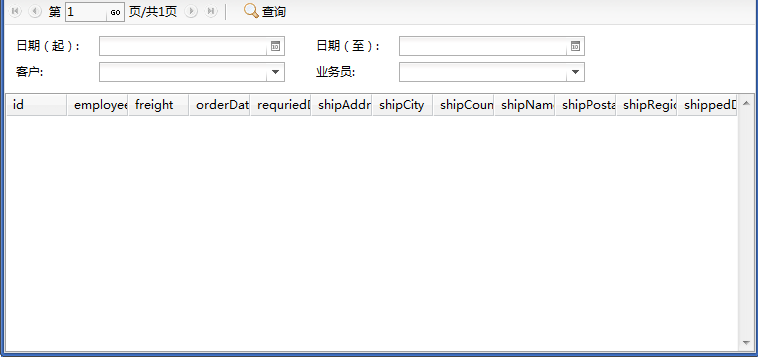
一个具有简单查询功能的页面，且具有数据导航操作功能，包括翻页。



## 准备后台数据服务

@DataProvider

public void query(Page<Order> page, Map<String, Object> parameter) {

Criteria criteria = orderDao.createCriteria();

if (parameter != null) {

if (parameter.get("orderDate1") != null) {

criteria.add(Restrictions.ge("orderDate", parameter

.get("orderDate1")));

}

if (parameter.get("orderDate2") != null) {

criteria.add(Restrictions.le("orderDate", parameter

.get("orderDate2")));

}

if (parameter.get("customer") != null) {

criteria.add(Restrictions.eq("customer", parameter

.get("customer")));

}

if (parameter.get("employee") != null) {

criteria.add(Restrictions.eq("employee", parameter

.get("employee")));

}

}

orderDao.find(page, criteria);

}

### page参数

要实现翻页功能，方法中需要包含一个Page参数，这是dorado的provider提供的一个工具类。把数据初始化到page对象中即可，通过orderDao进行初始化。

### parameter参数

实现查询功能，本例设计了一个Map类型的对象parameter，查询时只要将相关参数放置在parameter中即可。

## 定义数据展现控件

利用DataGrid展现查询后的数据：

<DataSet id="dsOrders" dataType="[Order]"

dataProvider="orderInterceptor#query"

pageSize="10" loadMode="manual" />

<DataGrid dataSet="dsOrders" autoCreateColumns="true"

stretchColumnsMode="stretchableColumns" />

此处定义一个绑定到dsOrders上的DataGrid对象。dsOrders的dataProvider就是绑定到前面定义好的查询方法。

## 定义查询元素

配合查询，在页面上定义一个表单元素，便于用户输入查询条件。

最后需要定义一个按钮收集这个AutoForm中用户输入的信息，并作为query方法的参数，发起query查询请求：

// @Bind #btnQuery.onClick

!function(){

var formCondition = view.get("#formCondition");

var dsOrders = view.get("#dsOrders");

dsOrders.set("parameter", formCondition.get("entity"));

dsOrders.flushAsync(); //以异步方式重新装载实体中的数据

}

AutoForm对象formCondition，通过.get(“entity”)方法，会自动返回一个JSON对象。将这个JSON对象直接赋值给dsOrders的parameter参数，并调用dsOrders的flushAsync激活Java服务层定义好的query方法，执行查询。

## 其他说明

为了翻页和数据控制操作，本例还添加了ToolBar和DataPilot对象。

另外为了查询条件的输入方便，在AutoForm中的客户端和业务员编辑器上都定义了trigger属性（触发），这就是这个视图模型中两个DropDown（下拉框）和DataSet存在的原因。

DropDown：

<DataSetDropDown id="ddCustomers"

dataSet="dsCustomers" dynaFilter="true" />

<DataSetDropDown id="ddEmployees" dataSet="dsEmployees" />

DataSet：

<DataSet id="dsCustomers" dataType="[BaseCustomer]"

dataProvider="customerInterceptor#findCustomersByCompanyName" />

<DataSet id="dsEmployees" dataType="[BaseEmployee]"

dataProvider="employeeInterceptor#getAll" />