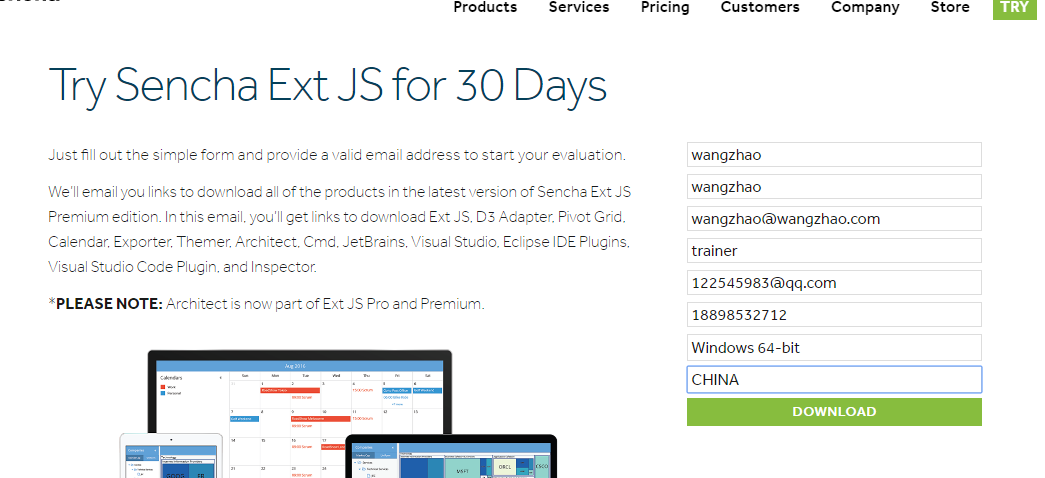
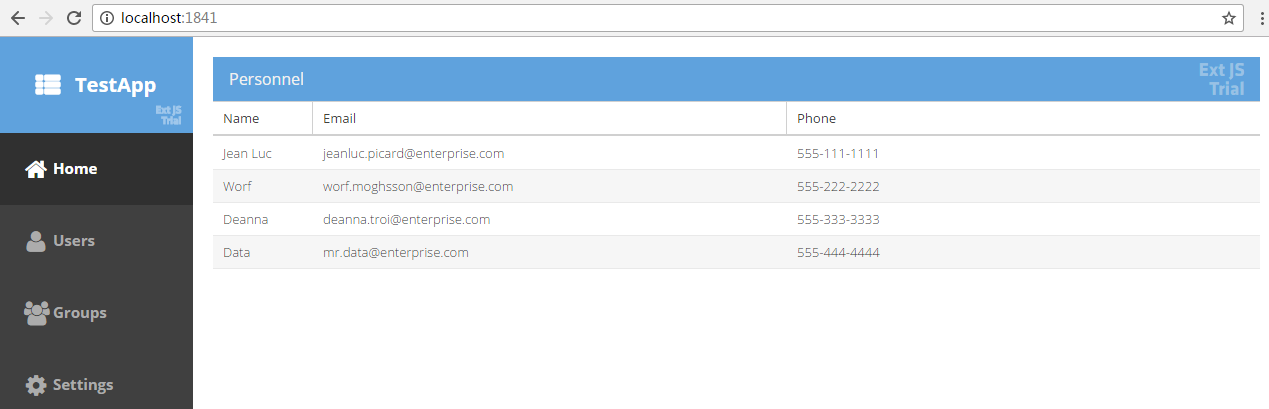
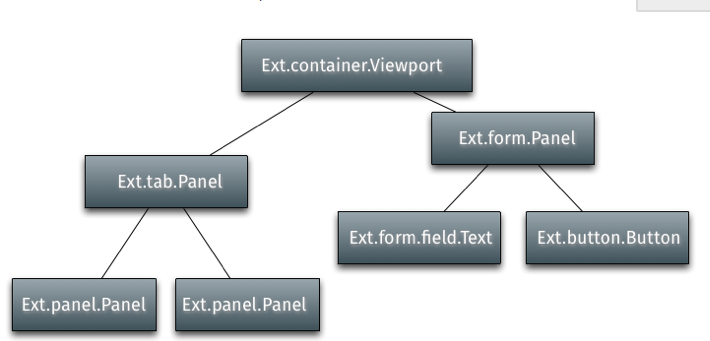
Extjs

C:\Users\LQC\Downloads\apache-tomcat-8.5.14\webapps\ROOT\build\examples\classic





环境搭建：（<http://www.cnblogs.com/yygoa/articles/6219698.html>）

第一步：下载安装sencha cmd

第二步：第二步：下载Extjs6.2。

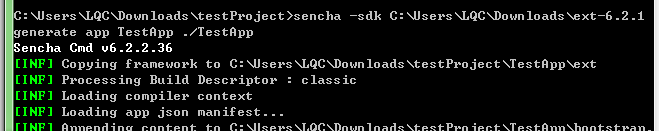
第三步： 使用sencha命令创建应用，部署到sencha自带的jetty服务器。（jetty是一个开源的servlet容器)。

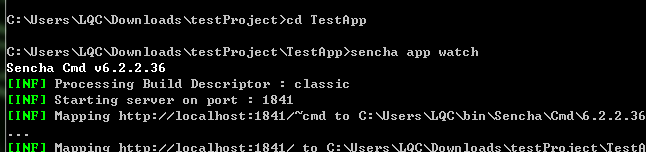
命令为:(命令结构：sencha -sdk 第二步下载的sdk解压的目录 generate app 应用名称 应用存放路径)。

下面是要在testProject中新建一个名字为TestApp的文件。

sencha -sdk C:\Users\LQC\Downloads\ext-6.2.1 generate app myApp C:\Users\LQC\Desktop\testAPP\myApp (第二次在桌面创建)

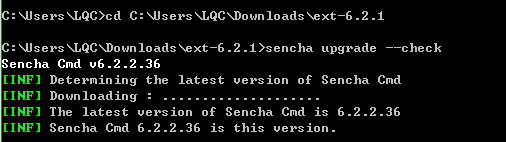




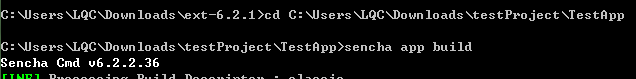




更新版本：sencha upgrade --check



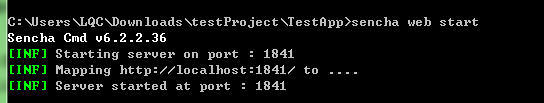
打包文件（要切换到项目根目录）





开启网络服务容器

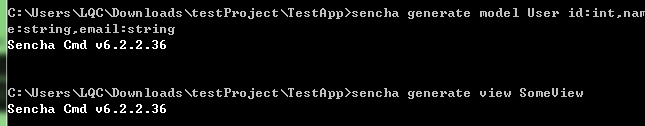
sencha web start（地址栏输入<http://localhost>:1841）



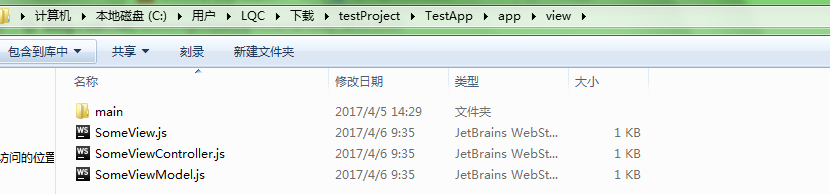
停掉服务器:sencha web stop



新增新的module view controller :sencha generate moel User

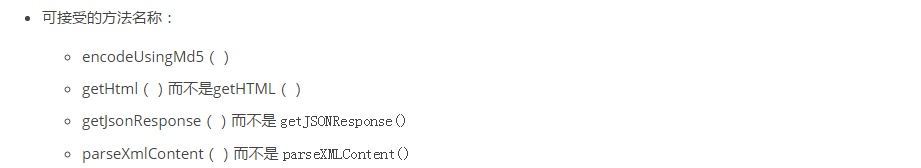






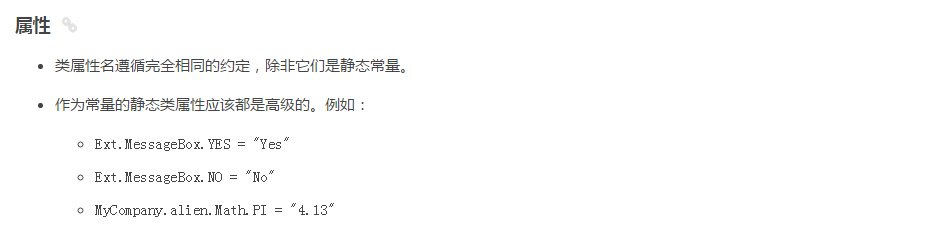
命名规范：

方法：（类似于类名）只能包含数字字母字符，一般不鼓励用数字，除非属于技术用语。



变量：同上（如果是常量）

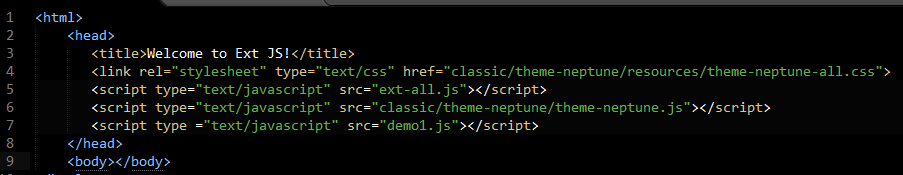




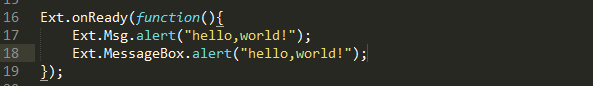
C:\Users\LQC\Downloads\ext-6.2.1\build\classic\theme-neptune\resources-----------css

