

# 喝喝茶 编编程

使用 dorado 基础技术进行 Web 应用开发

**BSTEK**

**2007 年 10 月**

## 作者介绍



[Cavin.wang@bstek.com](mailto:Cavin.wang@bstek.com) 左二，[Thomas.ding@bstek.com](mailto:Thomas.ding@bstek.com) 左一。

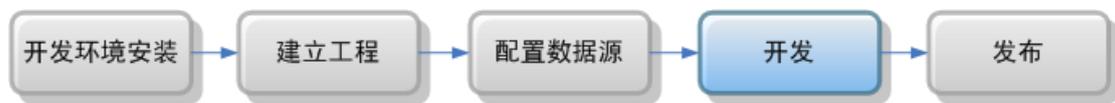
## 目录

1. 前言.....	4
1.1. 图例.....	4
1.2. 准备工作.....	5
dorado 原理: dorado 的三层开发模式与多层开发模式.....	6
2. 第一杯茶: 单表展现.....	9
2.1. 页面效果.....	9
2.2. 创建视图模型.....	9
2.3. 配置数据来源.....	11
2.4. 配置展现形式.....	12
2.5. 配置数据源与控件的关系.....	14
2.6. 创建 JSP.....	14
2.7. 单表开发口诀.....	20
dorado 原理: 数据展现的运行时.....	21
dorado 原理: 迭代式的 BRICH-MVC 架构.....	23
3. 第二杯茶: 页面调整.....	24
3.1. 添加页面导航条 PagePilot.....	24
dorado 原理: dorado 开发中 JSP 的作用.....	26
3.2. 设置分页.....	27
4. 第三杯茶: 单表的增改删存.....	30
4.1. 准备工作.....	30
4.2. 增加记录.....	30
4.3. 修改记录.....	32
4.4. 删除记录.....	35
4.5. 保存修改.....	36
dorado 原理: dorado 中的 commands.....	39
dorado 原理: dorado 中的保存.....	40
dorado 原理: 保存的运行时.....	41
5. 第四杯茶: 单表的查询.....	44
5.1. 准备工作.....	44
5.2. 设置匹配条件.....	44
5.3. 添加条件数据集.....	48
5.4. 添加用于条件输入的表单.....	49
5.5. 添加查询命令.....	50
5.6. 添加查询按钮.....	51
5.7. 改进.....	51
dorado 原理: dorado 中的查询.....	53
dorado 原理: dorado 查询的运行时态.....	55

# 1. 前言

初学游泳，纵身入水、冲刺触壁这些头尾的关键技术可以放在后面，先掌握换气划水这些基础动作，更前一些可以通过戏水了解水性。

这篇 dorado 技术文档没有涉及建立工程、数据源配置、打包发布等开头结尾的技术，重点介绍把数据库中的一张表与数张表在 Web 页面上进行展示与增改删查存等基本操作，中间穿插介绍了一些 dorado 的基本概念与开发技巧。



这篇文档的阅读群设想为第一次接触 dorado 的开发者，或者是 dorado 的销售人员，茶余饭后可以浏览这篇文档以快速掌握 dorado。本文的名称定为“喝喝茶 编编程”也是为了体现 dorado 的产品宣传口号之后 “In my spare time, I also design web application”



*Change Spare Time Style of  
Developers*

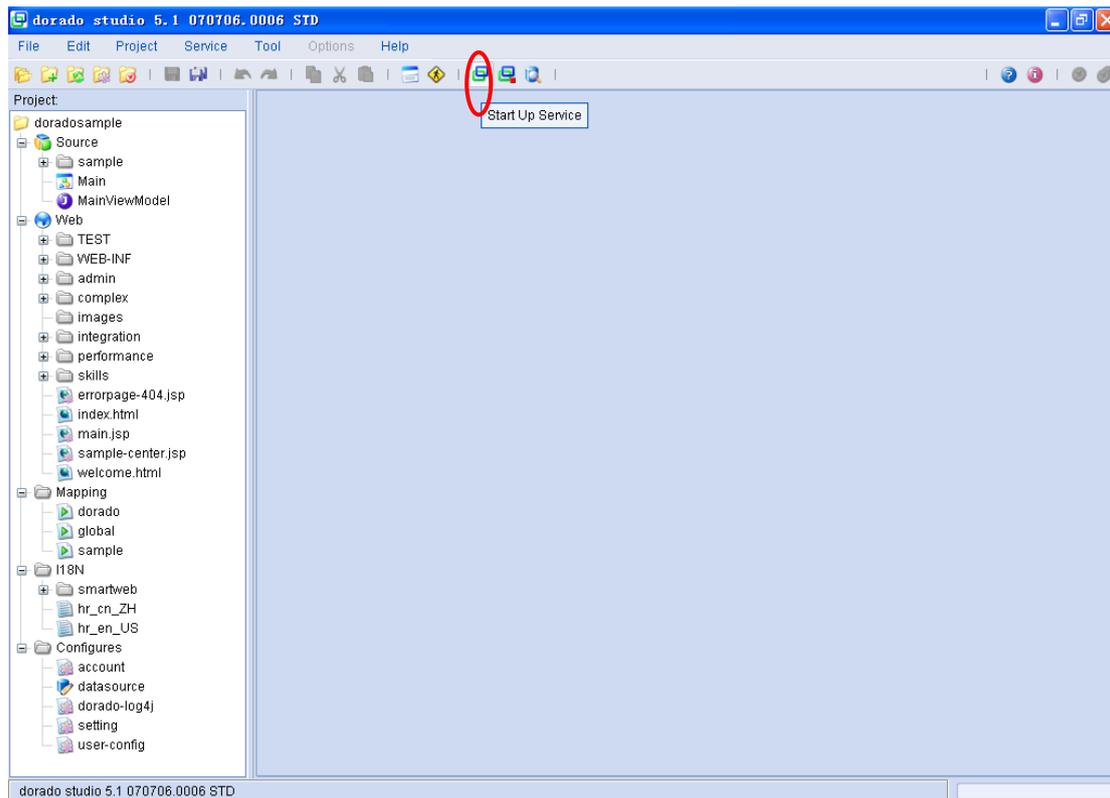
## 1.1. 图例

序号	图例	含义
No.1		dorado 基本概念
No.2		开发心得与小技巧
		dorado 开发时序
		dorado 运行时序

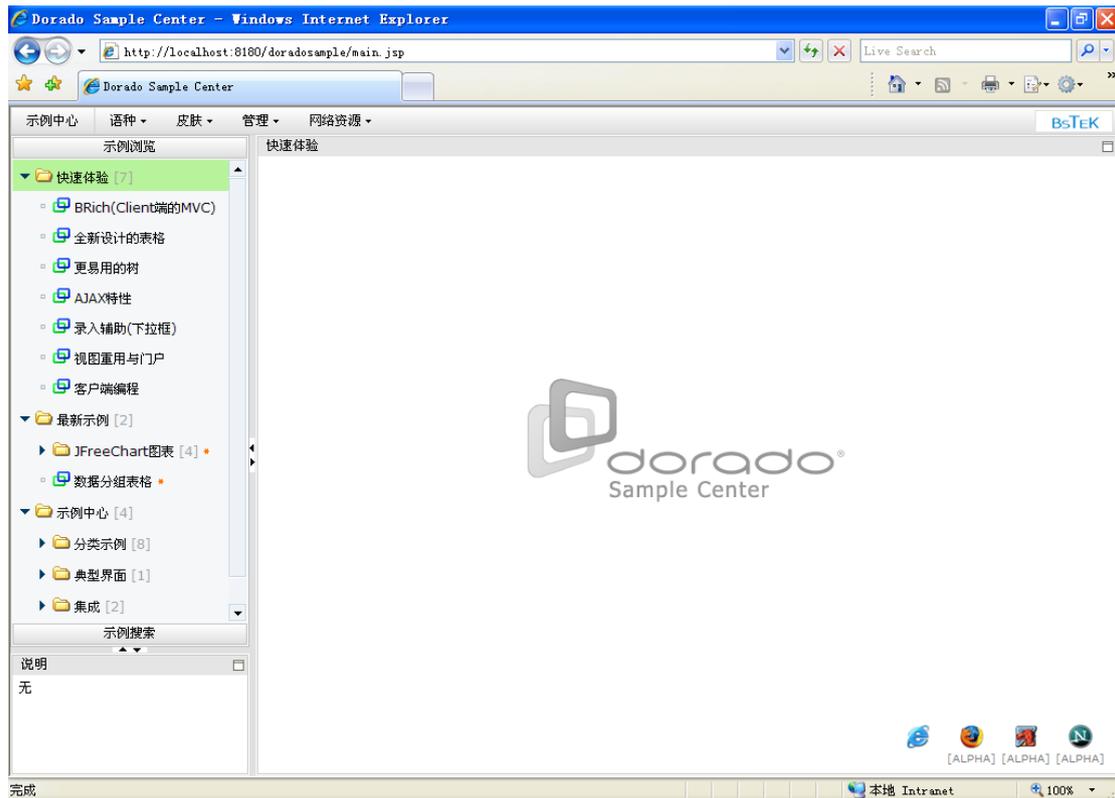
## 1.2. 准备工作

下载 dorado 安装包，解压到本地目录，目录中不能有中文也不能有空格，合格的目录如“D:/java/develop/dorado5”。JAVA\_HOME 需要已经配置好了。运行 dorado 5 开发包中的 setup.bat 文件进行初始化，最后选择 y 并按回车键确认打开 studio。

可以参见详细的安装说明《dorado5 安装指南-v10-20061106.doc》。



点击 Start Up Service 后，启动 dorado SampleCenter。



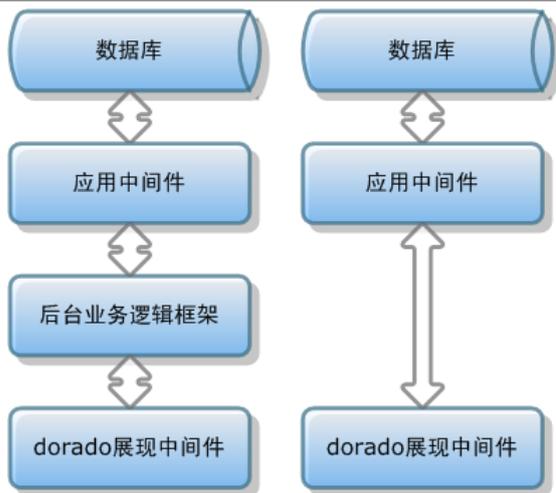
Dorado 示例中心的顺利启动标志开发环境已就绪，我们可以在 dorado Sample 工程的基础上体验基于 dorado 的开发过程。

## dorado 原理：dorado 的三层开发模式与多层开发模式



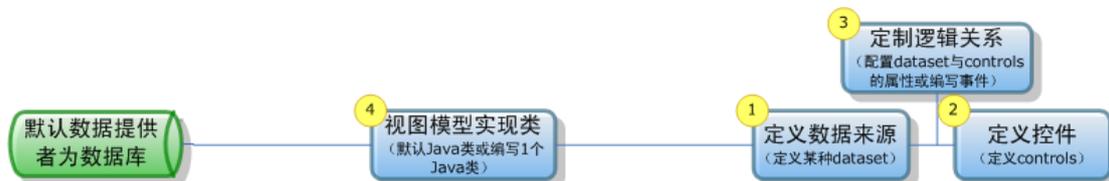
### dorado 的三层开发模式与多层开发模式

dorado 的开发套路大致分为两种，一种是通过 JDBC 连接数据库的三层开发模式，即 dorado + 应用中间件 + 数据库，一种是通过后台业务逻辑框架连接数据源的多层开发模式，即 dorado + 后台业务逻辑框架 + 应用中间件 + 数据库。



