

JAVA资格认证:利用ExtJS构建客户端三层初探Java认证考试

PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/571/2021_2022_JAVA_E8_B5

[_84_E6_A0_BC_c104_571345.htm](https://www.100test.com/kao_ti2020/571/2021_2022_JAVA_E8_B5_84_E6_A0_BC_c104_571345.htm) 在软件架构中，特别是B/S架构中，很流行使用三层架构（数据层、中间层、表现层）来建立软件。但这从来都是服务器端的专利。在传统的概念中，客户端的职责就只是用来对数据进行展现而已。如果说客户端也搞三层，确实是一样不可思议的事件。为什么要搞客户端三层？ExtJS这个框架是个功能很强大的框架，由于它给了开发者很大的自由度跟细粒度。开发者在开发的过程中，配置、使用一个组件需要编写许多的代码。工作量大不说，效率低，可重用性低却是致命的。所以，聪明的开发者，应该积极利用ExtJS里的面向对象功能，简化自己的工作。定制自己的组件。才能提高自己的效率。所以，除了在服务器端也搞三层，我们也可以在客户端搞三层，定制组件，简化工作。基本知识 Ext.ns()（或Ext.namespace()函数）。相信很少有人去使用这个函数，这个函数用以构建一个新的命名空间。概念跟C#的命名空间或Java中的包概念一样。观察源代码我们可以发现。实际上这个函数只是建立了一个跟命名空间一样的类而已。比如：Ext.ns(buzz.data.utility).，就会新建一个类，名称是buzz.data.utility。有了这个概念，我们就好办了。实例函数与静态函数，这两种函数的区别我就不讲了，主要讲讲如何在JS中实现。用例子说明或许会比较直观。 1

```
Ext.ns(test.cls). 2 test.cls = function() { 3 4 } 5 test.cls.staticShow =  
function() { 6 alert(这是静态函数). 7 } 8 9
```

```
test.cls.prototype.instanceShow = function() { 10 alert(这是实例函
```

```
数). 11 } 调用如下 : 1 var temp = new test.cls(). 2
temp.instanceShow() 3 test.cls.staticShow() 数据访问层 1
Ext.ns(buzz.data.utility). 2 buzz.data.utility.buildHttpStroe =
function() { 3 return new Ext.data.Store({ 4 proxy: new
Ext.data.HttpProxy() 5 }). 6 } 该层用于返回一个进行基本封装
的Store对象。 中间层 1 Ext.ns(buzz.user). 2 buzz.user =
Ext.data.Record.create([ 3 { name: PersonName, mapping: name,
type: string }, 4 { name: PersonID, mapping: id, type: int } 5 ]). //数
据模型 , 对应服务器端的数据模型 6 7 buzz.user.load =
function(url, loadedCallback) { 8 var st =
buzz.data.utility.buildHttpStroe(). 9 st.proxy.getConnection().url =
url. 10 st.reader = new Ext.data.JsonReader({ root: data },
buzz.user). 11 st.on(load, loadedCallback). 12 st.load(). 13 } //静态
方法 , 用以从服务器端返回数据 , 返回类型为buzz.user类型
, 参数url是要调用数据的地址 , loadedCallback是回调函数 中
间层部分 , 我们新建了一个buzz.user类 (命名空间) , 该类
直接创建自Record对象 , 该对象是定制的 , 与服务器端数据
模型对应的。注意使用mapping进行映射。而buzz.user.load方
法是一个静态方法。不需要实例化即可使用。 表现层部分 1
Ext.onReady(function() { 2 Ext.get(btn1).on(click, function() { 3
buzz.user.load(json1.ashx, callback). 4 }). 5 }). 6 7 function
callback(st, res, op) { 8 for (var i = 0. i &lt;. res.length. i ) { 9
alert(String.format({0},{1}, res[i].get(PersonID),
res[i].get(PersonName))). 10 } 11 } 12 回调函数三个参数 , st
: store对象 ; res : Record数组 ; op : 其它附加选项 该篇文章
只作抛砖引玉的作用。实际开发中 , 还需要更多考虑的地方
```

。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com