C:\Users\LQC\Downloads\ext-6.2.1\build---js

C:\Users\LQC\Downloads\ext-6.2.1\build\classic\theme-neptune---js



打印出hello world！



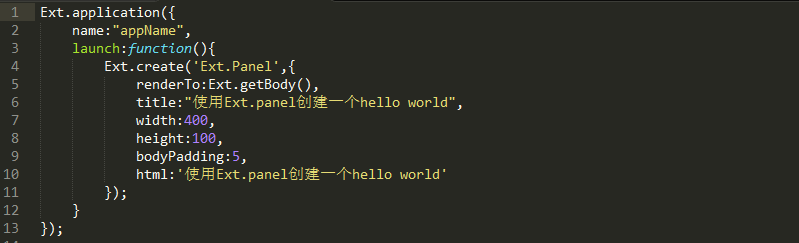
单一的方法创建类（define）

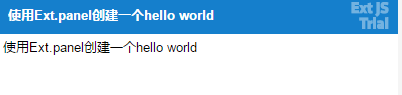
Ext.define(className（类名）, members（类成员集合对象）, onClassCreated（一般都y，回掉函数）);

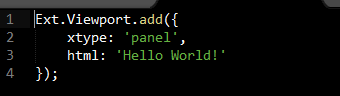
Ext.define(className（类名）, members（类成员集合对象）, onClassCreated（一般都有，回调函数）);

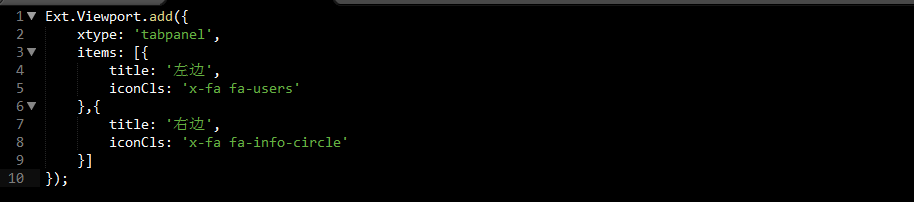


lauch函数是在页面加载完成之后调用该方法。









xtype是替换完全限定组件名称的方便别名。例如，“xtype：”面板“”替换“Ext.Panel”。

Ext.Viewport可以自动根据浏览器视口的大小自动调整大小，并方便地管理添加到视口的组件的大小和位置。经过测试xtype不可以随便写，似乎和加载对应的js有关。

2容器和布局（要么add 要么create（render to） 要么create放置在item对象数组里面）。

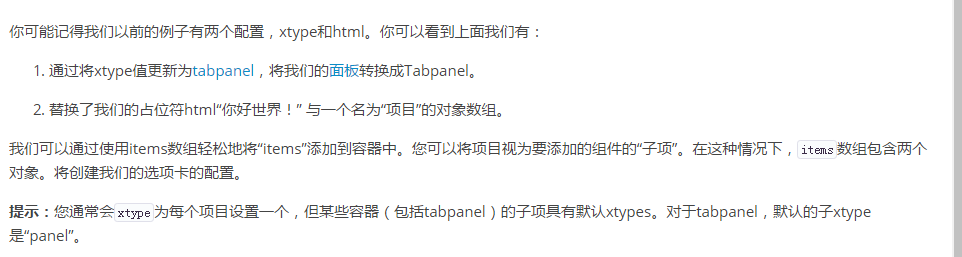
有些时候并不需要全部都渲染

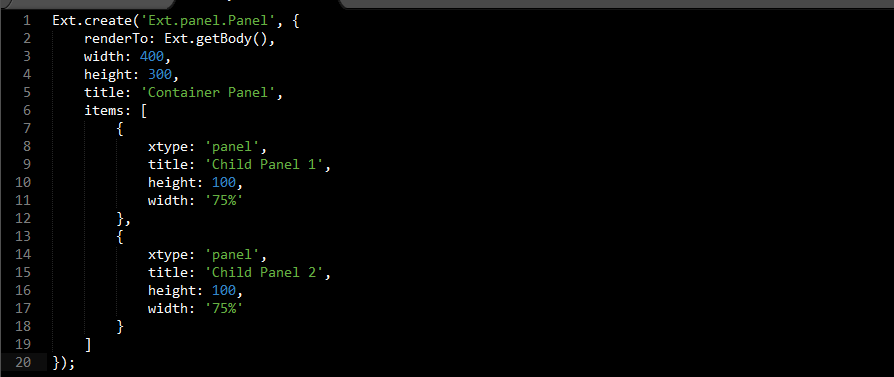


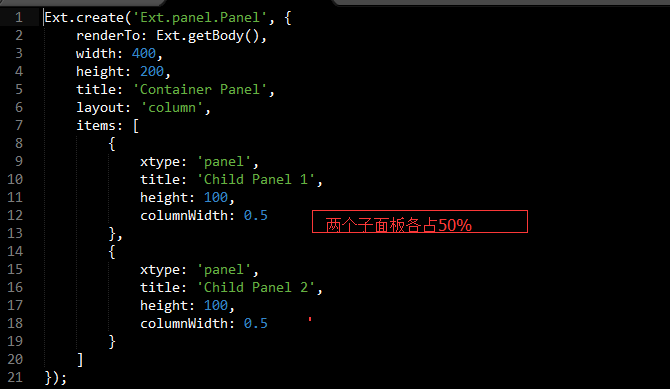
获取配置项中的内容

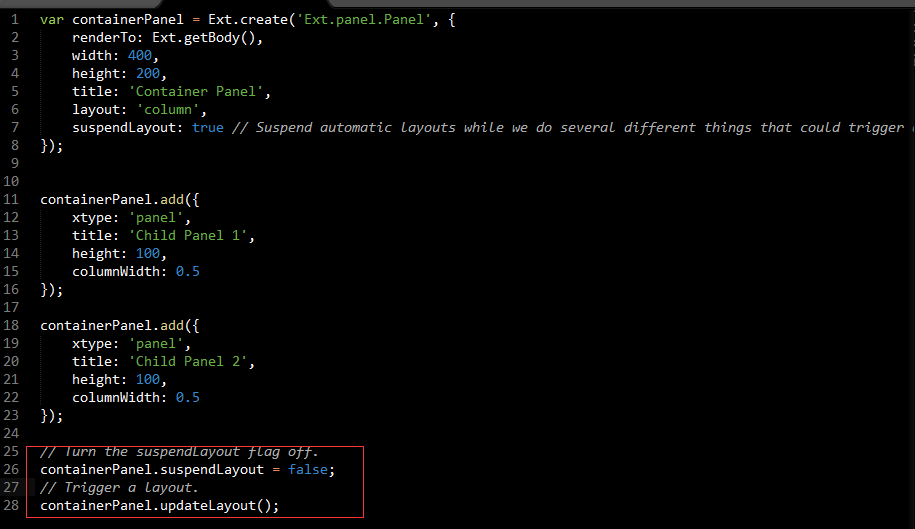










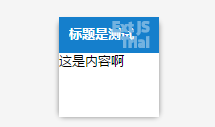
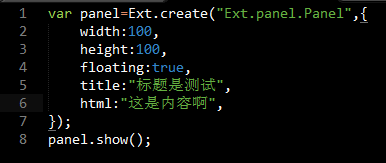