后台业务逻辑框架可以是数据访问层、业务对象层或控制层的框架（如 Struts、Webwork、Spring、Hibernate、iBatis 等），可以为流程引擎（如 WfMC 标准、BPEL 标准等或自有标准的流程引擎），也可以为软件开发商自有框架，或者是数据整合中间件、消息中间件、应用联接中间件、应用整合中间件等，同样可以由上述框架的一种或数种组成。

这里先从介绍三层开发模式开始，在这种模式下，代码管很少，以配置为主，之后再解释涉及“Java 代码写在哪里”的多层模式。前一种开发模式多用 AutoSqlDataset、SqlDataset，数据对象是 SQL 执行结果等，后一种开发模式多用 CustomDataset、MarmotDataset，数据对象是 POJO（Javabean/Map 的集合）等。



在三层开发模式下，数据提供者默认为数据库，dorado 通过视图模型实现类与其进行交互。在对单表操作等简单的需求场景下，这个视图模型实现类是系统默认生成的，在开发过程中无需关心。在多表复杂操作的需求场景下，需要编写这个 Java 类。



在多层开发模式下，数据提供者是业务逻辑层的一部分，dorado 需要构造一个胶合层来实现交互，一般需要编写这个 Java 类。此后的编程工作都是三板斧：定义数据源（即定义某种 dataset）、定义控件（定义 Controls）与定制逻辑关系（配置 dataset 与 controls 的属性

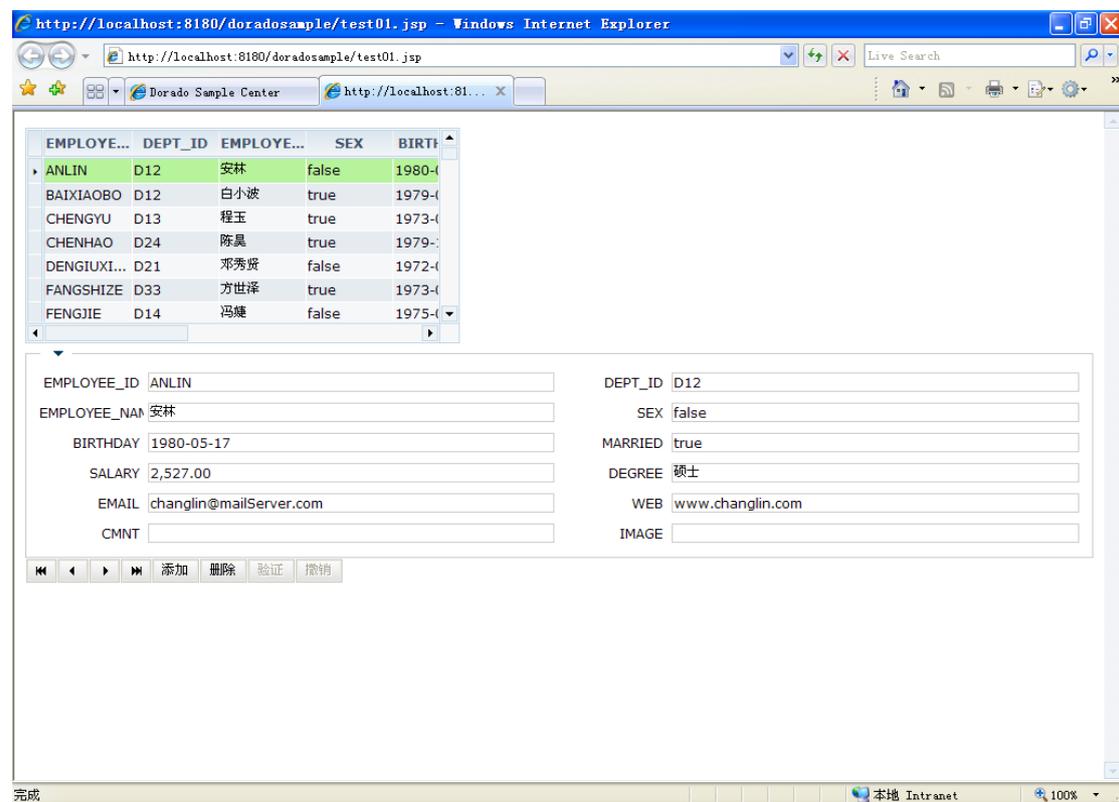
或编写事件)

因此在学习 dorado 时，建议在开始时采用三层开发模式，以集中精力快速掌握 dorado 自身的功能，在此基础上再进行多层架构的开发。

## 2. 第一杯茶：单表展现

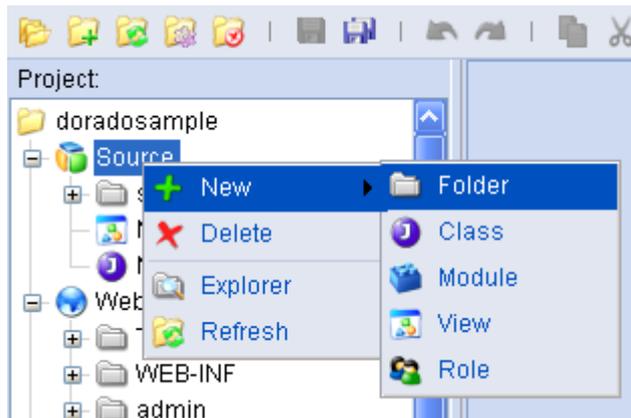
### 2.1. 页面效果

在 Web 页面上展现数据库中的一张表，使用到表格（Table）、表单（Form）与数据导航条（DataPilot），效果如下图。

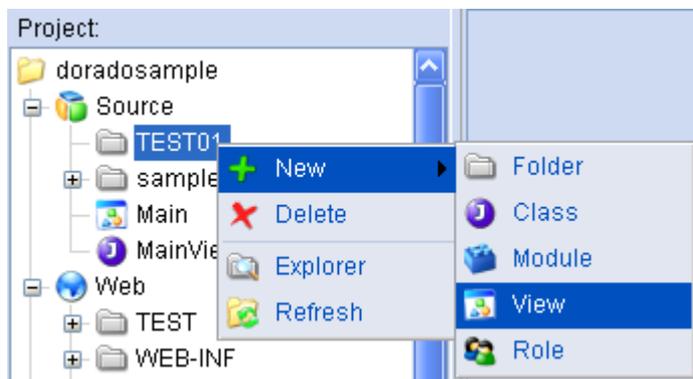


### 2.2. 创建视图模型

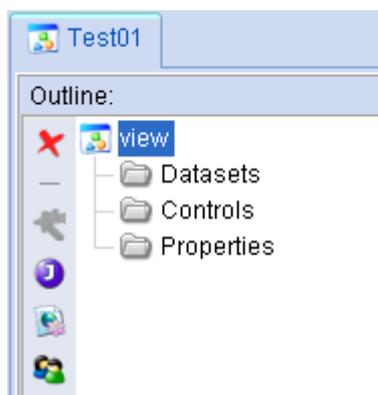
在左侧工程树（Project）的 Source 节点上右键新建文件夹“TEST01”。



在文件夹 TEST01 中新建 View “Test01”，视图模型种类（ViewModel Types）选择默认的“Common ViewModel”。

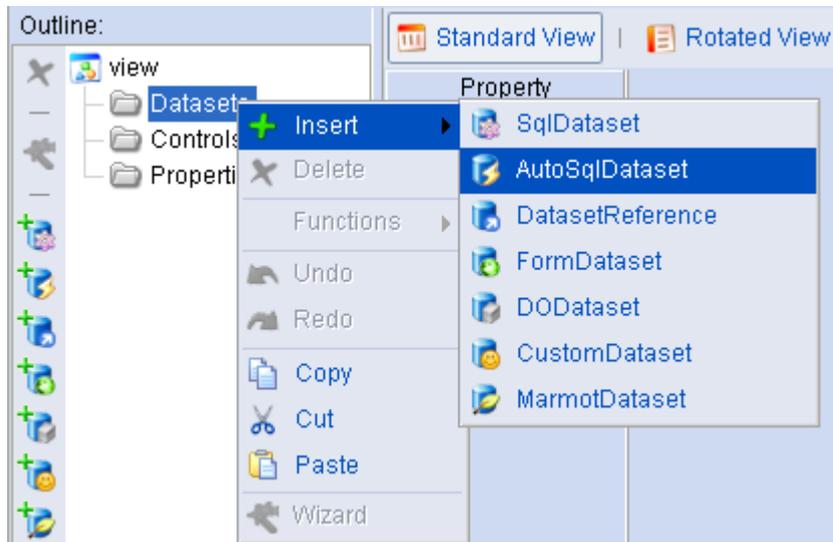


ViewModel 可以简单地理解为页面建模，这时我们看到在节点 View 下有三个子节点，分别是 Datasets、Controls、Properties，分别代表数据源、表现层控件与属性。

