第一和第二位，不然不会得到正确**

**开始显示的 ok ，加 ext-3.4.0\src\locale\** *ext-lang-zh\_CN.js显示出中文*



**Ext：**

Ext.MessageBox.confrim()

在加载ExtJs中js与css文件完后，ExtJs就执行Ext.onReady()中指定的函数

#### 组件之路：

**2：介绍**

**组件分为：基本组件，工具栏组件，表单及元素组件**

#### 2.1 Ext.Button

1. **构造方法：**

**text：按钮上的文本**

1. **属性：**

**text：获取当前按钮上的文本**

minWidth**： 按钮的最小宽度**

1. **方法：**

**s**etText**：设置按钮上的文本**\_button.setText('取消')//修改文本

**4．事件**

**Click：点击按钮事件**

Ext.onReady(**function**(){

**var** \_button = **new** Ext.Button({

text:'确定',//button上文本

// renderTo:document.body

renderTo:Ext.getBody(),//现在出来位置

minWidth:100

//handle是选择默认事件 click

// handler:function(){//增加事件

// Ext.MessageBox.alert('HelloWorld');

// }

//多个事件用

listeners:{

"click":function(){

alert("点击");

},

// "render":function(){

// alert("组件对象加载中");

// }//当renderTo把Button加载到指定位置时render响应

// }

});

\_button.on("click",**function**(){alert("点击")});

// Ext.MessageBox.alert(\_button.text);//text拿到Button中的文本

\_button.setText('取消')//修改文本

}) 

当renderTo把Button加载到指定位置时render响应

handle是选择默认事件 click

#### 2.2 Ext.form.TextField



**组件取代以前的Text（表单中组件都用Ext.form.）**

1. **属性**

**fieldLabel 文本标签的名称**

1. **方法**

**getValue() 得到当前文本框的值**

**3．示例**

**new** Ext.form.TextField({

fieldLabel:"文本框"

renderTo:Ext.getBody()

//所以的form组件必须放在容器组件中去

})

**Ext.form.中form组件必须方法在容器组件，所以我们必须前提是学习容器组件Panel**

**Panel 容器组件**

**此示例运行后结果并不显示 fieldLabel 的内容，这是为什么？**

**这是因为我们没用布局，只有在Ext.layout.FormLayout 布局下才能正确表示文本框的标签名，布局主要是**Ext.Container **或者子类**

**在应用FormLayout布局时，只需要在容器组件中加layout:"form",**

#### 2.3 Ext.Panel容器组件

**改变传统网页通过html表格形式步局的历史**

1. **属性**

**title : 面板的标题**

**width：面板的宽度**

**height: 面板的高度**

**frame: 为true时将四个角为圆角**

1. **方法**

panel.add()

1. **示例**

Ext.onReady(**function**(){

**var** \_panel = **new** Ext.Panel({

title:"登录",

width:260,

height:180,

layout:"form"**,**//在面板上必须有labelWidth 一般都是 45

labelWidth**:45//这个是调整字体与文本框之间的距离**

**//** renderTo**:Ext.getBody() 这个意思放到body中就加载，而不会在使用add中的功能**

});

\_panel.add(**new** Ext.form.TextField({

fieldLabel:"文本框"

// renderTo:Ext.getBody()//所以的form组件必须放在容器组件中去

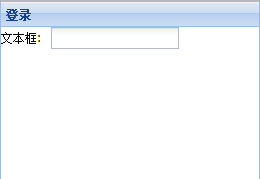
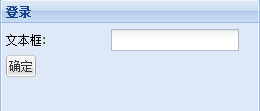
}));

\_panel.render(Ext.getBody());//**当我必须把panel放在body中去，就必须render**

**renderTo是在方法里面使用，是一个属性**

**render是把组件放进去，是一个方法**

**frame:true区别**



})

Ext.onReady(**function**(){

**var** \_panel = **new** Ext.Panel({

title:"登录",

width:260,

height:180,

layout:"form",

renderTo:Ext.getBody(),

listeners:{

"render":**function**(\_panel){

\_panel.add(**new** Ext.form.TextField({

fieldLabel:"文本框",

**id:"form.username"**

}));

\_panel.add(**new** Ext.Button({

text:"确定",

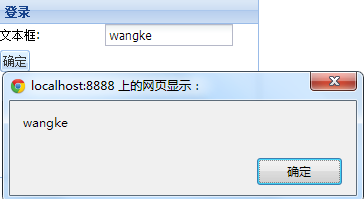
handler:**function**(){

alert(Ext.getCmp("form.username").getValue());//拿到id中值 }

}));}} });

})

Ext.getCmp("form.username").getValue()通过id来拿到id的值



**\_panel.AddButto()增加一个按钮到面板中**

**\_panel.addButton与\_panel.add不同之处**

\_panel.addButton({text:"ok"});

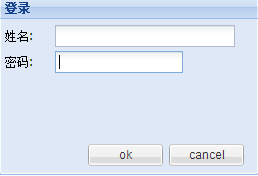
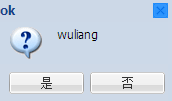
\_panel.addButton({text:"cancel"});