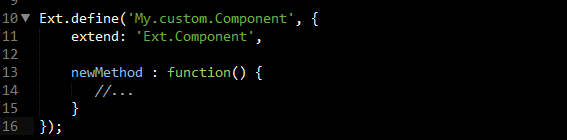
显示和隐藏（所有的组件都有这种方法）



通常，组件可以指定一个[renderTo](http://docs.sencha.com/extjs/6.2.0/classic/Ext.Component.html#cfg-renderTo) 配置，或者作为子组件添加到Container的 [Container中](http://docs.sencha.com/extjs/6.2.0/classic/Ext.container.Container.html)，但是在浮动组件的情况下，这两种方式都不需要。浮动组件在第一次调用[show](http://docs.sencha.com/extjs/6.2.0/classic/Ext.Component.html#method-show)方法时自动呈现给文档主体 。



2.创建组件（创建新的UI类时，必须决定该类是否拥有[组件](http://docs.sencha.com/extjs/6.2.0/classic/Ext.Component.html)的实例，或者扩展该组件。建议将最近的基类扩展到所需的功能。）

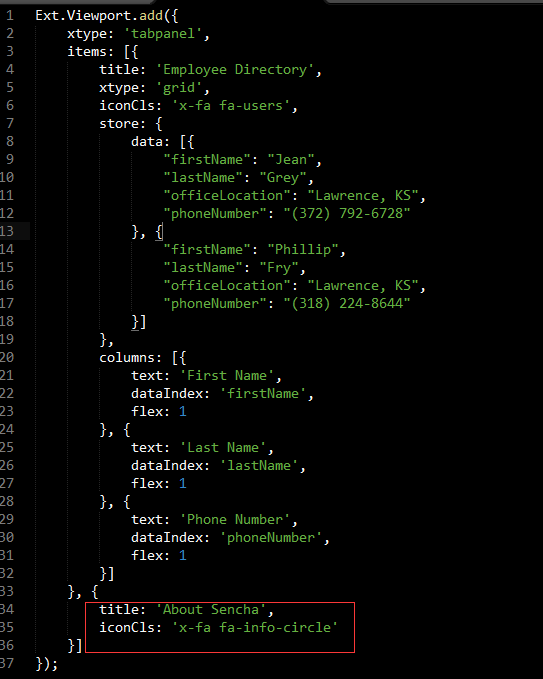


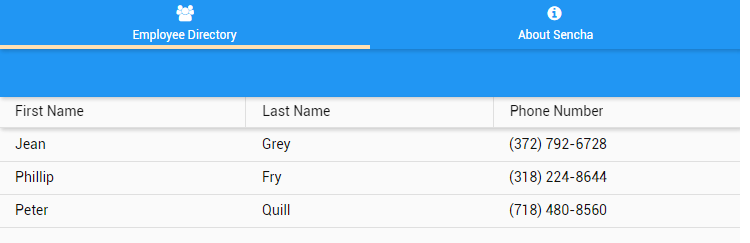
3.呈现数据（带有网格的tab页面）

的网格有一个新的配置 - “store”。Ext JS存储是存放所有数据记录的东西。商店对数据处理非常强大，可以轻松地对存储在其中的数据进行排序，过滤和查询。

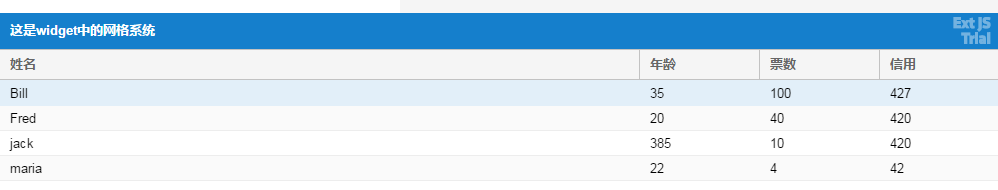
列中三列配置了dataIndex。dataIndex告诉列哪个记录字段将用于该列的显示值。