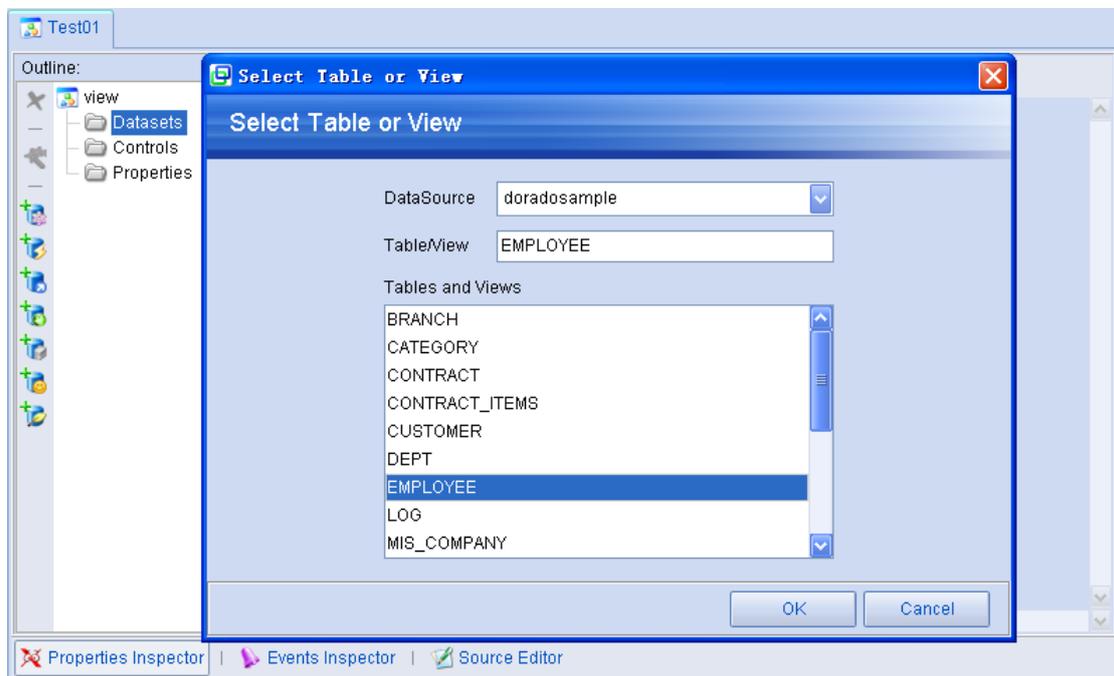


## 2.3. 配置数据来源

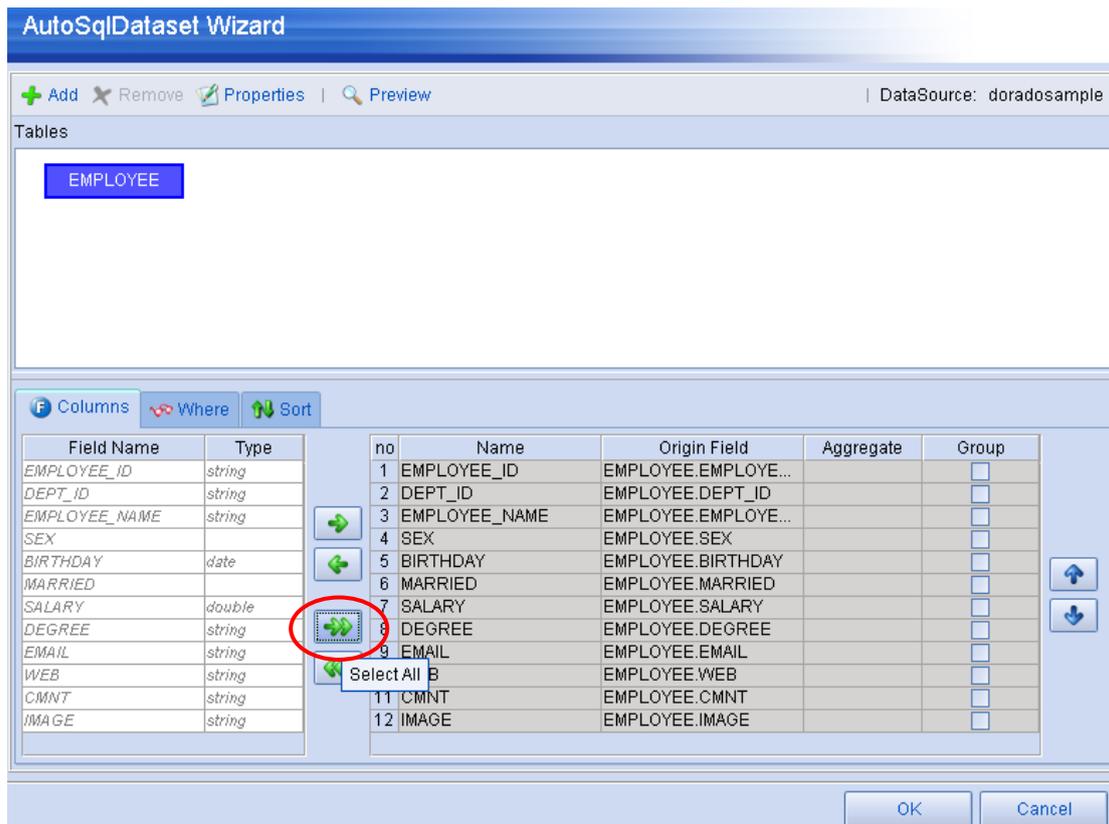
在 Datasets 节点中创建一个 AutoSqlDataset。



选择数据库中的“EMPLOYEE”表。



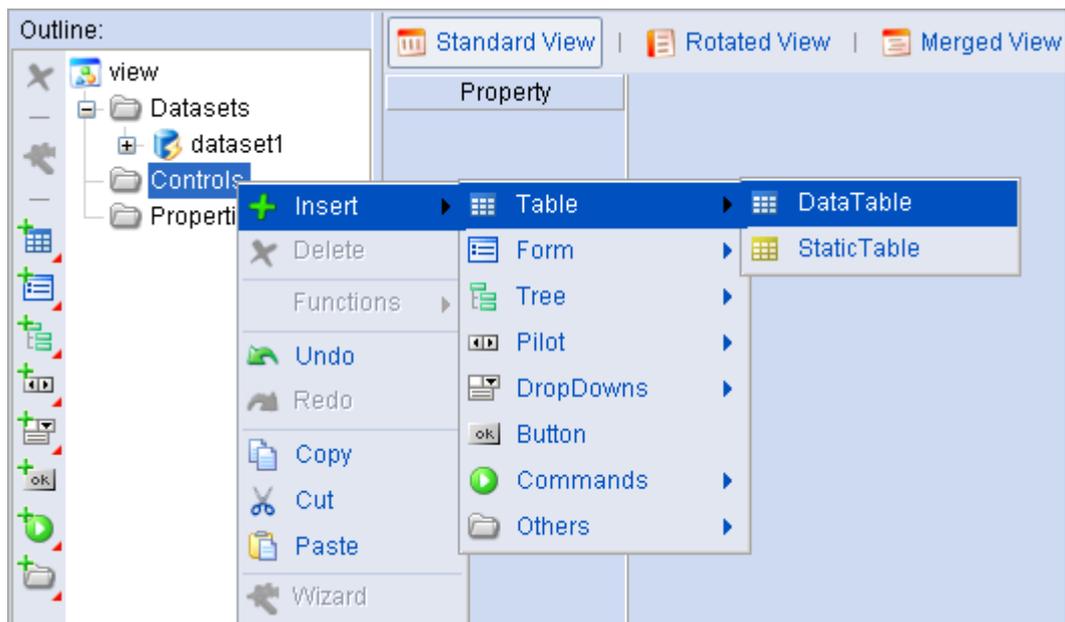
选择表中所有字段。



至此，数据源设置告一段落。

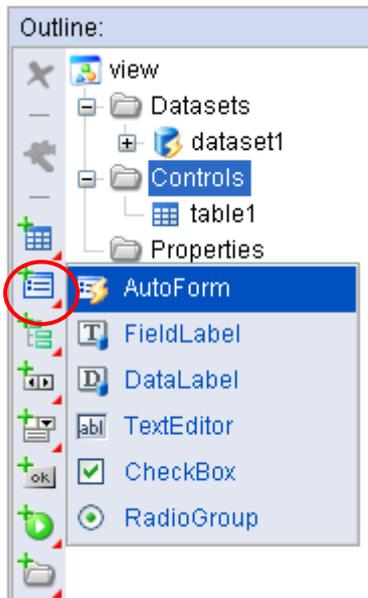
## 2.4. 配置展现形式

在 Controls 节点中右键添加 DataTable (Insert->Table->DataTable)。

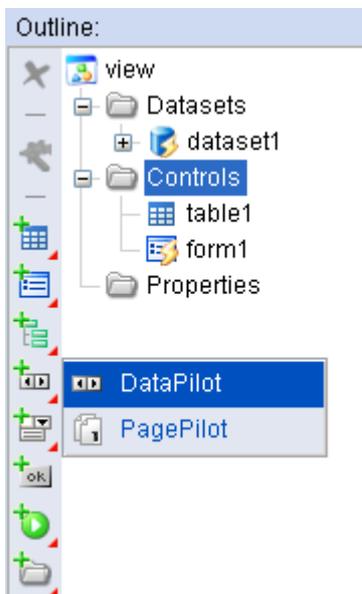


选中 Controls 节点，在左侧 Form 工具条中添加 AutoForm。这个操作相当于右键

(Insert->Form->AutoForm)。



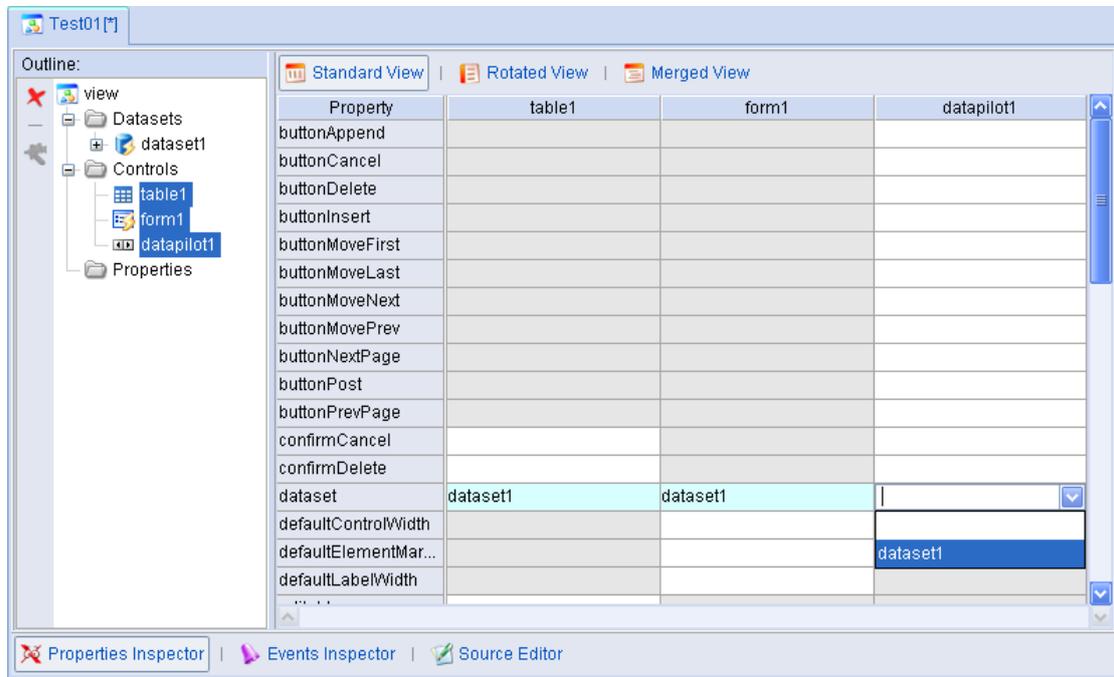
选中 **Controls** 节点，在左侧 **Pilot** 工具条中添加 **DataPilot**。这个操作相当于右键 (Insert->Pilot->DataPilot)。



