开发企业信息维护页面CompanyManage。掌握Dorado7和数据库进行交互，单表中的一条记录的修改与保存。

需求：加载页面后，点击[编辑]按钮，编辑企业信息后，点击[保存]按钮，保存修改信息。

效果如下：



## 1、准备工作

## 2、开发步骤

### 2.1创建视图并添加控件

1）创建com.bstek.dorado.sample.standardlesson.junior.base包，

新建CompanyManage.view.xml。

2）新建并配置DataType

[Model]节点下添加DataType并设置属性：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| name | SlCompany | DataType的名称 |
| parent | global:SlCompany | 继承models目录中Common.model.xml中定义的Company |

（DataType支持继承：包括多重继承。）

开发中还需要对[企业简称]和[企业全称]设置数据校验器，数据校验是基于DataType的，因此可以在DataType下的PropertyDef下添加这些Validator。

在dataType下添加两个PropertyDef并设置属性：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| name | companyName | propertyDef的名称 |
| label | 企业简称 | propertyDef标签显示 |
| name | fullName |  |
| label | 企业全称 |  |

为[companyName]添加[RequiredValidator]，为[fullName]添加[LengthValidator]并设置minLength为1，maxLength为50。

3）新建并配置DataSet

在[View]节点下添加一个DataSet控件并设置属性：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| id | datasetCompany | 控件的Id |
| dataType | [SlCompany] | DataType的数据类型 |
| dataProvider | companyService#getCompany | 由companyService这个bean中的getCompany方法为DataSet提供数据服务 |
| readOnly | true | 设置DataSet数据只读 |

（注：在dataType的属性设置上，需要选择当前View中私有的DataType）

4）新建并配置AutoForm

[View]节点下添加AutoForm控件并设置属性：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| id | autoformCompany | 控件的id |
| dataSet | datasetCompany | 通过dataSet属性告知AutoForm绑定是哪些数据，以便在AutoForm中展示和处理 |
| cols | \* | AutoForm列显示布局表达式 |
| width | 500 | 控件的宽度 |
| labelSeparator | : | AutoForm Element中标签和输入框间的分隔符 |

然后右键autoformCompany控件选择[Generate AutoForm Elements]，生成相关字段，删除companyId和多余的字段。

5）新增并配置UpdateAction控件

如果想做持久化处理，还需借助DataResolver技术。DataResolver与DataProvider类似，也是一个Ajax请求，只是作用与DataProvider相反，DataProvider用于从服务器端获取数据存放在客户端的DataSet中，而DataResolver技术却是为了将客户端的DataSet中的数据提取出来，并通过AJAX技术上传到服务器端，便于交给后台的Dao做持久化的处理工作。（在Dorado7中DataResolver一般都是通过UpdateAction控件来配置的）

设置UpdateAction属性：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| id | updateactionSave | 控件的Id |
| dataResolver | companyService#saveCompany | 通过执行companyService这个bean中的saveCompany方法对传递到后台的数据进行持久化 |
| executeMessage | 正在保存... | 档次Action正在执行时希望系统显示给用户的提示信息 |
| successMessage | 保存成功！ | 当此Action执行成功之后自动提示的信息 |
| hotkey | Ctrl+s | 执行此Action的热键 |

然后为UpadateAction添加子节点UpdateItem，绑定需要提交的DataSet数据集：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| dataSet | datasetCompany | 实现DataSet数据采集，一个UpdateAction可以添加多个UpdateItem |

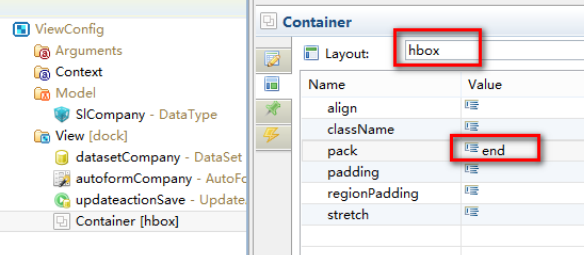
6）新增并配置Container控件

新增一个Container控件用于放置后面添加的2个按钮控件：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| width | 500 | 设置容器控件的宽度 |
| height | 50 | 设置容器控件的高度 |

并设置容器的布局属性：

Layout设置为hbox，pack属性设置为end。



hbox横向布局中的pack属性表示子控件的对齐方向，有3个属性可选：start，center，end，分别表示按照开始位置对齐，居中对齐和按照最后的位置对齐。

7）添加并配置按钮

在Container控件下添加2个Button控件：

为第1个Button设置属性id为buttonEdit，caption为编辑，绑定onClick事件，传入参数datasetCompany，添加如下代码：

datasetCompany.set(“readOnly”,false);

通过js将datasetCompany的只读属性设置为否，因此就能编辑数据。

设置第2个Button的属性：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| id | buttonSave | 控件的Id |
| caption | 保存 | 控件显示的标题 |
| action | updateactionSave | 通过绑定UpdateAction，点击按钮便会执行IpdateAction操作。 |
| tip | 使用Ctrl+s快捷键保存 | 鼠标移到Button控件上的提示信息 |

### 2.2准备后台服务

在CompanyService类中增加保存方法：

@DataResolver

@Transactional

public void saveCompany(Collection<SlCompany> slCompanys) {

slCompanyDao.persistEntities(slCompanys);

}

@DataResolver用于声明这个方法为DataResolver方法，可以被客户端的DataResolver技术所调用，@Transactional是Spring提供的一个事务性的标记声明。（注：忘记加入@Transactional注解，带来的结果就是修改的数据无法commit进行持久化）

HibernateDao的persistEntities方法代码如下：

@SuppressWarnings("unchecked")

public int persistEntities(Collection<T> entities) {

int i = 0;

for (Object entity : EntityUtils.getIterable(entities, FilterType.DELETED)) {

delete((T) entity);

i++;

}

for (Object entity : EntityUtils.getIterable(entities, FilterType.MODIFIED)) {

save((T) entity);

i++;

}

for (Object entity : EntityUtils.getIterable(entities, FilterType.MOVED)){

save((T) entity);

i++;

}

for (Object entity : EntityUtils.getIterable(entities, FilterType.NEW)) {

save((T) entity);

i++;

}

return i;

}

在方法中，根据提交到后台数据不同的状态，执行不同的持久化操作，并返回操作过的数据量。

### 2.3查看运行效果

<http://localhost:8080/standard-lesson/com.bstek.dorado.sample.standardlesson.junior.base.CompanyManage.d>



## 3、小结

* UpdateAction的使用
* RequiredValidator，LengthValidator的使用
* 掌握后台数据状态
* Dorado数据持久化机制