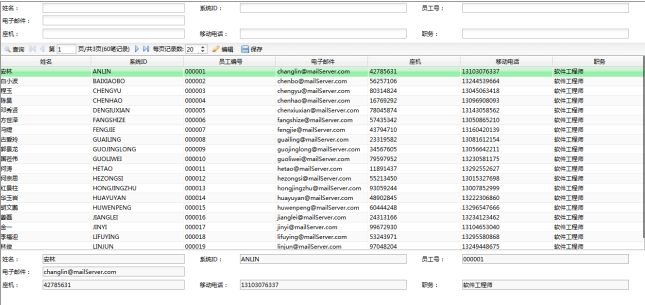
开发企业员工信息维护页面即EmployeeManage页面。掌握Dorado7和数据库进行交互，单表中的多条记录的修改与保存。

需求：

加载页面后，在查询框中输入查询条件，点击[查询]按钮，在表格控件中显示查询结果，在页面最下方的表格控件中联动显示员工信息。

点击[编辑]按钮，编辑员工信息后，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员工信息查询

1）创建视图并添加控件

在com.bstek.dorado.sample.standardlesson.junior.base包下新建

EmployeeManage.view.xml

在[Model]节点下添加一个DataType，设置其name属性为SlEmployee，parent使之继承公共的DataType。

在DataType下新增4个PropertyDef并设置属性：

|  |  |  |
| --- | --- | --- |
| PropertyDef | name | label |
| propertyDef1 | employeeCode | 员工编号 |
| propertyDef2 | userName | 系统Id |
| propertyDef3 | employeeName | 姓名 |
| propertyDef4 | email | 电子邮件 |

分别加入校验器：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段说明 | 数据校验器 | 说明 |
| employeeCode | 员工编号 | LengthValidator | 文本长度数据验证器 |
| userName | 系统Id | RequiredValidator | 必填数据验证器 |
| employeeName | 姓名 | RequiredValidator | 必填数据验证器 |
| email | 电子邮件 | RegExpValidator | 正则表达式数据验证器 |

LengthValidator

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| minLength | 1 | 允许输入字符最小长度 |
| maxLength | 6 | 允许输入字符最大长度 |

RegExpValidator

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| whiteRegExp | \w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+  )\* | RegExpValidator的whiteRegExp（白表达式） |

在[View]节点下添加一个DataSet控件并设置属性：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| id | datasetEmployee | 控件的id |
| dataType | [SlEmployee] | DataSet的数据类型 |
| dataProvider | employeeService#  queryForCondition | 由employeeService这个bean中的queryForCondition方法为DataSet提供数据加载服务 |
| readOnly | true | 设置DataSet数据只读 |
| pageSize | 15 | 装载数据时使用的分页大小，即按照每页多少条记录来进行分页装载 |

在[View]节点中国添加AutoForm控件并设置属性：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| id | autoformCondition | 控件的Id |
| createPrivateDataSet | true | 是否要在AutoForm没有实际绑定DataSet时自动创建一个私有的DataSet用于管理表单数据 |
| cols | 400,\*,\* | AutoForm列显示布局表达式 |
| labelSeparator | : | AutoForm Element中标签和输入框间的分隔符 |

由于Dorado中有很多类似于数据校验、键值映射这样的功能都依赖于DataSet，所以如果仅仅使用简单的Entity绑定，AutoForm必然是失去一些功能特性。启用本属性可以让AutoForm自动创建一个私有的DataSet，以便于使用上述各项功能。对于createPrivateDataSet为true的AutoForm，如果我们通过如下代码获取Entity对象，则它是被这个私有的DataSet所管理的实体对象。

var entity = autoForm.get(“entity”);

在autoformCondition中新建7个AutoForm Element并设置属性：

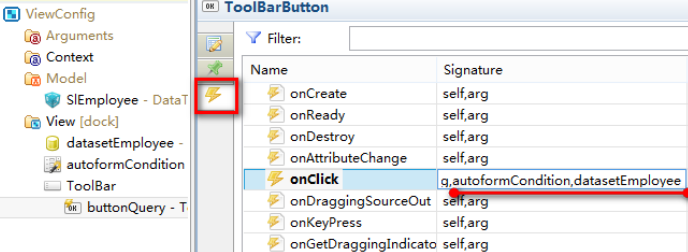
|  |  |  |
| --- | --- | --- |
| AutoForm Element | name | label |
| AutoForm Element1 | employeeName | 姓名 |
| AutoForm Element2 | userName | 系统ID |
| AutoForm Element3 | employeeCode | 员工号 |
| AutoForm Element4 | email | 电子邮件 |
| AutoForm Element5 | phone | 座机 |
| AutoForm Element6 | mobile | 移动电话 |
| AutoForm Element7 | position | 职务 |

将email字段的colSpan属性设置为3，width设置为400

在[View]节点下添加ToolBar控件（ToolBar是工具栏控件，同时也是一个容器控件，一般情况下ToolBar都是用以统一管理界面上的功能按钮或菜单项），接下来在工具栏中加入查询按钮[ToolBarButton]并设置属性：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| id | buttonQuery | 控件的Id |
| caption | 查询 | Button显示的标题 |
| icon | url(>skin>common/icons.gif) -20px -40px | 设置Button的图标 |

对[buttonQuery]绑定onClick事件，即在输入查询条件后，点击按钮，对数据结果进行查询。在事件上增加autoformCondition和datasetEmployee两个参数：



代码如下：

//获取autoformCondition绑定的实体对象

var entity = autoformCondition.get(“entity”);

//将实体对象作为参数传入，并异步刷新

datasetEmployee.set(“parameter”,entity).flushAsync();

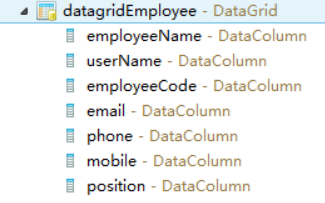
接下来在[ToolBar]下添加分页控件[DataPilot]，并设置属性：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| id | datapilotEmployee | 控件Id |
| dataSet | datasetEmployee | 通过dataSet属性告知DataPilot绑定是哪些数据，以便在DataPilot中进行分页和处理 |
| itemCode | pages,pageSize | 用于定义数据导航条中要显示那些子对象的表达式。  pages用于简单翻页按钮定义的特殊代码，表示|<,<,goto,>,>|这5个子对象的组合。  pageSize 显示和指定每页记录数的编辑框。 |

在[View]节点下添加[DataGrid]控件并设置属性：

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 说明 |
| id | datagridEmployee | 控件的Id |
| dataSet | datasetEmployee | 通过dataSet属性告知DataGrid绑定是哪些数据，以便在DataGrid中展示和处理 |
| height | 350 | 设置控件的高度 |

右键[DataGrid]控件生成相关字段，调整字段显示顺序并删除不需要的字段：



在[View]节点下

### 2.2员工信息维护

## 3、小结