至此，展现控件设置告一段落。

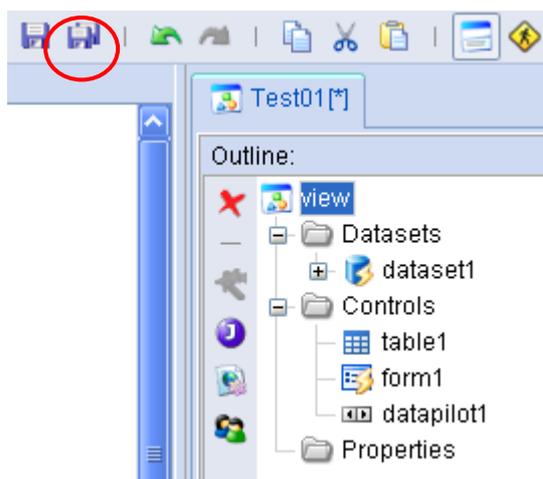
## 2.5. 配置数据源与控件的关系

使用 Ctrl + 鼠标左键选中 table1、form1、datapilot1, 将其 dataset 属性都设为 dataset1。



至此，展现控件与数据源之间关系的设置告一段落。

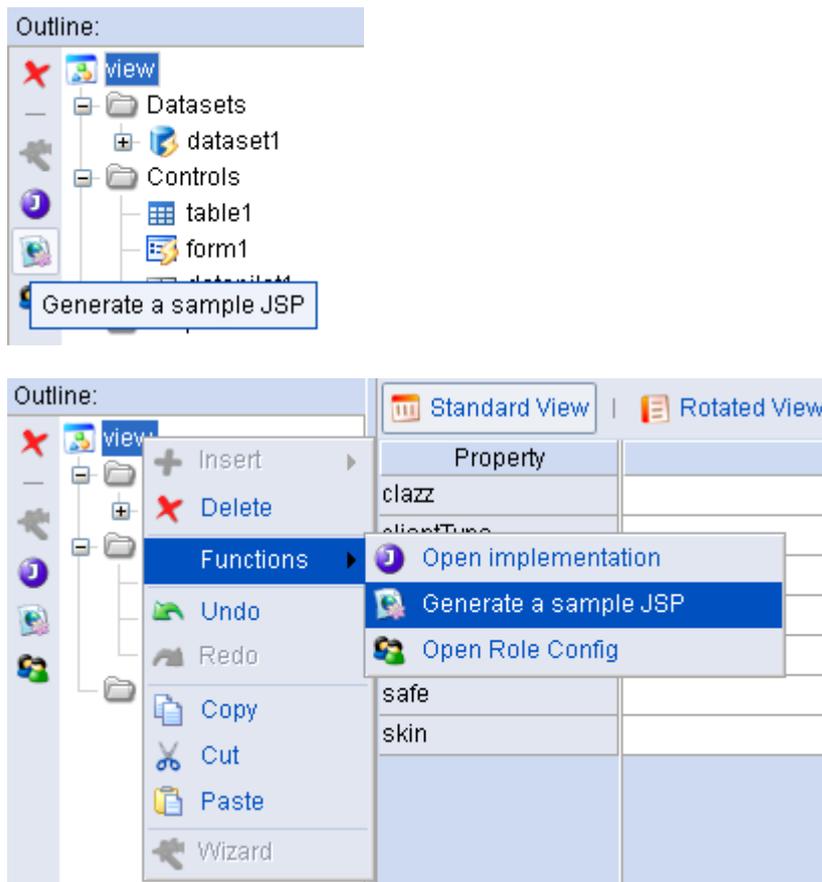
此时做个保存全部的操作。



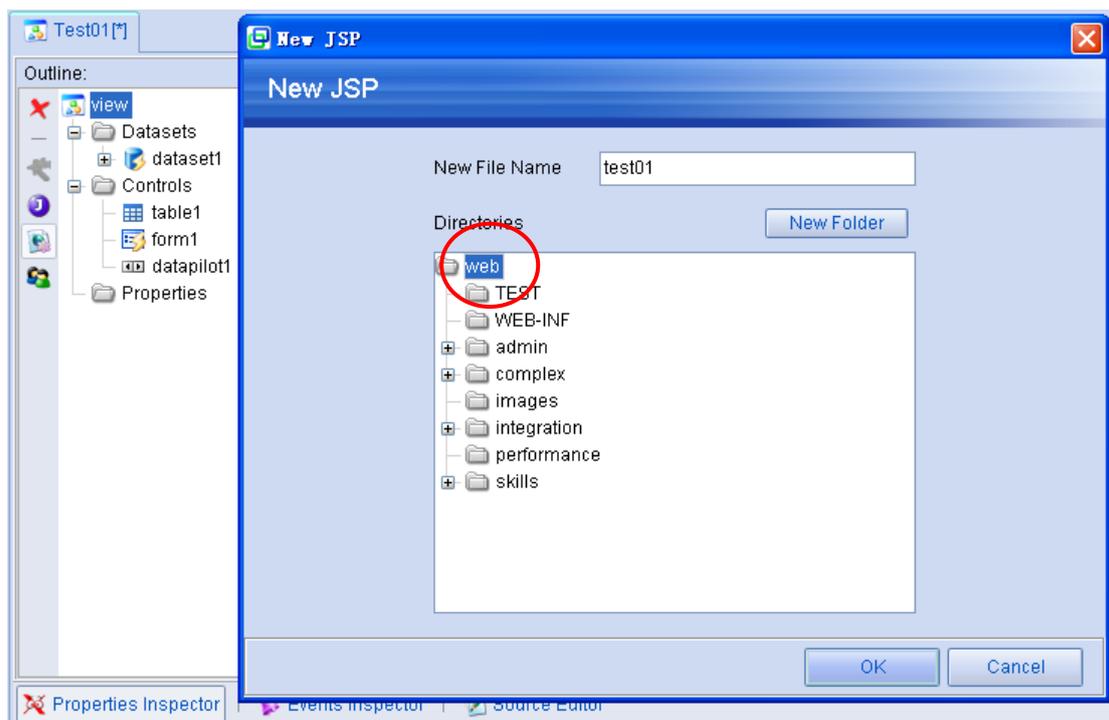
## 2.6. 创建 JSP

选中根节点 View，在左侧工具条中点击“Generate a sample JSP”，创建 JSP。这个

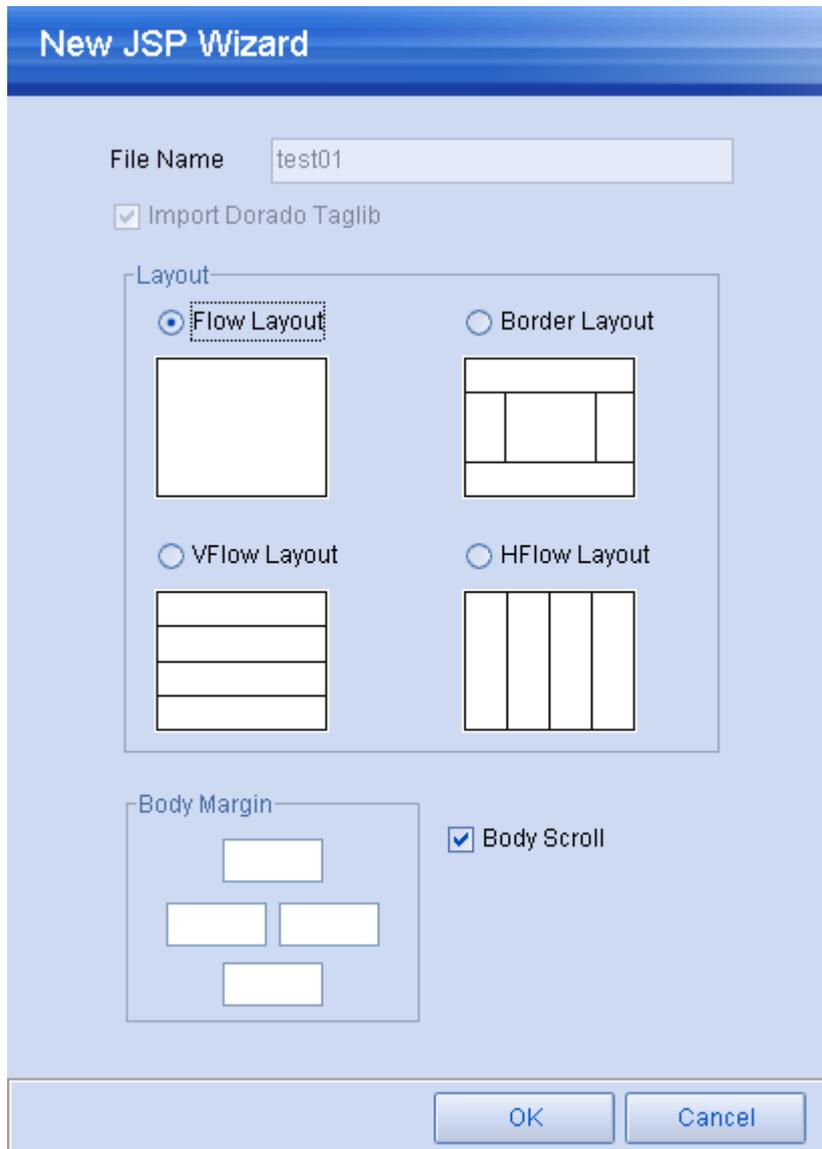
操作相当于右键（Functions-> Generate a sample JSP）。