似乎colums所在的dataIndex一行必须用花括号括起来。

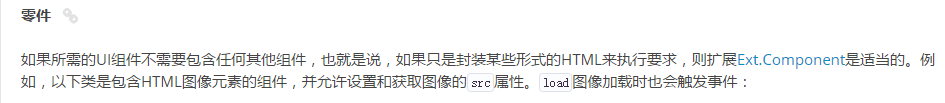






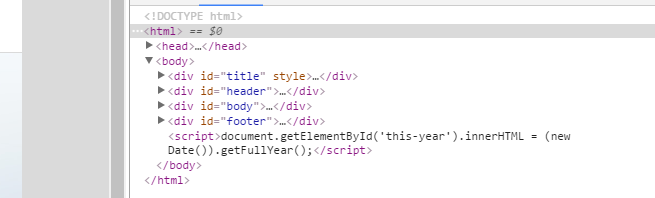


自定义component



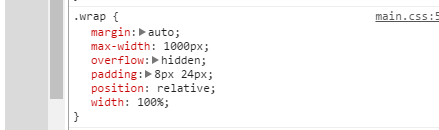
Example

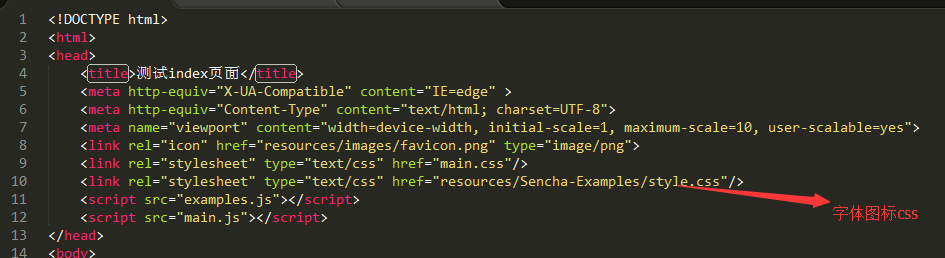
Body中的直系div设置为居中最大长度为1200px。



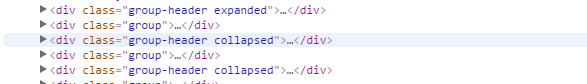


直系div的wrap类设置100%





Css实现隐藏



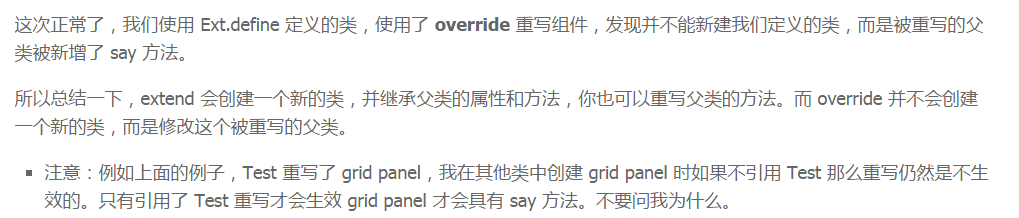


C:\Users\LQC\Downloads\ext-6.2.1\build\examples\classic

Panel



Extend和override的区别

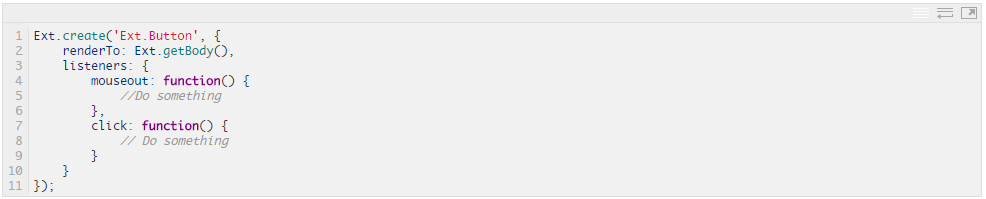


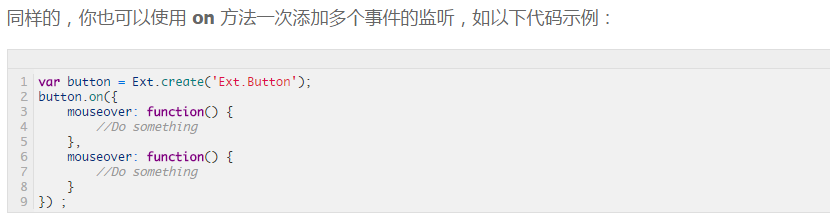


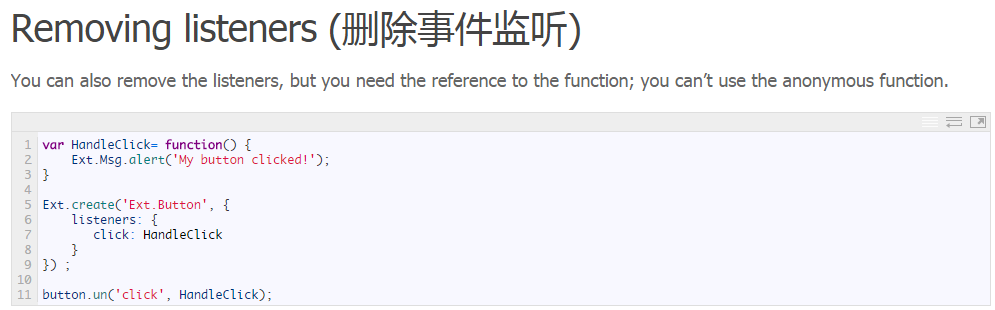




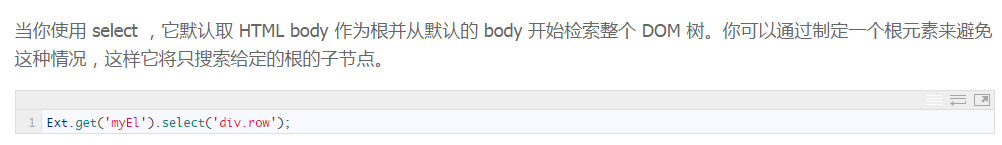
添加事件监听

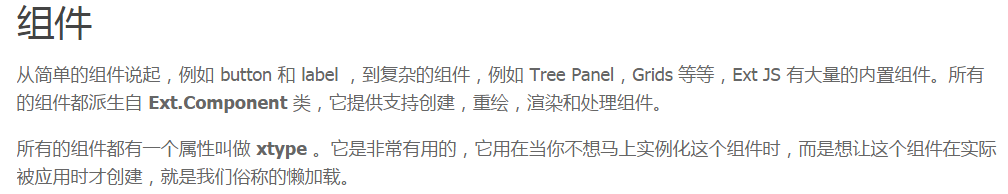


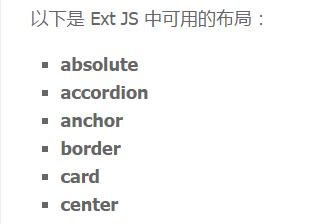




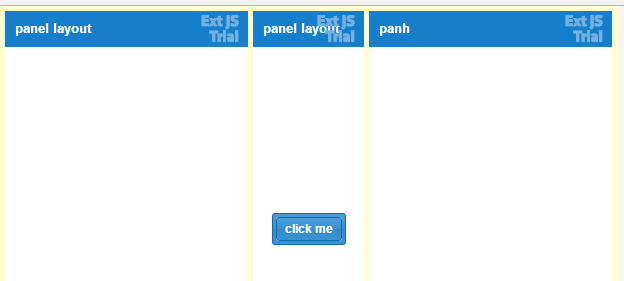
Dom操作 get query select



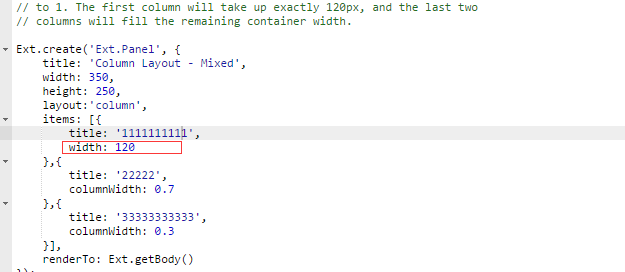


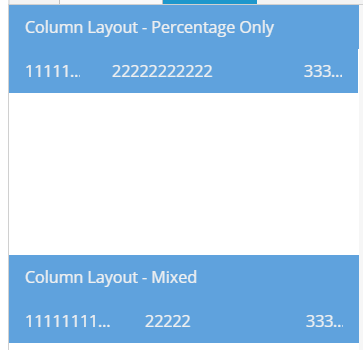


column布局和center布局（宽度怎么占百分百）

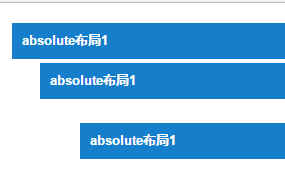
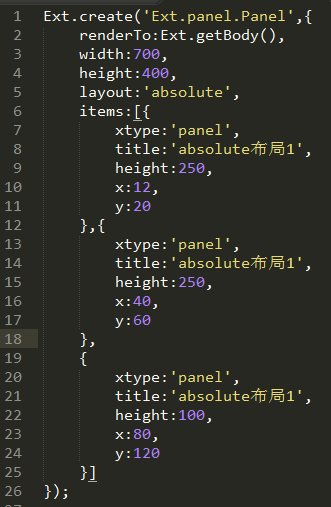




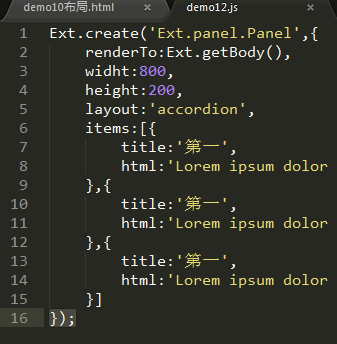


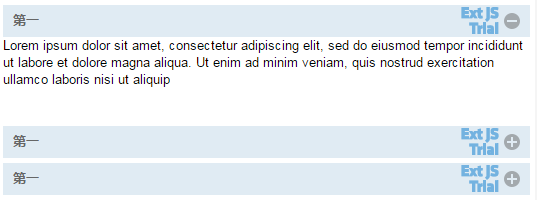


Absolute 使用x和y来指定组件的绝对定位布局



Accordion手风琴可折叠布局

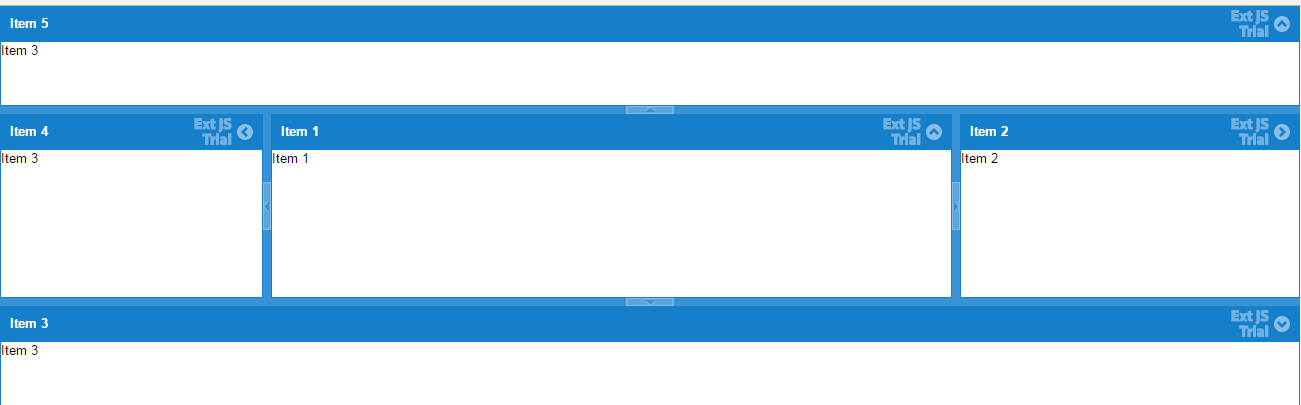


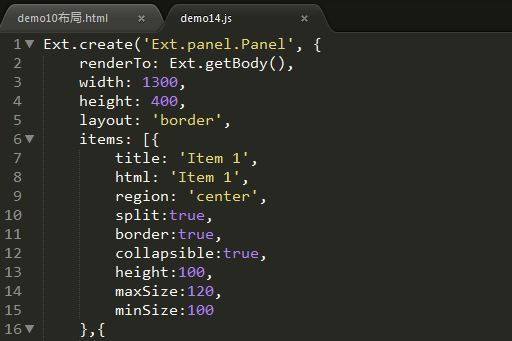


Anchor锚点布局（可以改变子组件的大小）

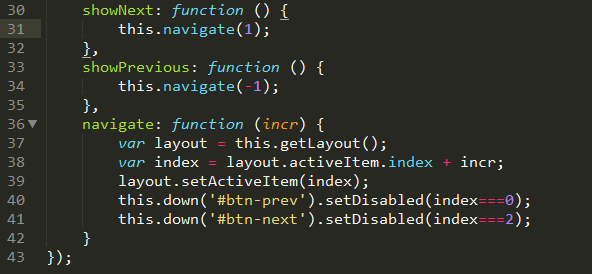
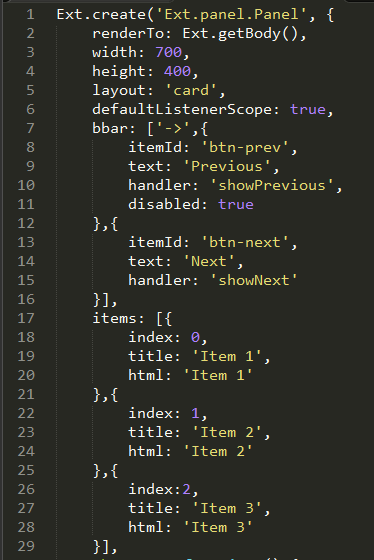


