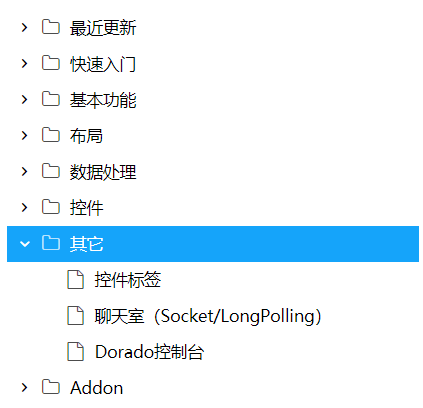
## 1、递归树

本例以系统菜单作为展现数据，用树的形式展现出来。由于DataSet的面向对象，以及DataType支持继承等特性，使得我们使用DataTree控件展现一个复杂结构和不同数据来源的数据都很轻松。



### 1）准备Java层服务

在Main.java中定义两个方法：

//根据父Category的ID获取子Category列表

//根据分类id获取Example对象列表

@DataProvider

**public** Collection<ExampleCategory> getCategories(Long parentCategoryId)

**throws** Exception {

List<ExampleCategory> categories = **new** ArrayList<ExampleCategory>();

List<Object[]> results = exampleCategoryDao

.find("select c, "

+ "(select count(\*) from ExampleCategory sc where sc.categoryId=c.id) as subCategoryCount, "

+ "(select count(e) from Example e, CategoryExampleRelation r where e.id = r.exampleId and r.categoryId=c.id) as subExampleCount "

+ "from ExampleCategory c where c.categoryId="

+ parentCategoryId + " order by c.sortFlag");

**for** (Object[] result : results) {

ExampleCategory category = EntityUtils

.*toEntity*((ExampleCategory) result[0]);

**int** childCount = (**int**) ((Long) result[1] + (Long) result[2]);

EntityUtils.*setValue*(category, "childCount", childCount);

EntityUtils.*setValue*(category, "hasChild", childCount > 0);

categories.add(category);

}

**if** (parentCategoryId == **null**) {

ExampleCategory category = **new** ExampleCategory();

category = EntityUtils.*toEntity*(category);

category.setId(*RECENT\_EXAMPLE\_CATEGORY*);

category.setLabel("最近更新");

category.setIcon("#fa fa-clock-o");

EntityUtils.*setValue*(category, "childCount", *RECENT\_EXAMPLE\_COUNT*);

EntityUtils.*setValue*(category, "hasChild", **true**);

categories.add(0, category);

}

**return** categories;

}

@DataProvider

**public** Collection<Example> getExamplesByCategoryId(Long categoryId)

**throws** Exception {

**if** (*RECENT\_EXAMPLE\_CATEGORY*.equals(categoryId)) {

Page<Example> page = **new** Page<Example>(*RECENT\_EXAMPLE\_COUNT*, 1);

exampleDao.find(page, "select e " + "from Example e "

+ "order by e.lastModify desc");

**return** page.getEntities();

} **else** {

**return** exampleDao.find("select e "

+ "from Example e, CategoryExampleRelation r "

+ "where e.id = r.exampleId and r.categoryId = "

+ categoryId + " order by r.sortFlag");

}

}

### 2）定义DataType

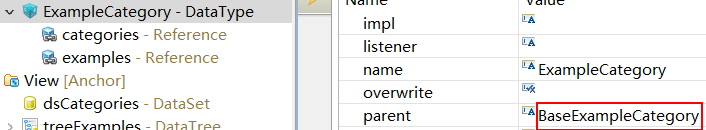
在视图模型中定义一个符合的DataType：ExmapleCategory：

<DataType name=*"ExampleCategory"* parent=*"BaseExampleCategory"*>

<Reference name=*"categories"* dataProvider=*"main#getCategories"* dataType=*"[SELF]"* parameter=*"$${this.id}"*/>

<Reference name=*"examples"* dataProvider=*"main#getExamplesByCategoryId"* dataType=*"[BaseExample]"* parameter=*"$${this.id}"*/>

</DataType>



其中定义了连个复杂的reference属性：

categories：是一个ExampleCategory组合的数组对象，其数据来自

main#getCategories

examples：是一个BaseExample组合的数组对象，其数据来自

main#getExamplesByCategoryId

注意这两个reference设置中parameter的设置：

动态EL表达式：$${this.id}

以当前ExampleCategory的id属性作为两个reference的dataProvider调用时的参数传入。

### 3）定义DataSet

定义dsCategory数据集：

<DataSet id="dsCategories"

dataProvider="main#getCategories" dataType="[ExampleCategory]"/>

通过这种定义方式，我们拿到dsCategories中的一个ExampleCategory对象时，其categories属性就是一个子的ExampleCategory集合；同时所属这个Category的Example集合就会自动初始化到examples中。

### 4）定义DataTree

treeExamples绑定了dsCategories对象，并通过BindingConfig定义节点属性。

<DataTree id=*"treeExamples"* dataSet=*"dsCategories"*

height=*"400"* width=*"300"*>

<BindingConfigs>

<BindingConfig childrenProperty=*"categories"* labelProperty=*"label"* recursive=*"true"*>

<Property name=*"iconClass"*>fa fa-folder-o</Property>

<Property name=*"expandedIconClass"*>fa-folder-open-o</Property>

<BindingConfig childrenProperty=*"examples"* labelProperty=*"label"*>

<Property name=*"iconClass"*>fa fa-file-o</Property>

</BindingConfig>

</BindingConfig>

</BindingConfigs>

</DataTree>

childrenProperty=”categories”，表示节点通过ExampleCategory的categories获取。

childrenProperty=”examples”，表示节点通过ExampleCategory的examples获取。

labelProperty=”Label”，表示树节点初始化时，默认以树节点所绑定对象的那个属性（本例为Label）作为节点标题显示。

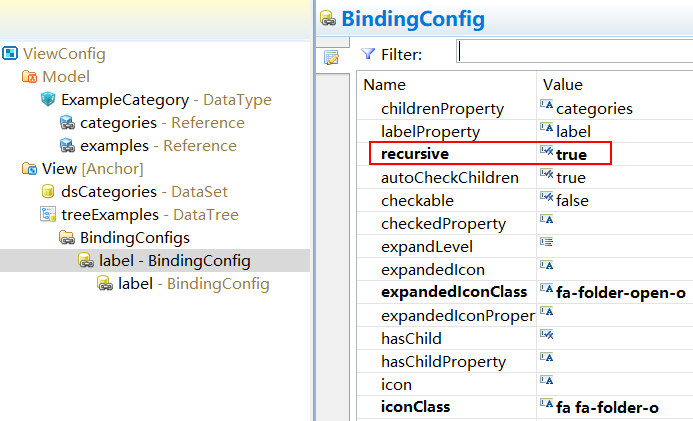
recursive=”true”，用来定义是否一个递归的BindingConfig，如果递归则内部会继续按照如下的规则加载树节点：

<BindingConfig childrenProperty="categories" labelProperty="label"

recursive="true" icon=">images/folder.gif">

<BindingConfig childrenProperty="examples" labelProperty="label" icon=">images/file.gif" />

</BindingConfig>



## 2、树形列表

3、与数据树联动

4、操作树节点

5、预处理树节点

6、虚拟根节点

7、查找树节点