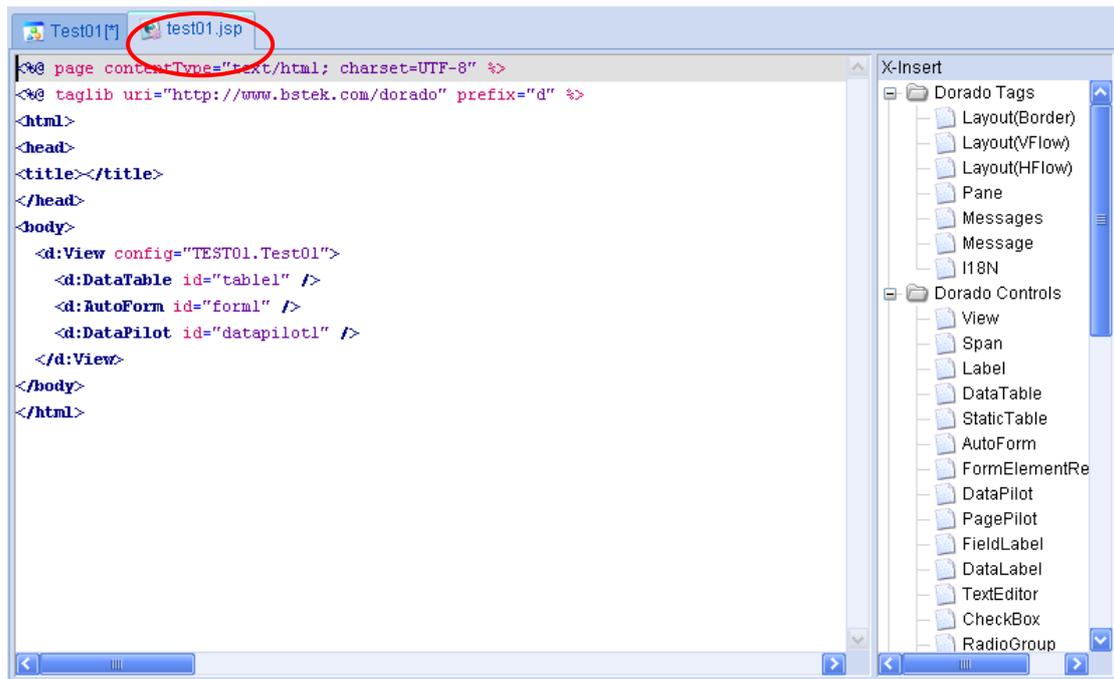
选择目录，如根节点“web”，点击按钮“OK”。



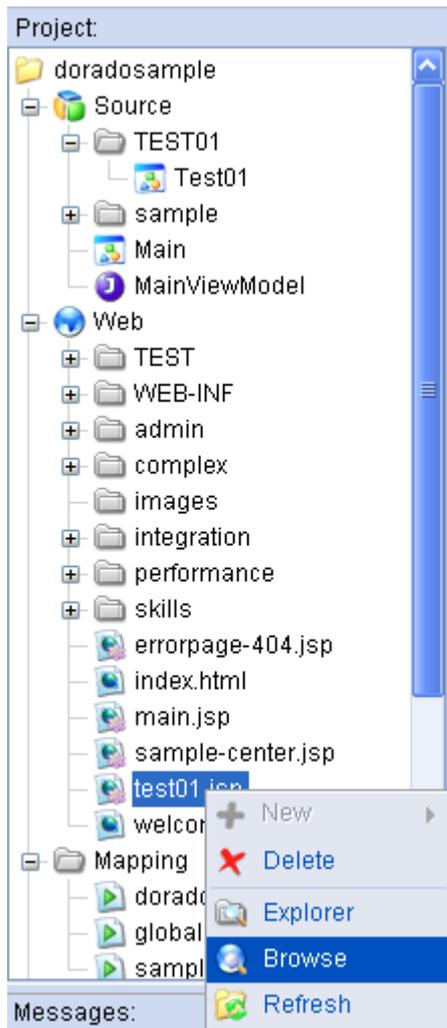
选择默认的布局，点击按钮“OK”。



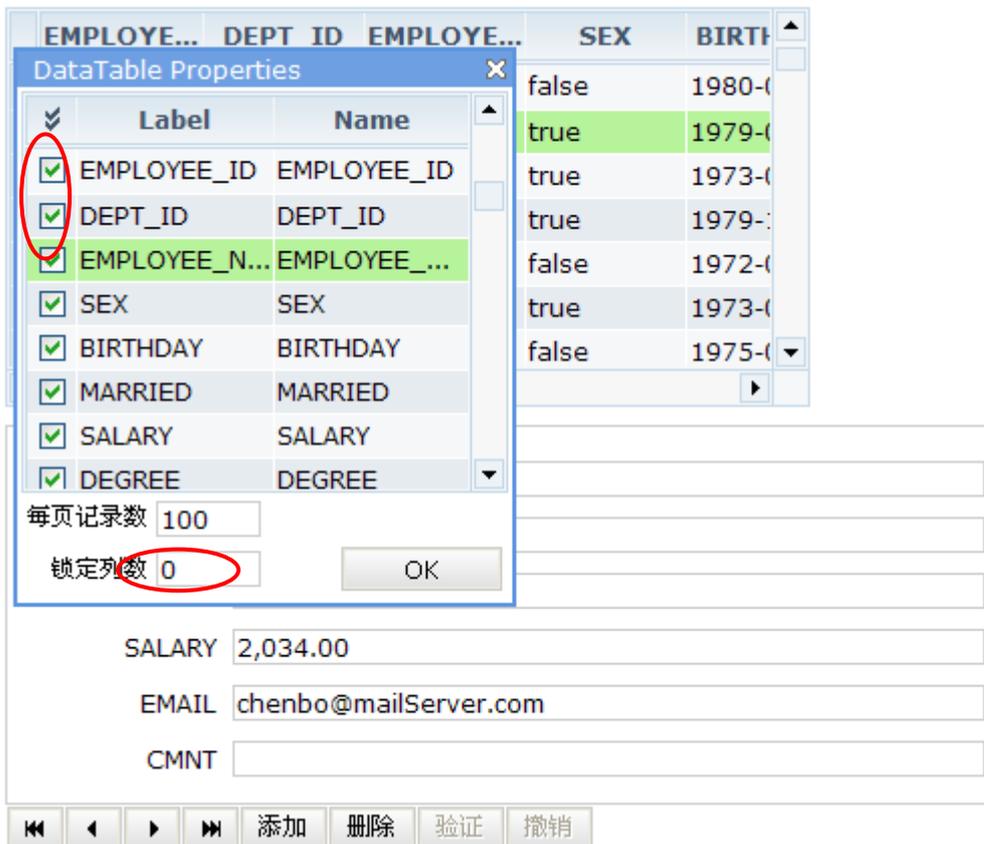
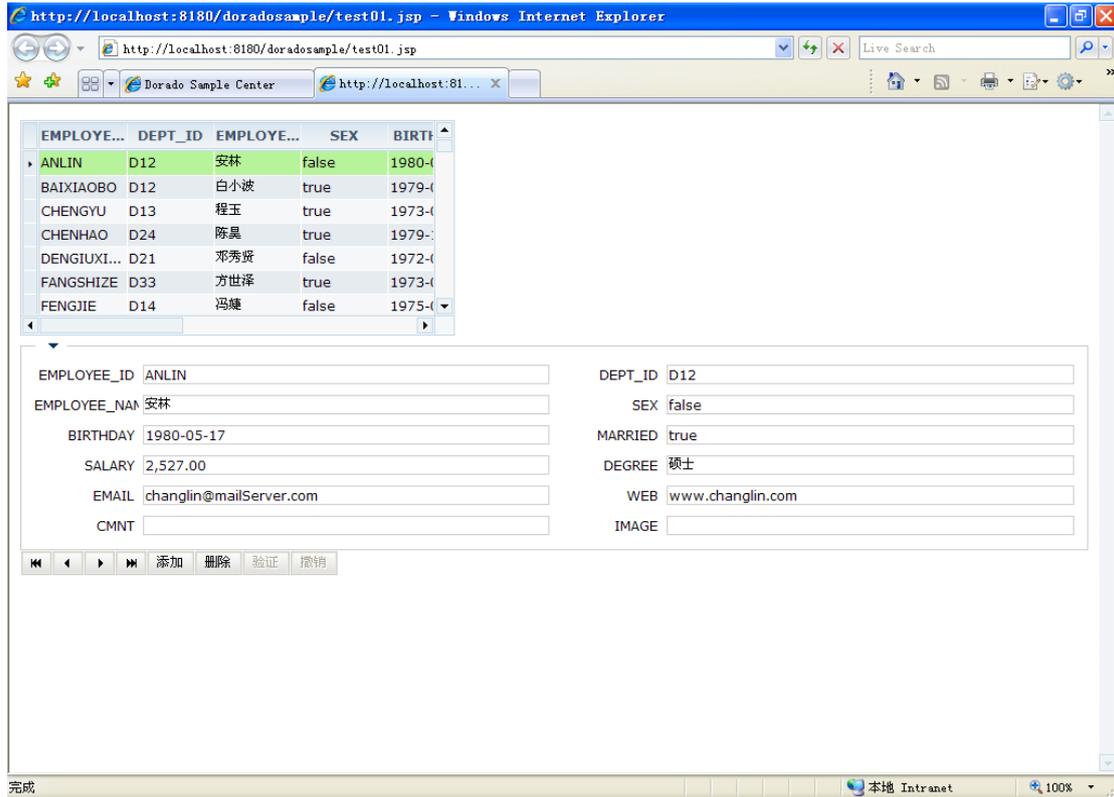
这样就生成了 test01.jsp。



在左侧 Project 树中找到 test01.jsp 右键选择“Browse”。这样一个页面就开发完成了。



这个页面具有开箱即用的功能，如支持表格、表单、数据导航条联动，支持鼠标调整表格列顺序与列宽等。



## 2.7. 单表开发口诀



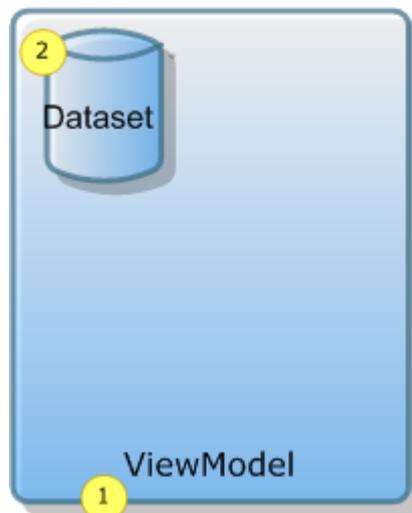
dorado 开发口诀：“单表开发极简单，五步即可成页面，先建视图数据源，再添表格与表单，一内驱动多外设，联动效果自呈现。”

简单回顾开发过程，第一步创建视图模型，第二步设置数据源，第三步添加控件，第四步设置控件的 dataset 属性，第五步生成 JSP。

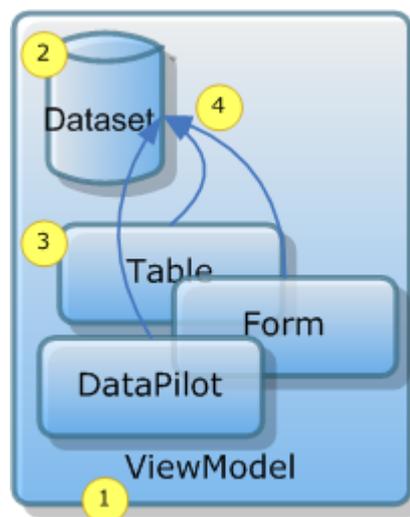
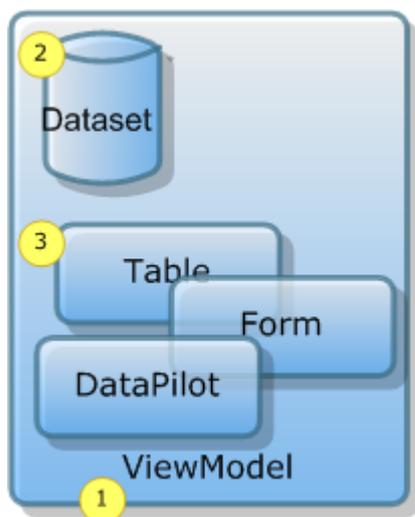
数据源 dataset 是内，在页面上展现的控件 Table、Form、DataPilot 等是外。“一内驱动多外设，联动效果自呈现。”指多个控件绑定到同一个 dataset 上，当用户通过某种展现形式操作 dataset 时，由 dorado 引擎负责将这种改变广播到其它与该 dataset 绑定的展现控件上，从而实现多个展现控件的联动效果。



