[MVC的模式，模型（Models）和控制器（Controllers） 1](#_Toc40)

[目录结构如下图所示： 2](#_Toc14990)

[index.html 文件如下 2](#_Toc5921)

[app.js 文件如下： 3](#_Toc9480)

[View 定义一个视图。 4](#_Toc31649)

[创建一个model 文件 5](#_Toc30047)

[controller 层 创建文件 6](#_Toc21560)

[store 创建文件。 8](#_Toc17761)

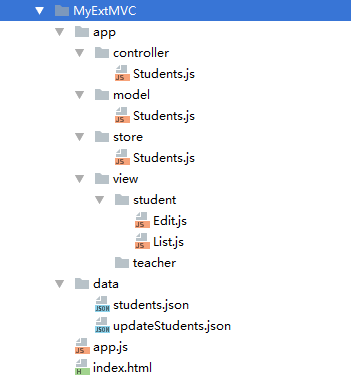
## MVC的模式，模型（Models）和控制器（Controllers）

Model模型 是字段和它们的数据的集合，例如User模型带有username和password字段，模型知道如何持久化自己的数据，并且可以和其他模型关联，模型跟ExtJS 3 中的Record类有点像（区别是，Record只是单纯的扁平结构，而Model可以nest），通常都用在Store中去展示grid和其他组件的数据

View视图 是组件的一种，专注于界面展示 - grid, tree, panel 都是view

Controllers控制器 一个安放所有使你的app正确工作的代码的位置，具体一点应该是所有动作，例如如何渲染view，如何初始化model，和app的其他逻辑

### 目录结构如下图所示：



index.html 文件如下**：**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title></title>

<link rel="stylesheet" href="../resources/css/ext-all.css">

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

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

</head>

<body>

</body>

</html>

Ext.app.application

代表整个应用

Ext.container.Viewport

Viewport渲染自身到网页的documet body区域， 并自动将自己调整到适合浏 览器窗口的大小，

在窗口大小发生改变时自动适应大小，

继承于 ：[Ext.Component](http://extjs-doc-cn.github.io/ext4api/#!/api/Ext.Component)

## app.js 文件如下：

Ext.application({

// 动态加载 这个类。

requires: ['Ext.container.Viewport'],

// 这个应用的名字。

name: 'FWY',

// 应用程序的路径

appFolder: 'app',

// 应用程序控制器名称

controllers: ['Students'],

// 页面 装载完成后自动调用。

launch: function () {

*Ext*.create('Ext.container.Viewport',{

// 布局

layou:'fit',

items: [{

xtype: 'studentlist'

}]

})

}

});

## View 定义一个视图。

Ext.define('FWY.view.student.List', {

extend: 'Ext.grid.Panel',

alias: 'widget.studentlist',

store: 'Students',

title: '学生信息列表',

initComponent: function () {

this.columns = [

{header: '编号', dataIndex: 'id', flex:1},

{header: '姓名', dataIndex: 'name', flex:1},

{header: '年龄', dataIndex: 'age', flex:1},

{header: '性别', dataIndex: 'sex', flex:1}

];

this.callParent(arguments);

}

});

## 创建一个model 文件

*Ext*.define('FWY.view.student.List', {

extend: 'Ext.grid.Panel',

alias: 'widget.studentlist',

store: 'Students',

title: '学生信息列表',

initComponent: function () {

this.columns = [

{header: '编号', dataIndex: 'id', flex:1},

{header: '姓名', dataIndex: 'name', flex:1},

{header: '年龄', dataIndex: 'age', flex:1},

{header: '性别', dataIndex: 'sex', flex:1}

];

this.callParent(arguments);

}

});

## controller 层 创建文件

*Ext*.define('FWY.controller.Students', {

extend: 'Ext.app.Controller',

views: [

'student.List',

'student.Edit'

],

stores: ['Students'],

models: ['Students'],

init: function () {

this.control({

'studentlist': {

itemdblclick: this.editStudent

},

'studentedit button[action = save]' : {

click: this.updateStudent

}

});

},

onPanelRendered:function() {

console.log("panel rendered!");

},

updateStudent: function(button) {

*// 获取window 下面的 下面的 按钮， 提交。*

var win = button.up('window'),

form = win.down('form'),

record = form.getReader()

},

editStudent: function (grid,record) {

*// 通过别名获得这个组件*

var view = *Ext*.widget('studentedit');

*// 这个对象向下查找 form 组件,自动赋值*

view.down('form').loadRecord(record);

}

});

## store 创建文件。

*Ext*.define('FWY.store.Students',{

extend: 'Ext.data.Store',

model:'FWY.model.Students',

data: [

{id:1,name:'zhangsan', age:18,sex:'boy'},

{id:2,name:'lisi', age:20,sex:'gril'}

]

});