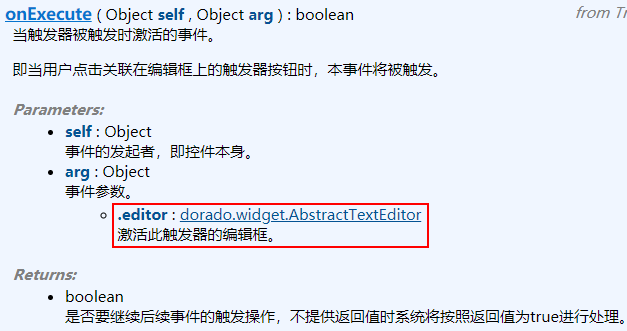
## 1、编辑框和下拉框（）

本例采用了FormElement代替TextEditor进行定。FormElement是Dorado7特别提供的表单元素：表单元素是一种组合式的控件。一般而言，一个表单元素包含文本标签、编辑器、提示信息三个部分。

### 1）自定义的trigger

trigger1绑定onExecute（当触发器被触发时激活的事件）事件：

arg.editor.set(“text”,”Trigger设置的值”);



### 2）自定义的Trigger2

基于第一个编辑框的使用机制，我们知道可以在trigger的onExecute自定义自己的逻辑。则可以在trigger2中写入如下代码：

var dialog = view.get(“#dialog2”);

dialog.set(“userData”,arg.editor); //设置编辑框

dialog.show();

为dialog里的button绑定onClick事件：

var dialog = self.findParent(dorado.widget.Dialog);

var editor = dialog.get("userData"); //取得对应的编辑框

editor.set("text", "Trigger设置的值");

dialog.hide();

### 3）自定义的Trigger3

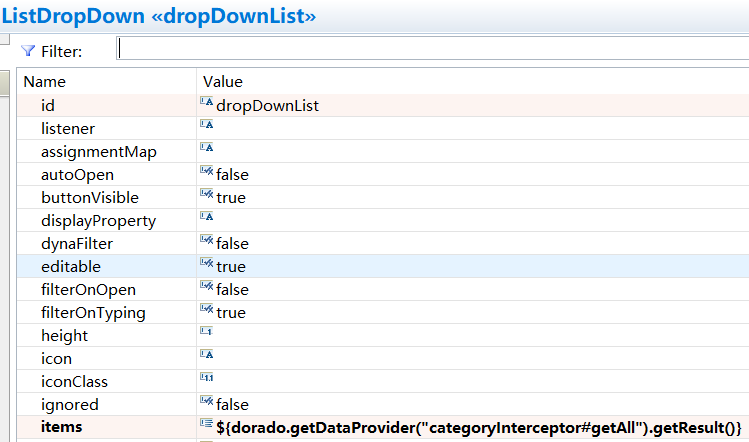
Trigger除了自定义的逻辑之外，还可以自由定义它的图标，它提供了icon属性，并为trigger绑定onExecute事件：

view.$onMathTriggerExecute(self, arg);

### 4）下拉框

普通列表下拉框：

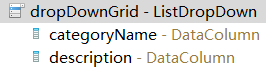
DataSetDropDown也是一种Trigger，可以在View中定义，然后在通过trigger属性进行关联。



其中items属性时Dorado的一个内置EL表达式，用于激活DataProvider机制，定位到categoryInterceptor服务类的getAll方法。这个方法返回一个Category的列表。

包含Grid的下拉框：

可以将一个Grid也放在下拉框中进行trigger绑定。



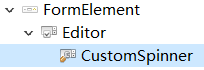


此方法返回一个Category的列表。

### 5）格式化输入编辑框

IP地址编辑框：

在应用中存在很多有格式化输入的编辑框，希望在这种编辑框在我们输入的时候就进行格式限定，如IP地址，日期时间，邮件，网址等。



通过CustomSpinner控件（自定义微调编辑器）完成格式化的控制。设置其属性：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| pattern | IP: [0,255].[0,255].[0,255].[0,255] | 样式表达式，基本语法是利用[]表示一个微调槽。  []中可以放置该微调槽的取值范围 |
| value | 192,168,18,223 | 当前各槽数值的数组 |