第 1 步创建视图模型

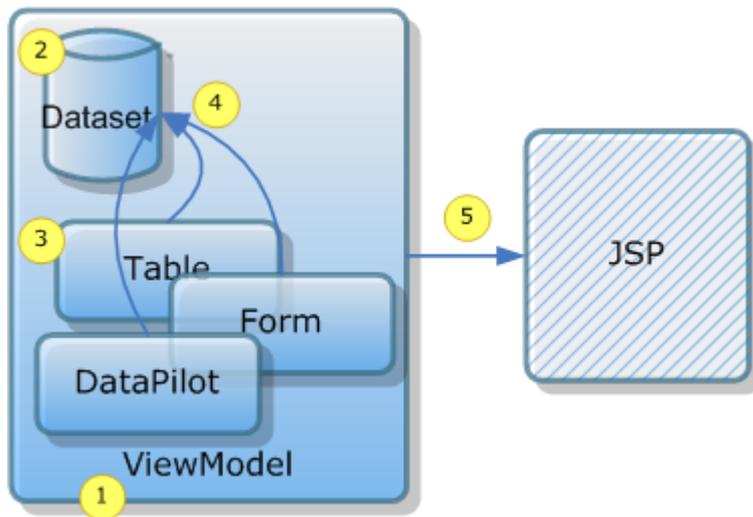


第 2 步设置数据源



第 3 步添加控件

第 4 步设置控件的 dataset 属性

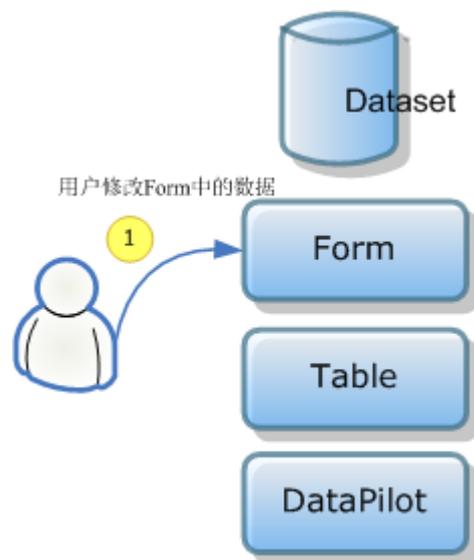


第 5 步生成 JSP。

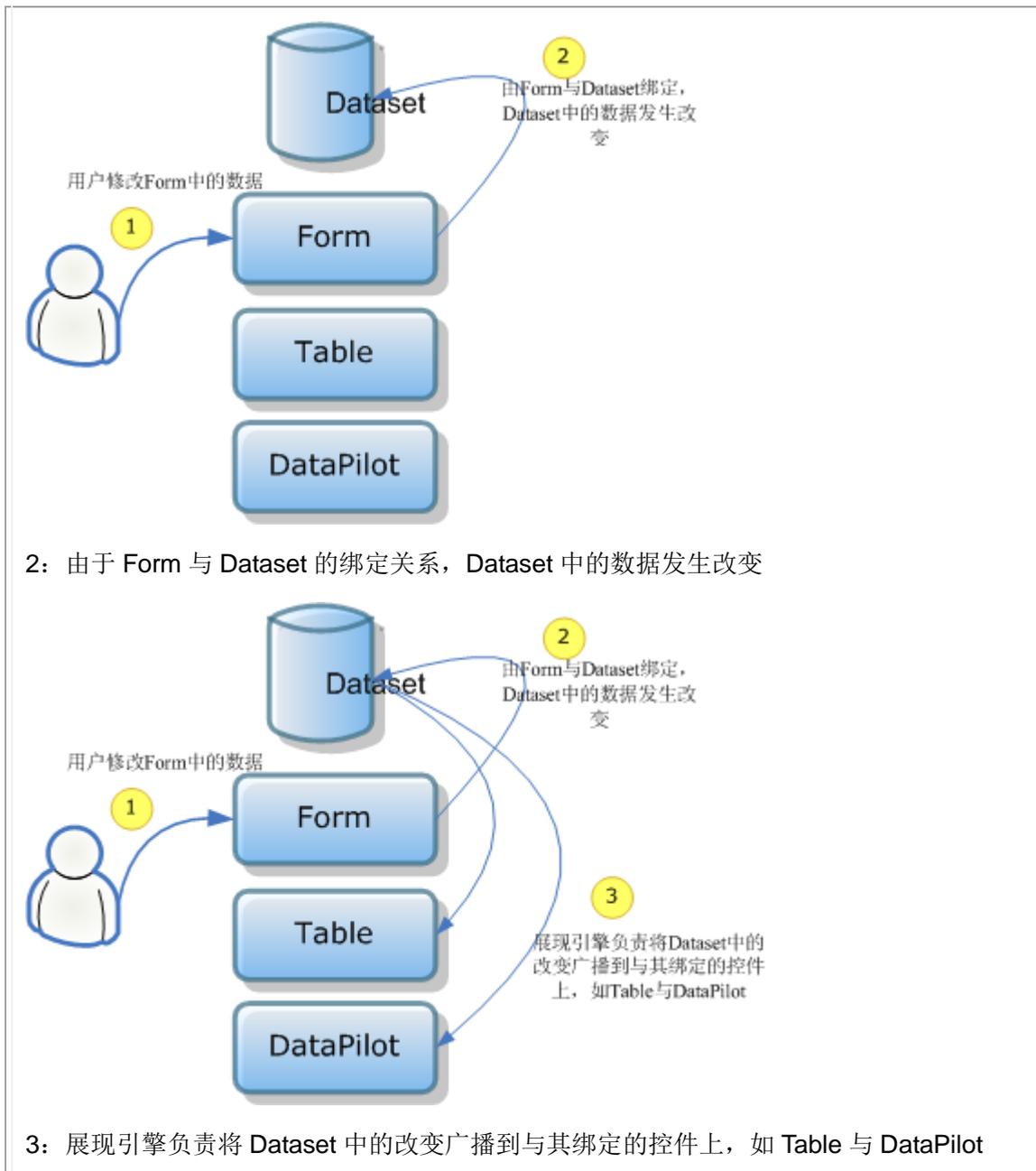
## dorado 原理：数据展现的运行时

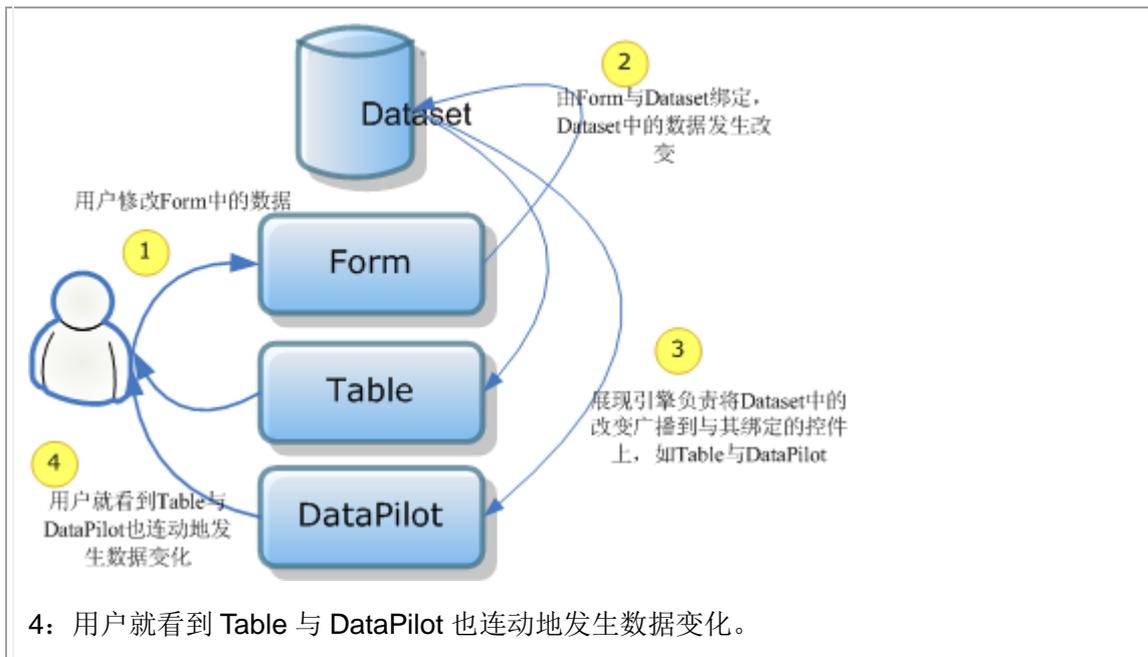


dorado 原理：数据展现的运行时

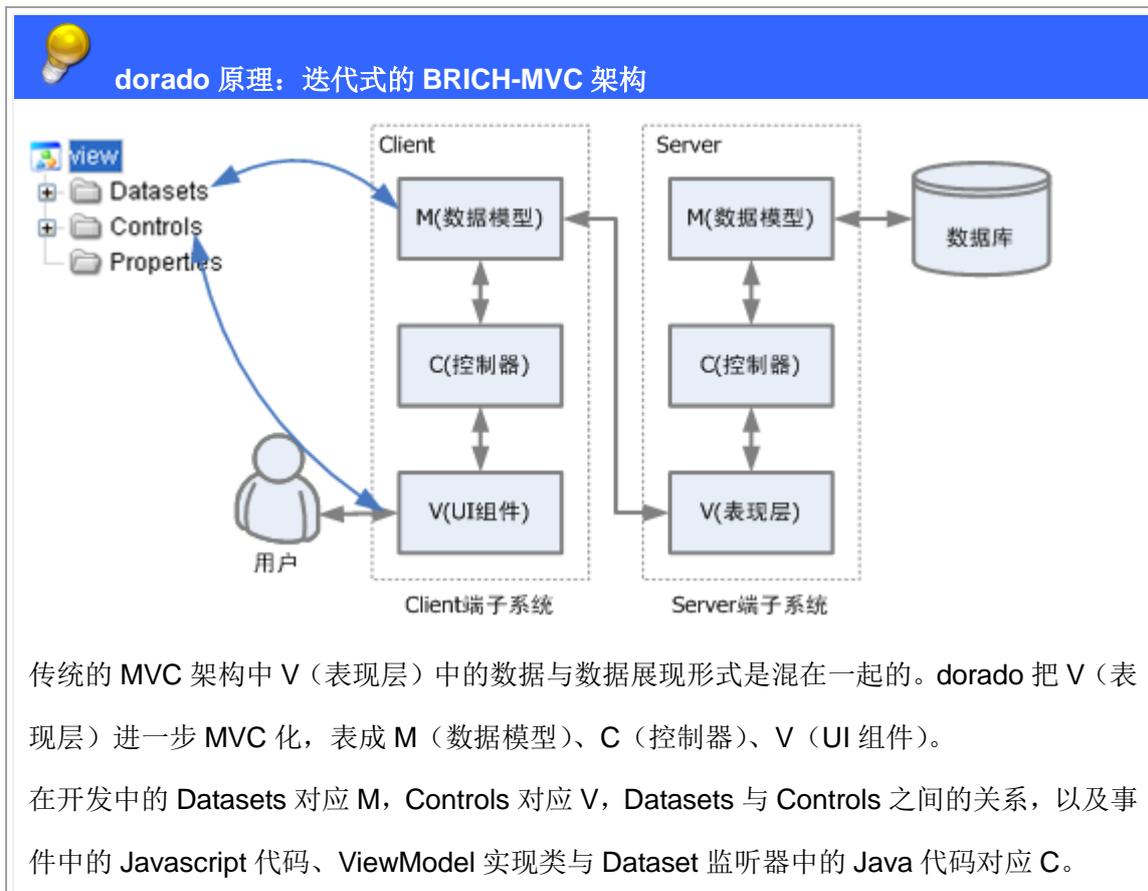


1: 用户修改 Form 中的数据





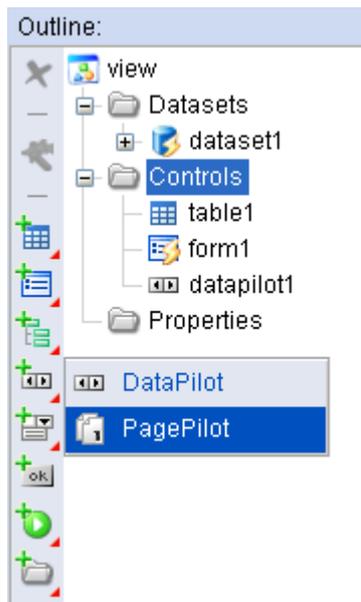
## dorado 原理：迭代式的 BRICH-MVC 架构



## 3. 第二杯茶：页面调整

### 3.1. 添加页面导航条 PagePilot

先做一个与 TEST01 完全相同的页面 TEST02。添加一个页面导航条 PagePilot。

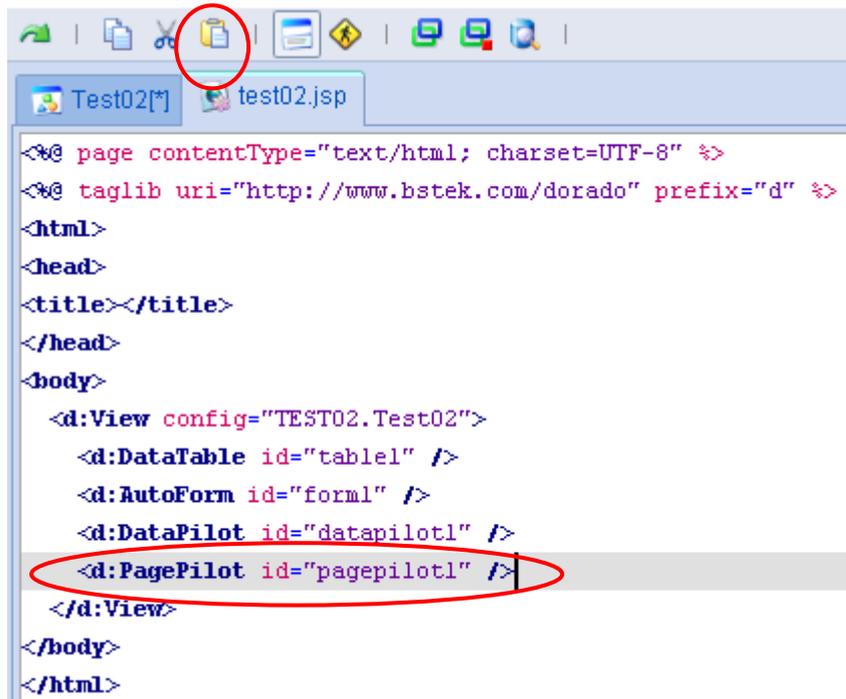
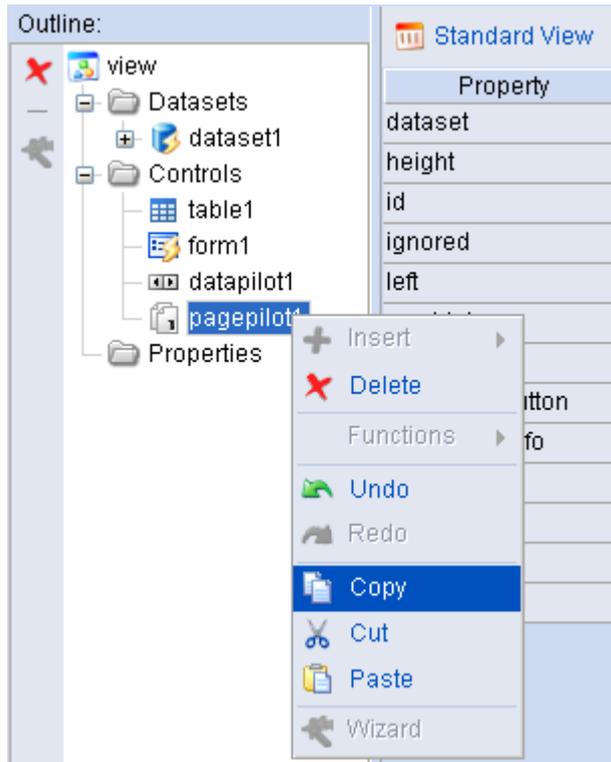