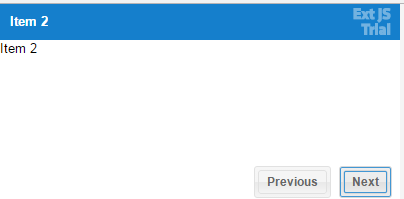
Border布局（五个方向，必须要指出center）可以设置margin north和sounth可以设置高度



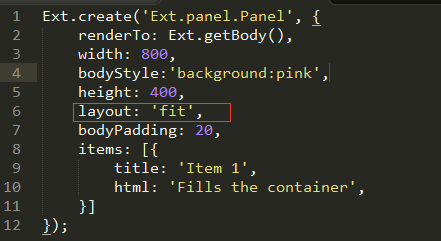


Card卡片布局

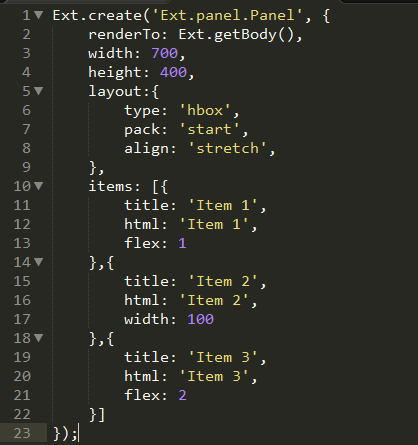


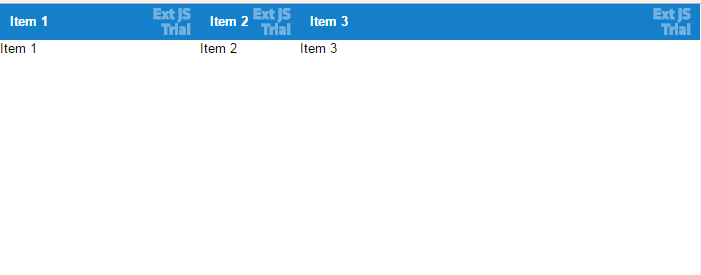


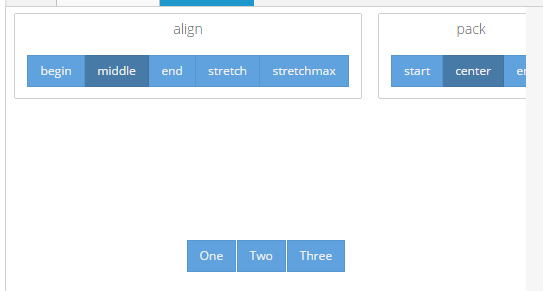
Fit布局（子组件自适应容器的尺寸）



Hbox布局（这种布局与 column 布局几乎是一样的，但是这种布局允许你拉伸列的高度。这里使用 flex 选项为子组件设置水平的相对值）





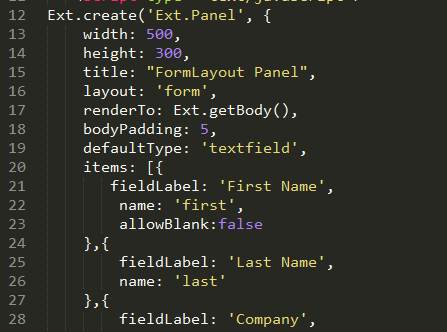


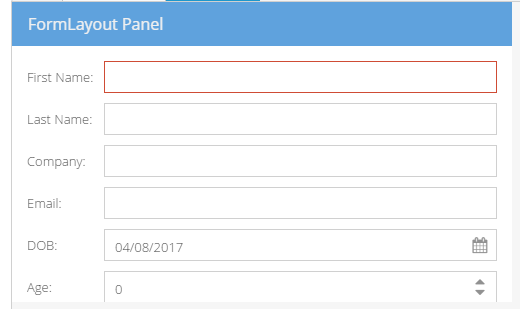
还有table布局和vbox（竖直下来）。

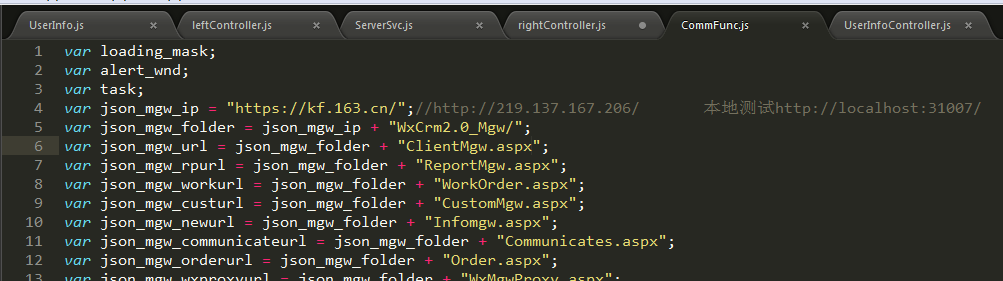
Vbox

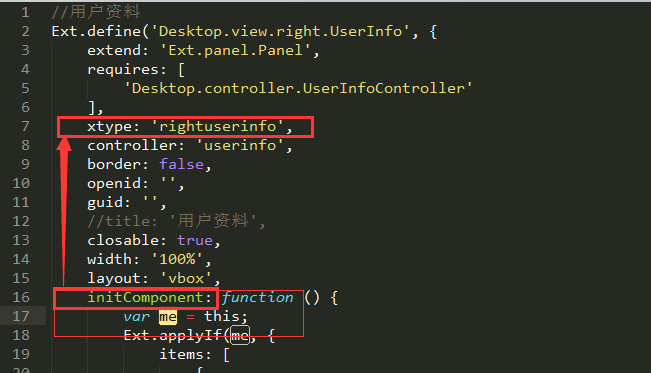


Form布局 （充满容器）

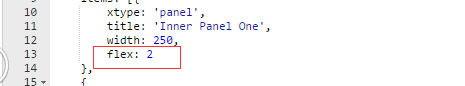




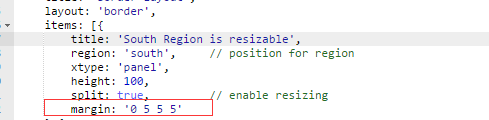




Width和flex同时有，width优先。



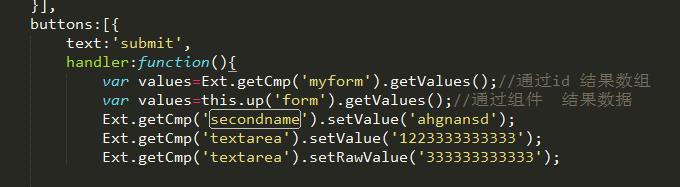
panel的bodyStyle本身有个背景色会覆盖。



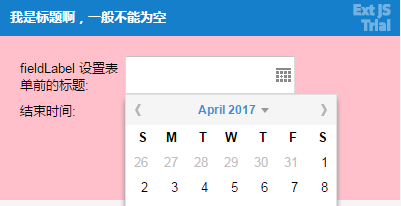
Xtype:textfiled datefield

获取组件：  
- Ext.getCmp('组件id')  
- Ext.ComponentQuery.query('查询条件')  
- 某组件.up('查询条件')  
- 某组件.down('查询条件')  
- 某组件.nextNode('查询条件')  
- 某组件.previousNode('查询条件')  
- 某组件.findParentByType(xtype)  
弹框  
  
Ext.Msg.alert('信息提示','提交成功');  
Ext.Msg.confirm('信息提示','确认删除吗？',function(opt){  
    if(opt=='yes'){  
          
    }  
});  
Ext.toast({  
     html: '提交成功',  
     title: '标题',  
     width: 200,  
     align: 't'  
 });