\_panel.add(Ext.MessageBox.confirm("ok","wuliang"));

\_panel.render(Ext.getBody());

\_panel.add(Ext.Button({text:"soo"}));

**两种添加按钮方式的区别，第一种有panel的创建管理，第二种是我们自己创建的按钮添加到panel中去，两种方式添加按钮的样式不同**

##### 区中布局：

由css布局样式表：

<style type=*"text/css"*>

*.contain*{

width:*100%*;

height:*100%*;

top:*0*;

left:*0*;

}

*.center*{

position:*absolute*;

top:*30%*;

left:*43%*;

text\_align:*left*;

}

</style>

Js中改动：

Ext.onReady(**function**(){

**var** \_panel = **new** Ext.Panel({

title:"登录",

width:260,

height:180,

frame:**true**,

layout:"form",

labelWidth:45

});

\_panel.add(**new** Ext.form.TextField({

fieldLabel:'姓名',

width:180

}));

\_panel.add(**new** Ext.form.TextField({

fieldLabel:'密码',

width:180

}));

\_panel.addButton({text:"ok"});

\_panel.addButton({text:"cancel"});

**// Ext.DomHelper.append("添加对象的位置 ",{添加对象，div**

**// tag："标签名称",cls:"class名"},**

**// cn:{子节点中 tag：与cls**

**// })，true是返回对象EXT而不是document对象**

**var** \_div = Ext.DomHelper.append(

Ext.getBody(),{

tag:"div",

cls:"contain",

cn:{

tag:"div",

cls:"center"

}

},**true**

);

//整体返回对象是一个div

**\_panel.render(\_div.child("div"));//找其中div对象的子节点就是cn中的，“div”是子节点名称**

})

applend 在一个对象增加一个子节点

其中可以是这样的

<body>

<div calss=*"contain"*>

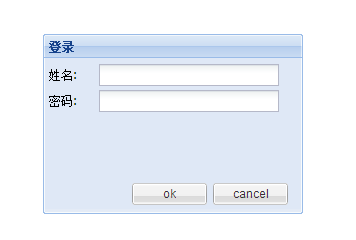
<div class=*"center"*>

再把panel放进子节点中去

</div>

</div>

</body>



例如：var list = Ext.DomHelper.append('parent', {tag: 'div', cls: 'red'});它就是向id=parent这个元素里，添加一个div元素。 按照文档里讲的，第二个参数{}里，除了四个特殊属性以外都会复制给新生成元素的属性，这四个特殊属性是：

1.tag，告诉我们要生成一个什么标签，如div、span等。

2.cls，指的是<div class="red"></div>这种标签中的class属性，因为class是关键字，正常情况下 应该写成className，cls是缩写。诸如此类缩写的还有（dataStore写成ds，DomHelper写成dh， Element写成el，ColumnModel写成cm，SelectionModel是sm。）

3.children，用来指定子节点，它的值是一个数组，里边包含了更多节点。

4.html，对应innerHTML，觉得用children描述太烦琐，直接告诉节点里边的html内容也是一样

document.body 与 Ext.getBody()是不同的对象 下次讨论

##### 新版的创建组件：

**var** \_panel = **new** Ext.Panel({

buttons:[{

text:"确定"

},{

text:"取消"

}]

});

旧版：\_panel.addButton({text:"ok"});

从新版本开始，可以不需要new出来对象

\_panel.add(**new** Ext.form.TextField({

fieldLabel:'密码',

width:180

}));

\_panel.add({

xtype:"textfield",

fieldLabel:"生日",

width:180

});

还可以在创建容器对象中去

items:[{

xtype:"textfield",

fieldLabel:"生日",

width:180

},{

xtype:"textfield",

fieldLabel:"日期",

width:180

}]

当如果重复的可以用default

defaults:{ xtype:"textfield", width:180},

items:[{

fieldLabel:"生日"

},{

fieldLabel:"日期"

}]

defaults默认，所有items对象默认公共属性

defaultType：只能指定items里面xtype，defaultType:"textfield"

如果要查看xtype时<http://localhost:8888/ext/docs/> component

#### 2.4Ext.Window

该组件是实现窗体化开发的基础

* + 1. 构造参数

title 窗体名称

Minimizable 窗体最小化

Maximizable 窗体最大化

* + 1. 方法

show：显示窗体

wide：隐藏窗体

其中用panel改成window必须show()出来

**其实window是panel子类，panel容器包含组件不能拖动的，所以有了window组件，可以拖动的，但是结果来说我的是不能拖动的，但是可以拖动大小**

plain:**true**,强制背景色，使背景色相同

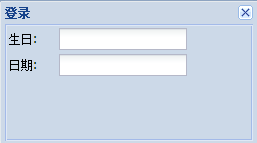
 

resizable:**false**,不能改变大小 默认为true

buttonAlign:"center",改变窗体中button的位置



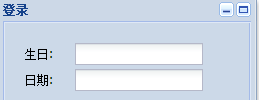
bodyStyle:"padding:2px", 20px 改变文本框与padding距离

minimizable:**true**,minHeight:0,minWidth:0,最小化图标 目前使用没有用

maximizable:**true**,最大化图标

closable:**false**, 关闭图标 默认为true



##### 窗体的优化：

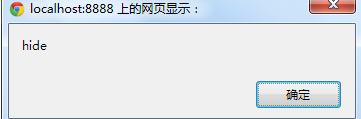
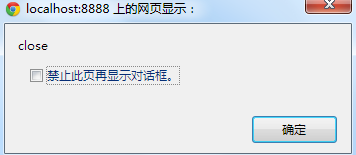
**不建议调用close对象，建议隐藏对象**