将 PagePilot1 的 dataset 属性设为 dataset1，并把在视图模型中新添加的页面导航条 PagePilot1，输出到 JSP 中。

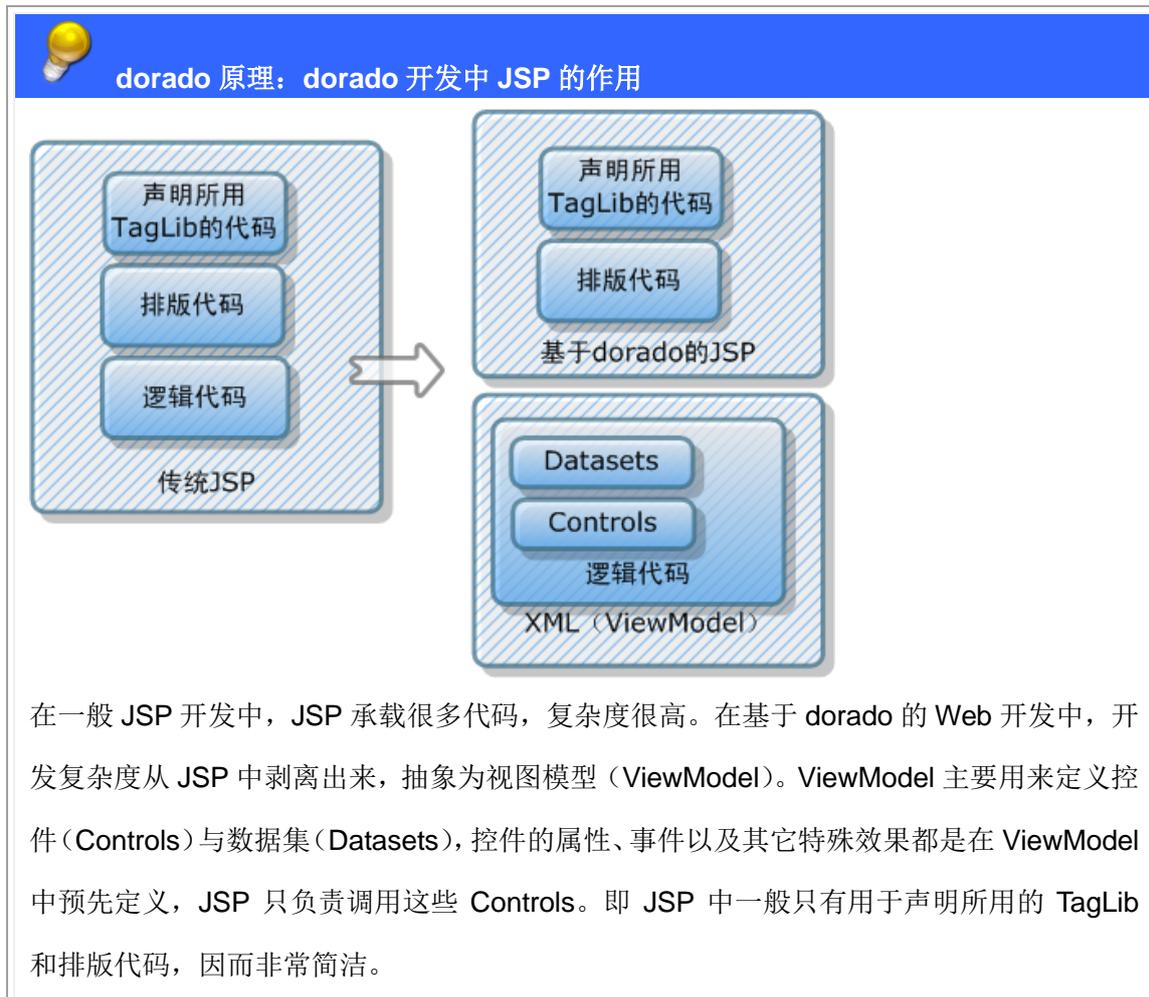


**dorado 开发小技巧，快速添加控件到 JSP 中。**

第一次由视图模型生成 JSP 是 dorado Studio 自动完成的，之后新增的控件就需要手动加到 JSP 中了。将视图模型中新添加的控件在输出到 JSP 中，可以手工写也可以在视图模型界面上 **Ctrl+C**，在 JSP 界面上 **Ctrl+V** 输出，或者利用右键菜单与工具栏菜单。



## dorado 原理：dorado 开发中 JSP 的作用



这时浏览 test02.jsp，可以看到页面导航条已呈现在 web 页面中，但没有分页效果。

EMPLOYEE_ID	ANLIN
EMPLOYEE_NAME	安林
BIRTHDAY	1980-05-17
SALARY	2,527.00
EMAIL	changlin@mailServer.com
CMNT	

⏪ ⏩ 添加 删除 验证 撤销

1 第1页/共1页(共60条记录) 1 转到



### dorado 开发小技巧。

页面导航条 PagePilot 与数据导航条 DataPilot 不同, 如果都没有绑定 dataset, 前者不出现, 后者出现, 但所有按钮灰化。

## 3.2. 设置分页

将 dataset1 的 pageSize 属性设为 10, 保存全部后再浏览就出现了分页效果。

The screenshot shows the Dorado IDE interface. On the left, the 'Outline' pane displays a tree view of the project structure: 'view' contains 'Datasets' (with 'dataset1' selected), 'Controls' (containing 'table1', 'form1', 'datapilot1', and 'pagepilot1'), and 'Properties'. On the right, the 'Property' window for 'dataset1' is visible, showing various properties. The 'pageSize' property is highlighted with a red circle and has the value '10' entered in its text box. Other visible properties include 'originTable' (EMPLOYEE), 'pageCount', 'pageIndex', 'readOnly', 'retrieveAfterUpdate', 'selectAllField', and 'showLoadingTip'.

不仅页面导航条能看出分页效果, 与 dataset1 绑定的 table1 右侧的数据定位滚动条也体现分页效果。

EMPLOYEE...	DEPT_ID	EMPLOYEE...	SEX	BIRTHDAY
ANLIN	D12	安林	false	1980-05-17
BAIXIAOBO	D12	白小波	true	1979-05-17
CHENGYU	D13	程玉	true	1973-05-17
CHENHAO	D24	陈昊	true	1979-05-17
DENGIUXI...	D21	邓秀贤	false	1972-05-17
FANGSHIZE	D33	方世泽	true	1973-05-17
FENGJIE	D14	冯婕	false	1975-05-17

EMPLOYEE_ID	ANLIN
EMPLOYEE_NAME	安林
BIRTHDAY	1980-05-17
SALARY	2,527.00
EMAIL	changlin@mailServer.com
CMNT	

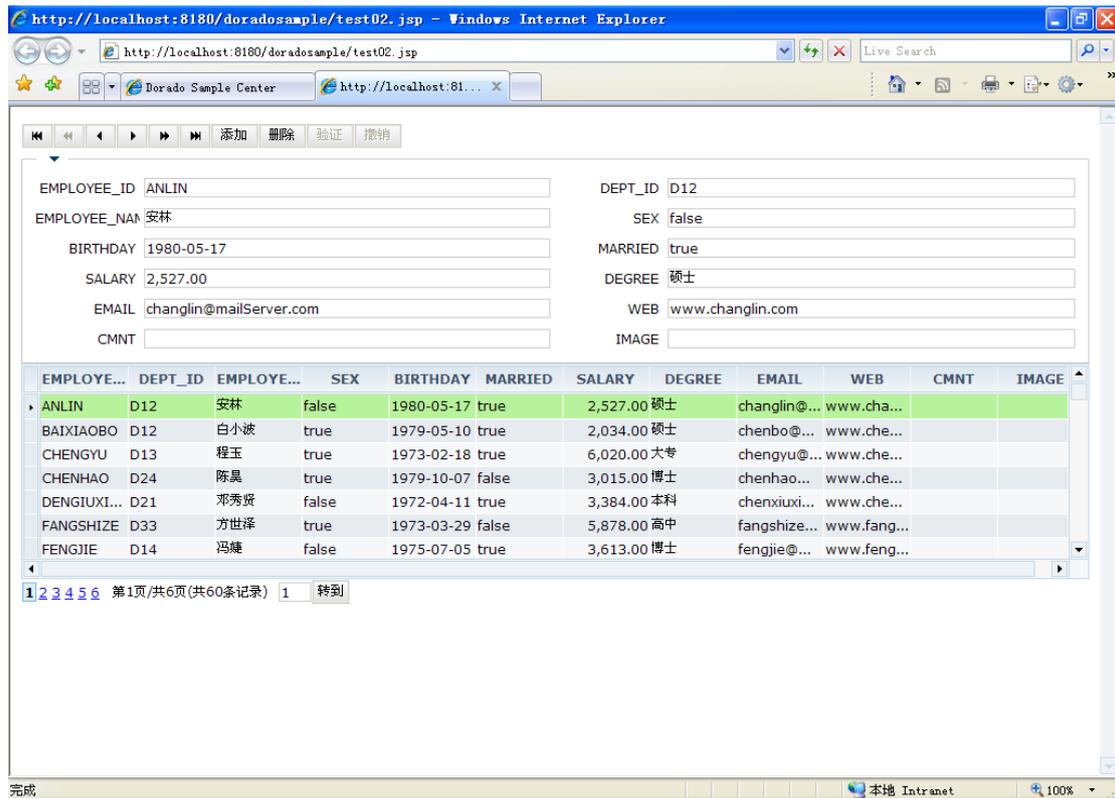
第1页/共6页(共60条记录)

最后再把 table1 的 width 属性设为 100%，调整 JSP 界面中的 TagLib 的输出位置，得到页面。

```

<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib uri="http://www.bstek.com/dorado" prefix="d" %>
<html>
<head>
<title></title>
</head>
<body>
  <d:View config="TEST02.Test02">
    <d:DataPilot id="datapilot1" />
    <d:AutoForm id="form1" />
    <d:DataTable id="table1" />
    <d:PagePilot id="pagepilot1" />
  </d:View>
</body>
</html>

```



EMPLOYEE\_ID: ANLIN DEPT\_ID: D12  
EMPLOYEE\_NAME: 安林 SEX: false  
BIRTHDAY: 1980-05-17 MARRIED: true  
SALARY: 2,527.00 DEGREE: 硕士  
EMAIL: changlin@mailServer.com WEB: www.changlin.com  
CMNT: IMAGE:

EMPLOYEE...	DEPT_ID	EMPLOYEE...	SEX	BIRTHDAY	MARRIED	SALARY	DEGREE	EMAIL	WEB	CMNT	IMAGE
ANLIN	D12	安林	false	1980-05-17	true	2,527.00	硕士	changlin@...	www.cha...		
BAIXIAOBO	D12	白小波	true	1979-05-10	true	2,034.00	硕士	chenbo@...	www.che...		
CHENGYU	D13	程玉	true	1973-02-18	true	6,020.00	大专	chengyu@...	www.che...		
CHENHAO	D24	陈昊	true	1979-10-07	false	3,015.00	博士	chenhao@...	www.che...		
DENGIUXI...	D21	邓秀贤	false	1972-04-11	true	3,384.00	本科	chenxiuxi...	www.che...		
FANGSHIZE	D33	方世泽	true	1973-03-29	false	5,878.00	高中	fangshize...	www.fang...		
FENGJIE	D14	冯婕	false	1975-07-05	true	3,613.00	博士	fengjie@...	www.feng...		