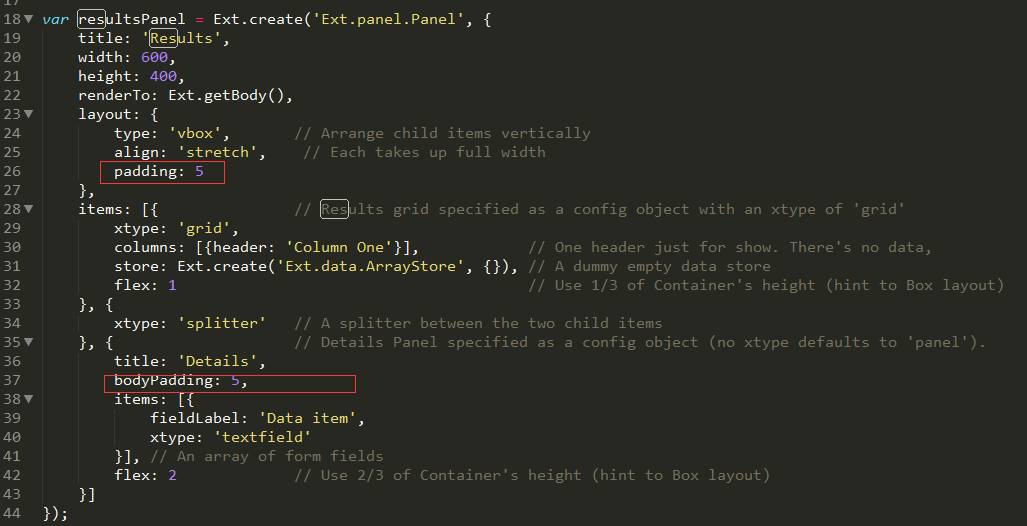
获取值和设置值



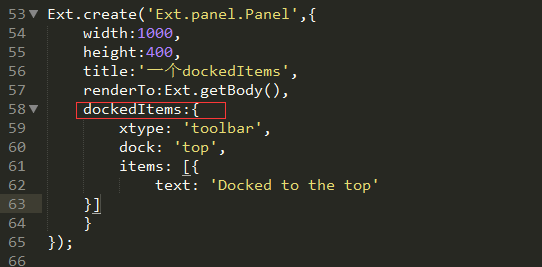




Vbox 中的items下的元素设置flex 一般指的是占据的高度比例



dockItems：中的属性dock可以设置上下左右，从而改变起对应的位置





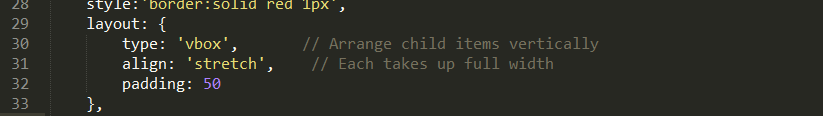
Padding(似乎是整体的)bodyPadding似乎只针对pannel 的内容（标题并不压缩）

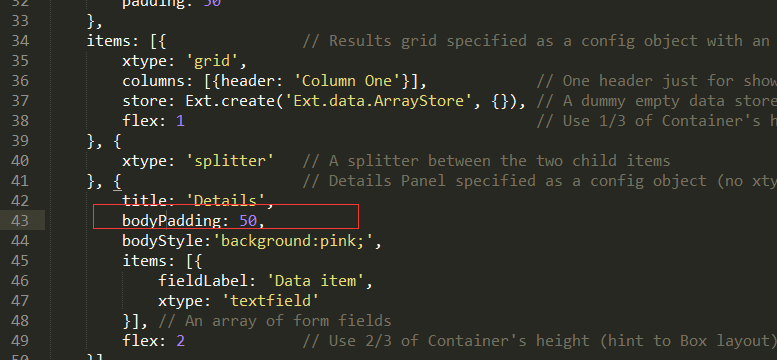


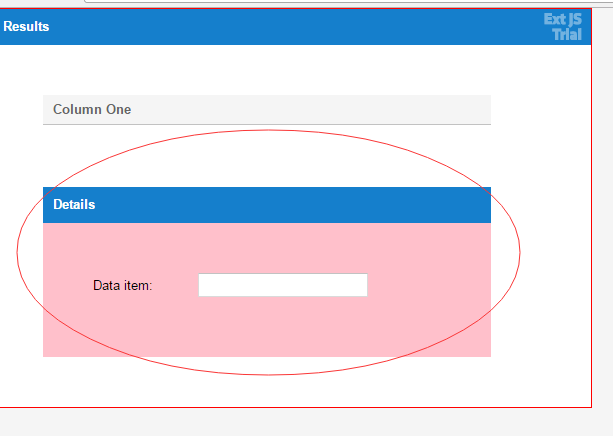




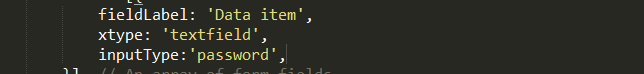






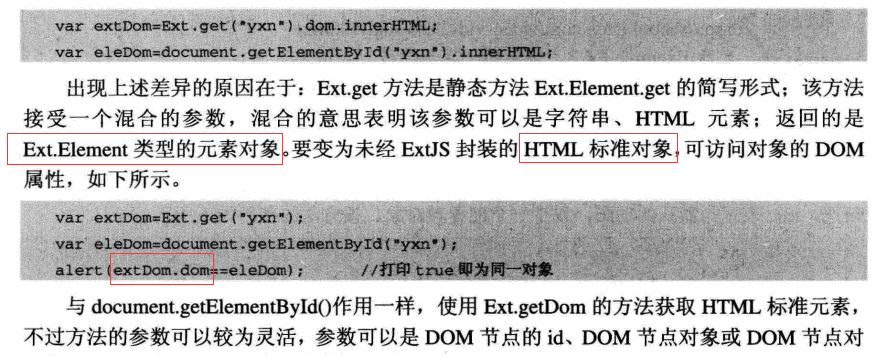


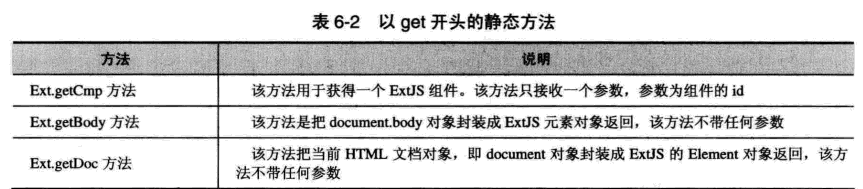
InputType:password





在modern中出现了（使用总是出错）





http://docs.sencha.com/extjs/6.2.0/modern/Ext.Toolbar.html

defer 延迟

container.getEL().on()

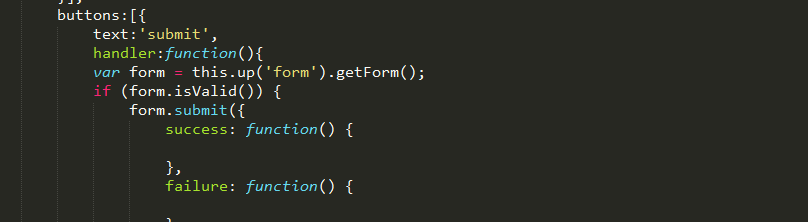
fireEvent

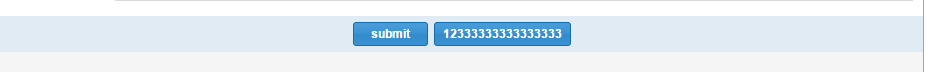
Ext.Msg.alert('Success!', this.getXType() )