**先创建Ext对象，才创建document对象**

close是先从Dom中删除EXT对象，对cpu消耗过大

EXT中有dom，建议先隐藏，关闭是，重新出来的时候还是需要new，

**原因是**：Ext对象比依附于dom，EXt只是一种扩展，事实上是先创建对象EXT对象再创建dom对象，EXT中有dom对象，当关闭这个EXT对象的时候，必然会调用dom中close对象来删除EXT对象，消耗cpu过大，所以建议隐藏EXT对象，不会调用dom对象，重新调用时效率更高

closeAction:"hide",//默认close 加上这个才能真正的hide，而不是关闭

listeners:{//多事件监听

"hide":**function**(){

alert("hide");

},

"close":**function**(){

alert("close");

}

},

##### This，ownerCt，items

**1.this用来获取发生事件的当前对象**

**2.ownerCt是拥有当前对象的Contaiter对象**

**3.items是当前容器对象所拥有的对象的集合**

**Items是当前容器对象存放各种组件**

buttons:[{

text:"确定" ,

handler:**function**(){

alert(**this**.text);

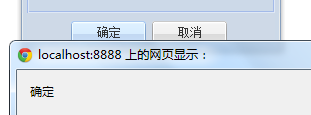
// alert(this.ownerCt.ownerCt.title);

// alert(this.ownerCt.ownerCt.items.first().getValue())//拿到是一个集合

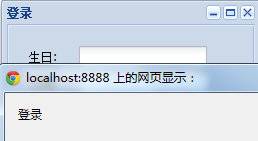
}

}

1.拿到button中：alert(**this**.text);



2.在3.x中最外层是window，还有bar（我们窗体就放在这），**this**.ownerCt.ownerCt.title



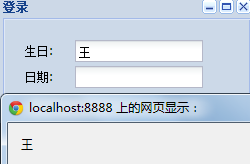
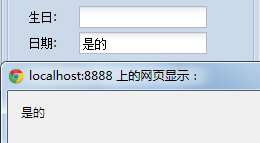
3.拿到文本框中字符，可以看到items中就

**this**.ownerCt.ownerCt.items.first().getValue()

得到第二个可以用itemAt(1)

alert(**this**.ownerCt.ownerCt.items.itemAt(1).getValue());

items.get(1).getValue()

##### EXT.Window例子

Ext.onReady(**function**(){

**var** \_window = **new** Ext.Window({

title:"登录",

width:260,

height:180,

frame:**true**,

**plain:true,**

**resizable:false,**

**buttonAlign:"center",**

**bodyStyle:"padding:20px",**

**minimizable:true,**

**minHeight:0,**

**minWidth:0,**

**maximizable:true,**

**closable:true,**

**closeAction:"hide",//默认close**

layout:"form",

labelWidth:45,

buttons:[{

text:"确定" ,

handler:**function**(){

// alert(this.text);

// alert(this.ownerCt.ownerCt.title);

// alert(this.ownerCt.ownerCt.items.first().getValue())//拿到 b 是一个集合

alert(**this**.ownerCt.ownerCt.items.get(1).getValue());

}

},{

text:"取消",

handler:**function**(\_window){

\_window.hide();//点击后隐藏

}

}],

listeners:{ //多事件监听

"hide":**function**(){

alert("hide");

},

"close":**function**(){

alert("close");

}

},

defaultType:"textfield", //默认Items中xtype

items:[{

fieldLabel:"生日"

},{

fieldLabel:"日期"

}]

});

\_window.show(Ext.getBody());

})

如果是写死了，用不好指定id就用this

一开始以前render时候是把框架给加载上，而不是把整体

from

swing等等都没学，cs架构的思路，必须要这个思想，ext中有布局，一种例的布局，网格，上中左右布局的方法，ext不同传统的网页，

**panel一个大容器，form布局，想单选框就是上姓名等等label都是form布局。要显示label必须上form布局**

###### **labelWidth**

labelWidth**:55,**//只要是显示出label 名字叫文本框 都是form表单，而且labelWidth中必须55正好是四个字的标准

###### Plain 与 x-plain 不同之处

在window容器中有plain：true，可以设置背景色相同，而在panel面板上没有这个属性，所以只能用css样式表，而且window是panel的子类，其中plain也是通过css的x-plain来改变背景色，所以说只要是设置背景色相同，就直接用baseCls:’x-plain’，就可以了

**各种错误：**

1. **Uncaught SyntaxError: Unexpected token ILLEGAL**

**说明一处出现致命语法错误**