日期时间编辑框：

添加DateTimeSpinner控件，设置其type属性为dateTime。



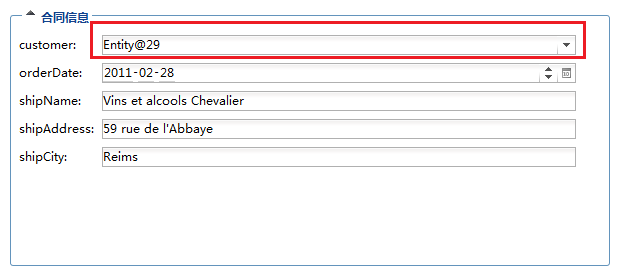
## 2、虚拟属性下拉框（）

## 3、引用实体下拉选择（）

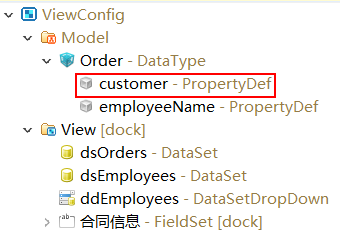
如果DataType的一个propertyDef对应的是一个数据实体，而不是普通的数据类型，那么在Grid或在AutoForm中如何显示，以及如果还要支持下拉框选择输入，该通过什么方式定义。

### 1）PropertyDef为实体对象在Grid或AutoForm中如何显示

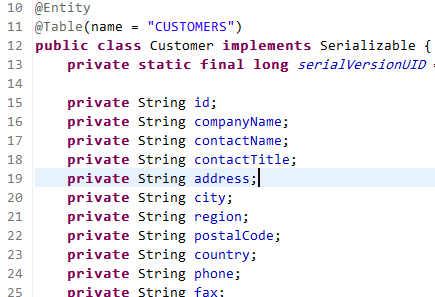
一个PropertyDef为实体对象的时候，默认在Grid或AutoForm中显示的效果如：



其PropertyDef的定义为：



Customer.java

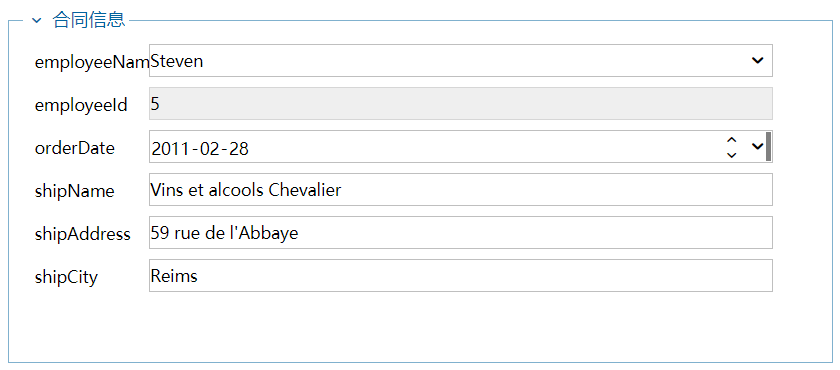


如果希望在Autoform或Grid中显示Customer的companyName属性，则常规方法有：利用虚拟属性，给DataType建一个新的PropertyDef，命名为companyName，并设置这个propertyDef的property属性为customer.companyName。另外AutoForm中删除customer，添加与companyName对应的AutoFormElement：



有时候由于业务逻辑的需要，需要直接面向这个实体对象编程，则我们可以利用DataType的defaultDisplayProperty属性实现：  


这样页面上就可以直接显示companyName的值了：

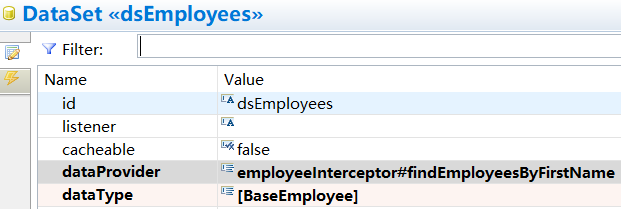


### 2）PropertyDef为实体对象实现下拉框选择输入如何实现

希望实体对象类型的PropertyDef允许下拉选择输入，并且希望最终在浏览器端建立两者下拉的一个实体对象。

定义DataSet：

采用DataSetDropDown实现下拉选择，那么首先创建一个EmployeeName相关的DataSet：dsEmployees



利用DataSet提供的dataProvider技术，提供下拉框的动态数据。

@DataProvider

**public** Collection<Employee> findEmployeesByFirstName(String namePattern) {

**if** (StringUtils.*isEmpty*(namePattern)) {

**return** employeeDao.getAll();

} **else** {

**return** employeeDao.find(Restrictions.*like*("firstName",

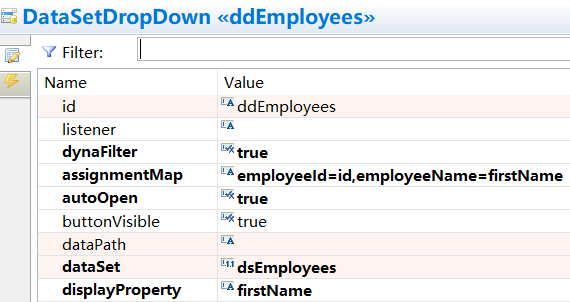
namePattern,MatchMode.*ANYWHERE*));