声明很重要（Ext.create('Ext.Button',像这样如果不声明，导致button占得高度整个高度）



Button居中 （在panel下声明buttonAlign）





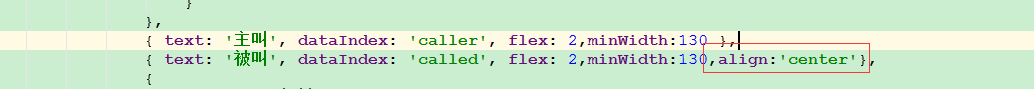
任务

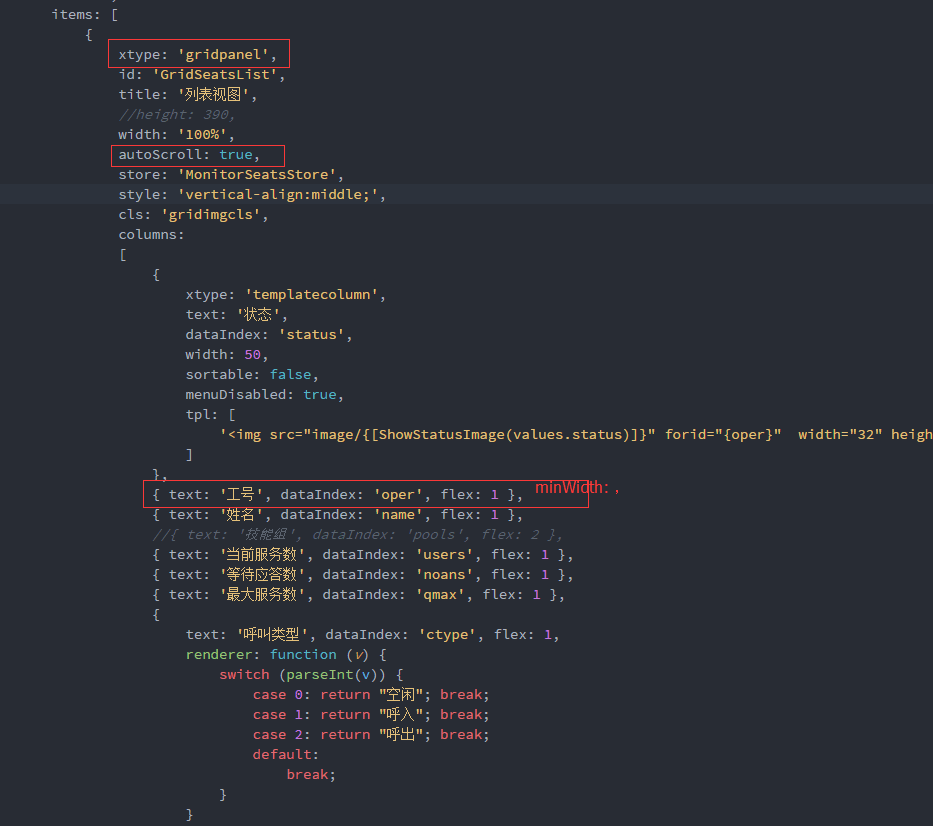
D:\qq_file\1225459083\Image\Group\NSQJSSTBEWYRB0{5X2V{N7C.png

有些列的宽度是固定的（如手机号码、时间），则定义固定宽度width

columns下的项设minWidth和flex，整个grid设autoScroll: true,.

自选项中可以设置align：center来尽心居中。





组件管理器 Ext.ComponentManager

组件的查询方式

通过id来查询

Ext.ComponentQuery.query(“#mypanel”)

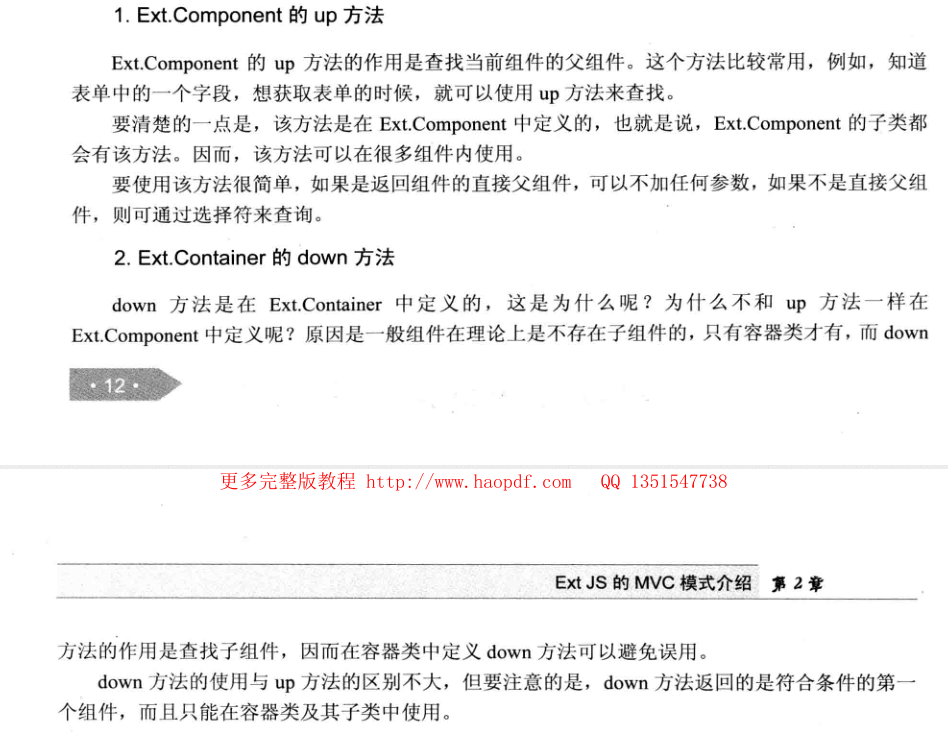
通过别名（xtype）来查询

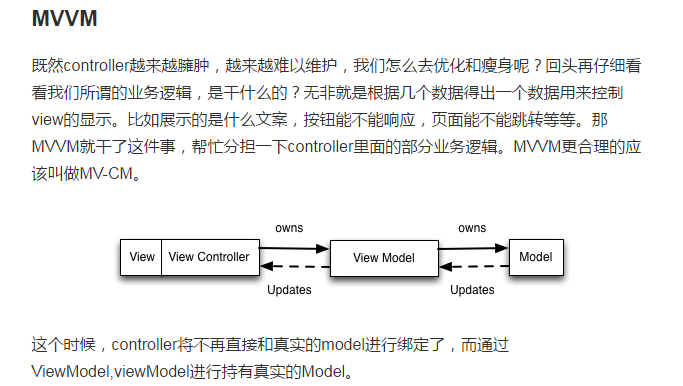
Ext.ComponentQuery.query(“panel”)

Ext.ComponentQuery.query(“.panel”)

使用属性查询

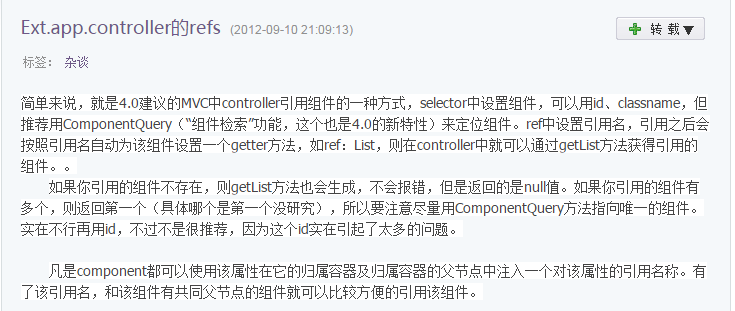
Ext.ComponentQuery,query(“panel[title=我的面板]”);





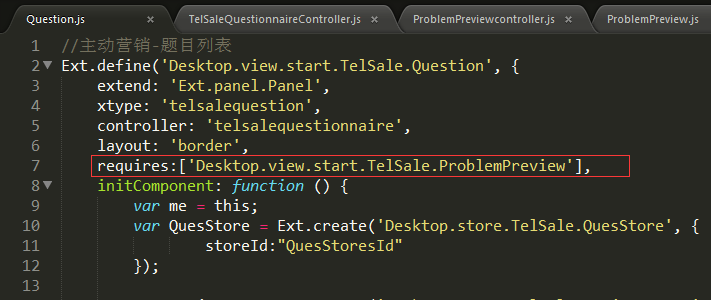


refs



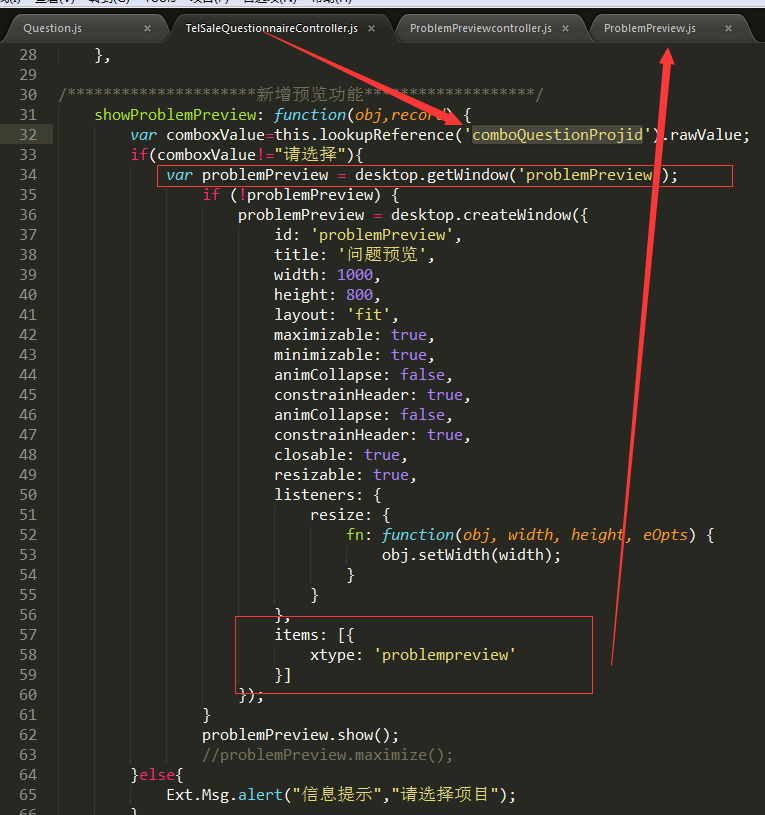
控制器的control功能

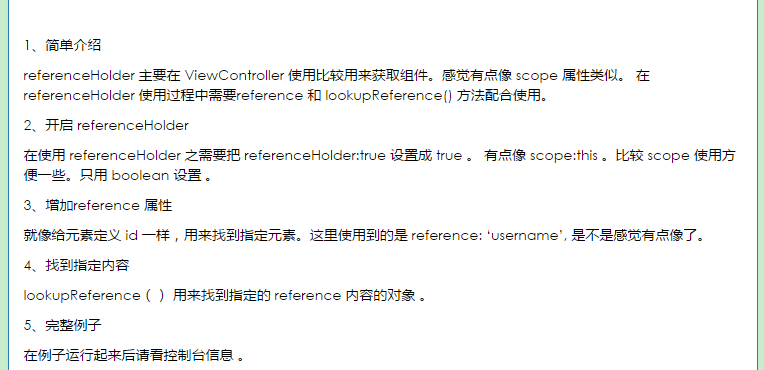
问卷调查问题预览

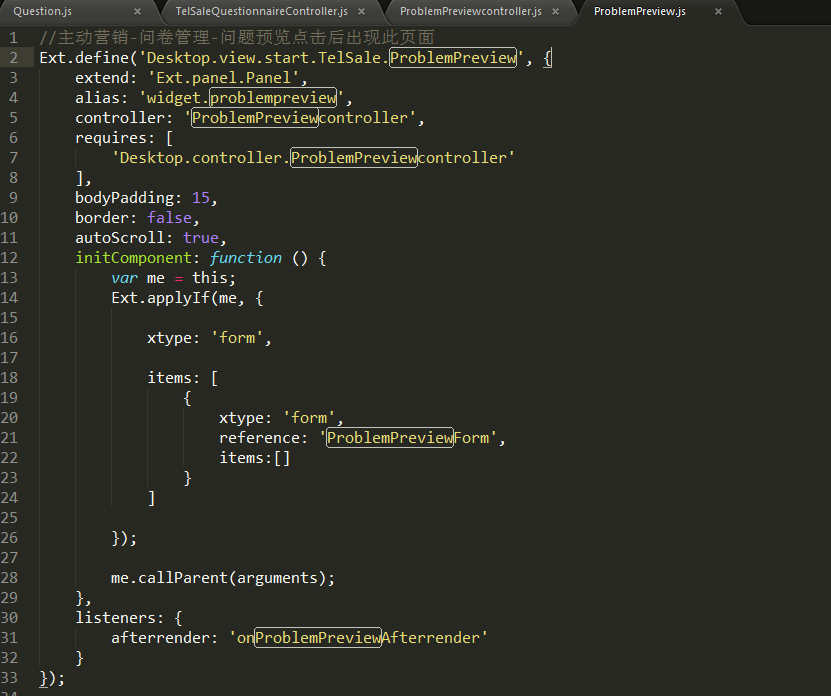




Xtype似乎是调用alias的内容







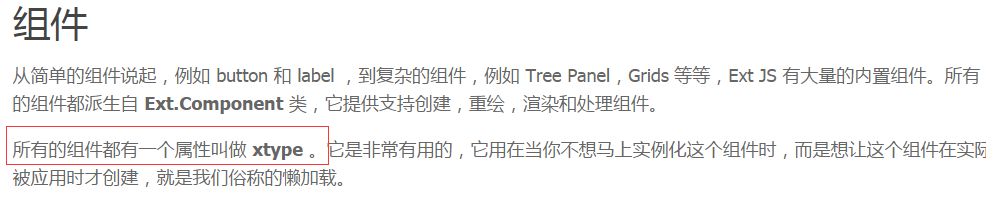


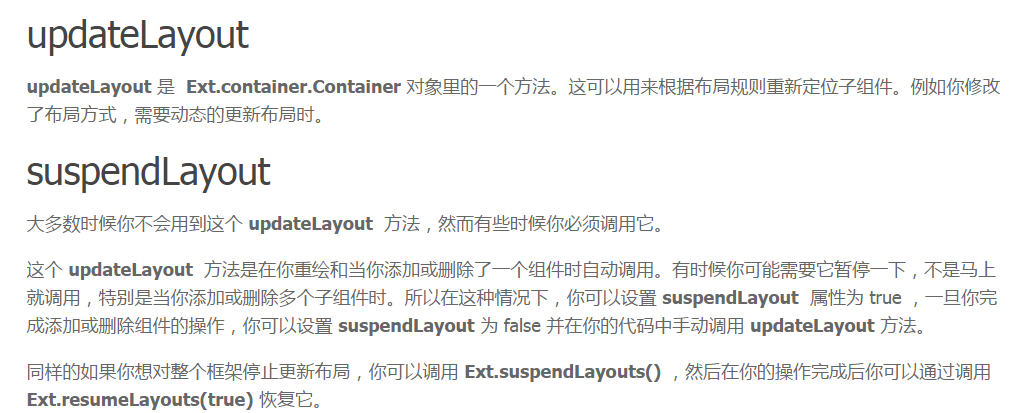
Ext.get(id)

Ext.query(css)

Ext.select(css/xpath)

Ext.ComponentQuery(ID xtype 组件)





Ext.form.field.Checkbox

Ext.form.field.ComboBox

Ext.form.field.Date

Ext.form.field.File

Ext.form.field.Hidden

Ext.form.field.HtmlEditor

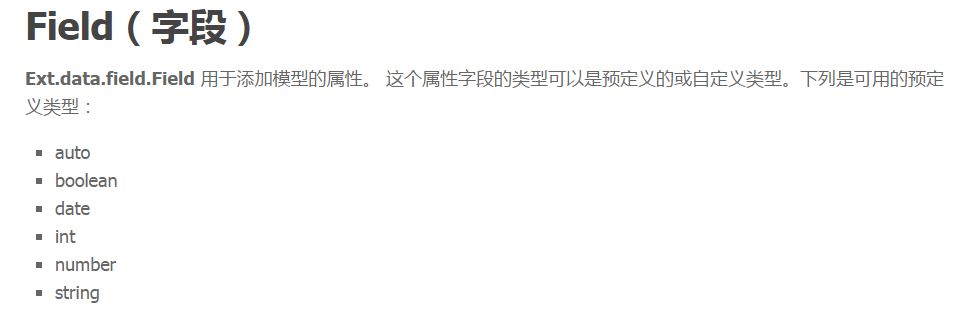
Ext.form.field.Number

Ext.form.field.Radio

Ext.form.field.Text

Ext.form.field.TextArea

Ext.form.field.Time



访问store（创建的时候可以自己添加一个storeId，它会覆盖本身的storeid）

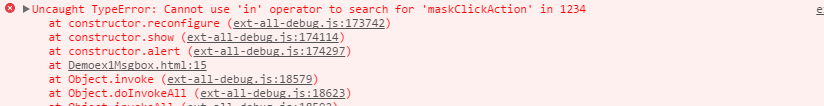
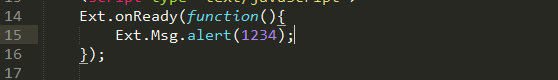
例如storeId:”mystore”;

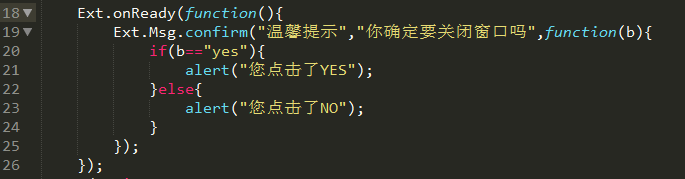
可以通过这种方式访问

Ext.data.StoreManager.lookup(“myStore”);

简写为Ext.getStore(“myStore);

Extjs 中的alert弹窗，直接写数字会报错





获取view和控制器里面的值。