}

}

DataSetDropDown的定义：

添加DataSetDropDown，dataSet属性绑定到前面创建的dataSet。

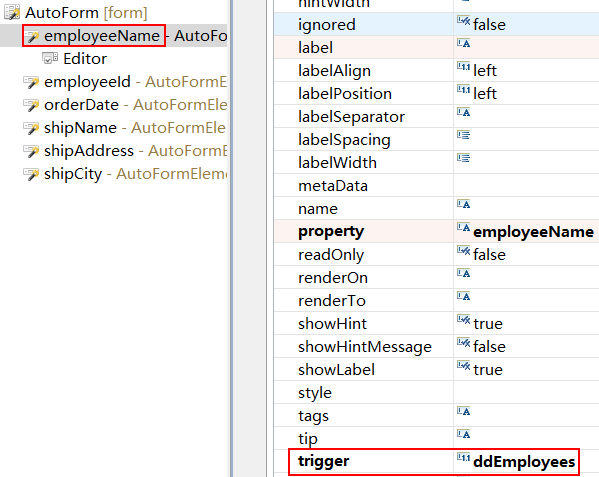


设置DataSetDropDown的属性：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| autoOpen | true | 当相应的编辑框获得输入焦点时自动打开此下拉框 |
| dynaFilter | true | 是否启用动态数据过滤功能 |
| assignmentMap | employeeId=id,  employeeName=firstName | 用于设定当用户选中某个条目之后，系统应该进行怎样的赋值操作 |
| diaplayProperty | firstName | 数据实体中的某个属性名 |
| dataSet | dsEmployees | 绑定的数据集 |

下拉框的绑定：

如果希望在emplyoeeName编辑框中提供下拉框，则可以利用dorado中的trigger技术进行下拉框的绑定。



### 3）关于DataType的defaultDisplayProperty

在下拉框的定义中，dorado中引入了ORM技术，则下拉框中列出的数据以及我们选择操作所面对的都是数据实体而不是简单的键值。

本例的DataSetDropDown通过employeeInterceptor#findEmployeesByFirstName这个DataProvider最终获得的是一些数据实体。那在本例中我们只希望展现该实体中的部门名称又该怎么办呢？我们注意到dsCustomers对应的DataType为[BaseCustomer]，在本例中我们是通过DataType的defaultDisplayProperty解决这个问题的。

### 4）dynaFilter（动态数据过滤）

数据下拉框还具有过滤查询功能，通过DataSetDropDown的dynaFilter属性进行定义，当我们设置该属性为true时，系统就会自动的将编辑框输入的值自动传输到DataSet对应dataProvider的数据请求方法中，也就是会重新调用employeeInterceptor#findEmployeesByFirstName，并将编辑框中的值作为参数传入。本例中根据智能方法适配机制，会自动调用com.bstek.dorado.sample.interceptor.EmployeeInterceptor类的findEmployeesByFirstName方法。

@DataProvider

public Collection<Customer> findCustomersByCompanyName(String namePattern) {

if (StringUtils.isEmpty(namePattern)) {

return customerDao.getAll();

}

else {

return customerDao.find(Restrictions.like("companyName",

namePattern, MatchMode.ANYWHERE));

}

}

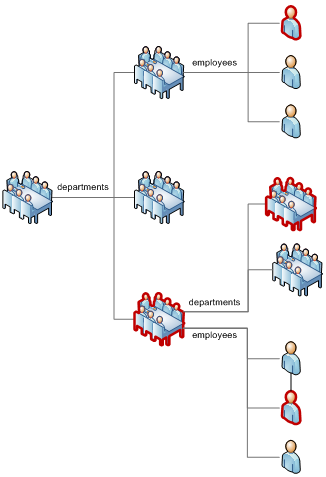
其中我们输入的过滤值会通过namePattern传入，这样在该方法内部根据上下文环境我们对employeeName做一个过滤查询。

## 3、联动下拉框（）

如下视图，产品分类和产品：



Reference类型的联动下拉框是指针对立体数据模型而言的，如上视图如果将产品分类做到一个数据模型中就是一颗属性结构，如下的一个立体数据模型：



如果将产品分类转换为department，而产品转换为employee，则当选择第一个部门的时候，下面的下拉框可以显示该部门的员工：

由于Dorado7中我们可以通过一个DataSet装载以上的这个立体数据结构，下面我们可以利用立体数据模型特有的机制，快速实现联动下拉框（当然有一定的局限性）首先按照DataType立体数据模型的定义方式定义好DataType，如下：