1 2 3 4 5 6 第1页/共6页(共60条记录) 1 转到



### dorado 开发技巧小结

新增控件首次要添到 JSP 中，如果没输出，就看看是否绑定了 dataset。分页在 dataset 中设置 pageSize 即可。

## 4. 第三杯茶：单表的增改删存

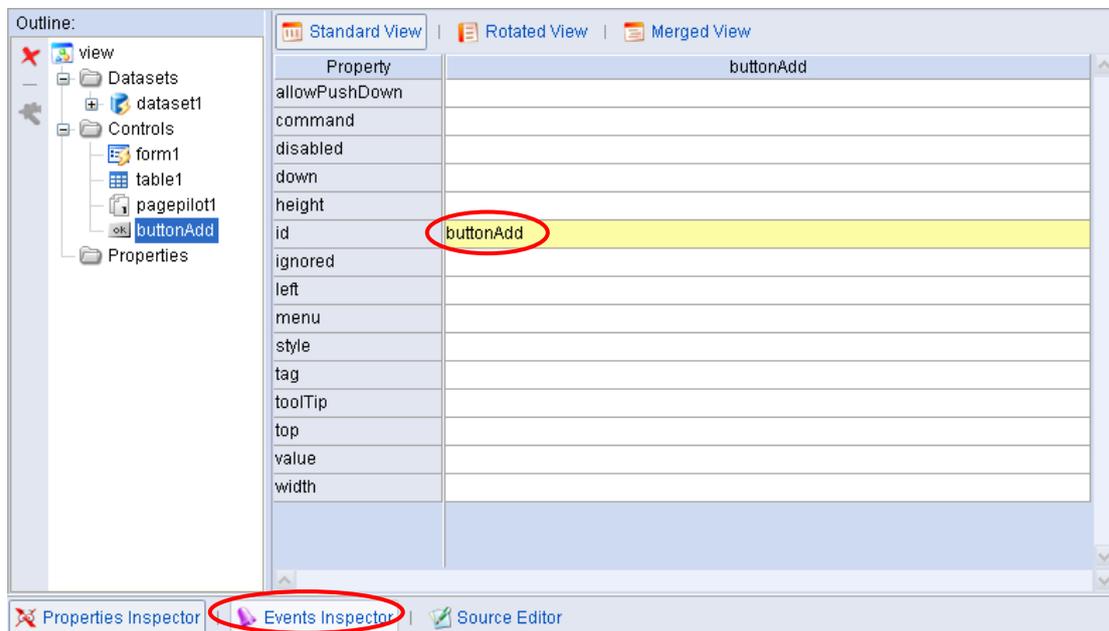
修改最容易，表格表单的全编辑功能是开箱即用的。增加、删除、保存需要写几行代码。查询配置略复杂一点，我们放在后面讲。

### 4.1. 准备工作

创建 TEST03 文件夹，在其中创建视图模型 Test03，添加一个 dataset1 数据来自 EMPLOYEE 表，再添加一个 DataTable，一个 Autoform，一个 DataPilot，全部与 dataset1 绑定。

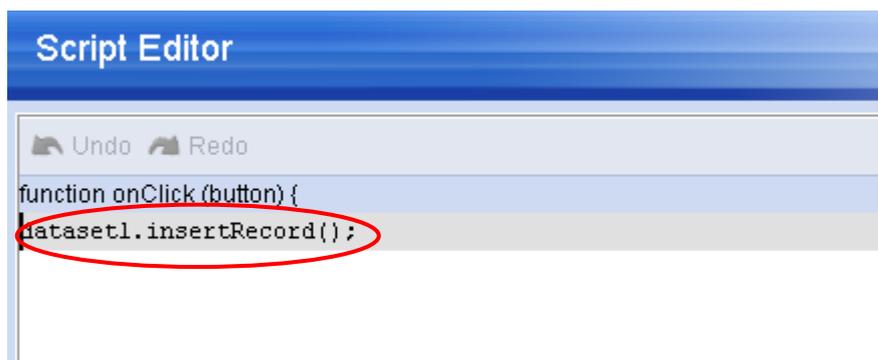
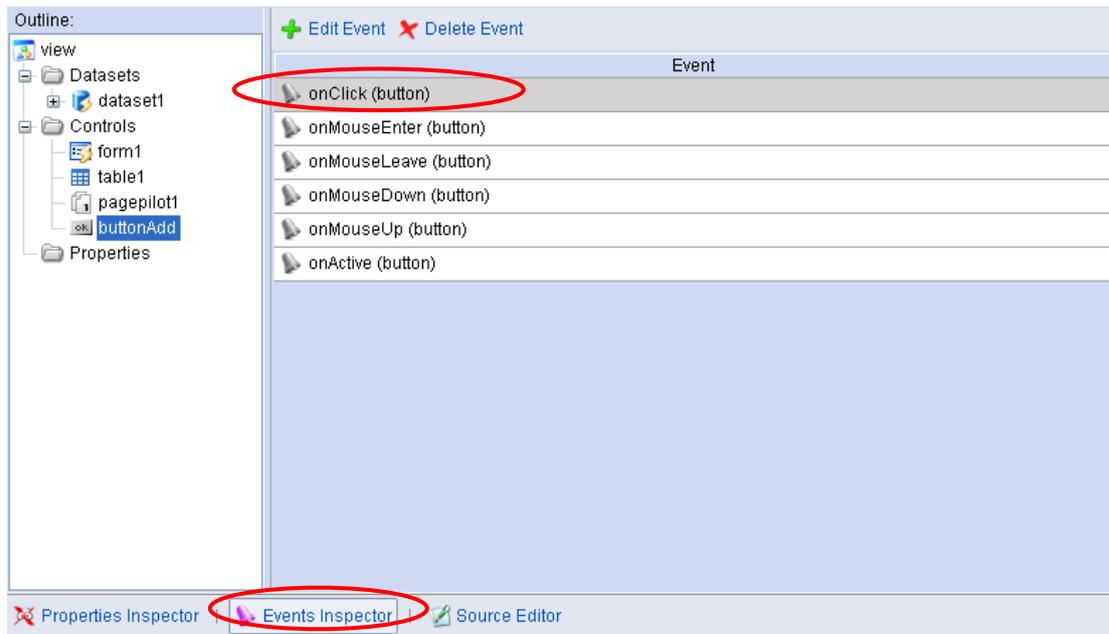
### 4.2. 增加记录

第一步，增加一个按钮 Button，重命名为 buttonAdd。



第二步，切换到 Events Inspector 界面。在 buttonAdd 的 onClick 方法中添加事件代码。

```
dataset1.insertRecord();
```



 dorado 开发小技巧：方法中添加事件后的 highlight 效果。

Outline:

- view
  - Datasets
    - dataset1
  - Controls
    - form1
      - table1
    - pagepilot1
      - buttonAdd
  - Properties

Event

- onClick (button)**
- onMouseEnter (button)
- onMouseLeave (button)
- onMouseDown (button)
- onMouseUp (button)
- onActive (button)

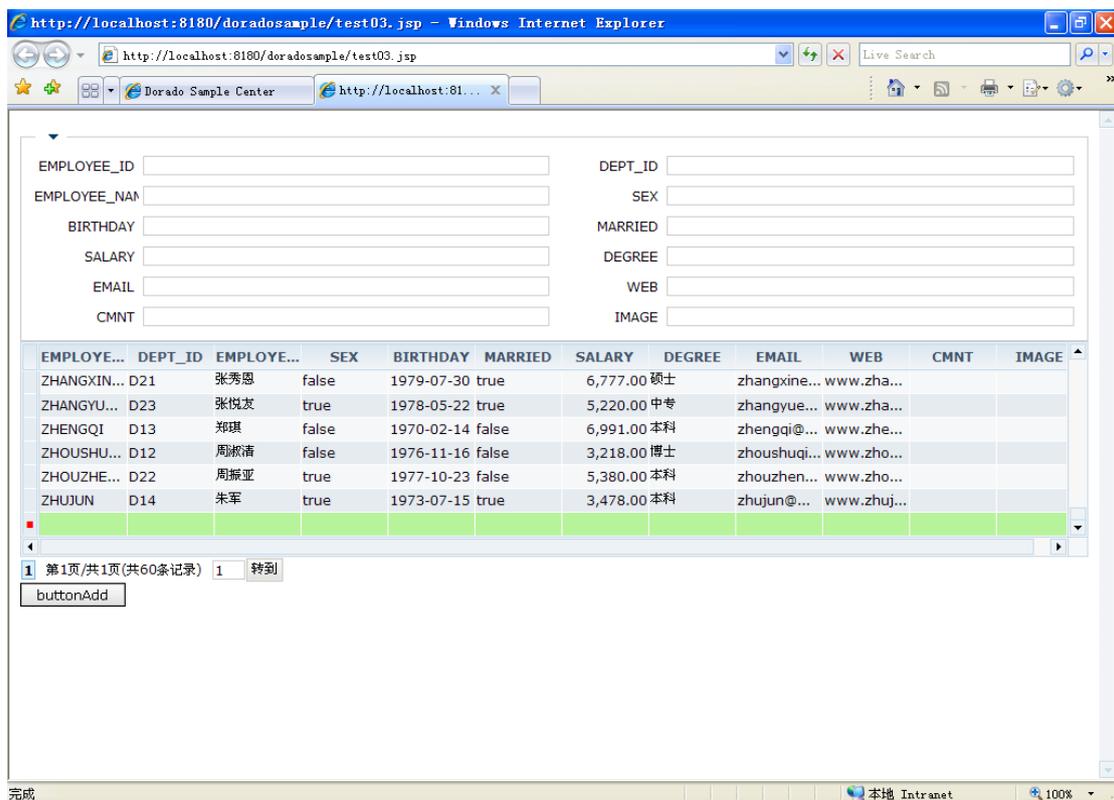
在组件的方法中添加事件后，该方法在 studio 的 Event Inspector 界面中以粗体显示。代表事件的铃铛图标也由灰色变为粉色。

第三步，保存之后，将 `buttonAdd` 添加到 JSP 之中。



**dorado 开发小技巧：**切换到属性查看器 Properties Inspector，才能 Copy Paste 控件到 JSP 中。

浏览 `test03.jsp`，点击页面上的按钮 `buttonAdd`，追加一条新记录到最后。



## 4.3. 修改记录

第一步，添加两个按钮，分别为 `buttonReadOnly` 与 `buttonRevise`。

第二步，在 `buttonReadOnly` 的 `onClick` 方法中添加代码：

```
dataset1.setReadOnly(true);  
//将数据源设为只读  
buttonReadOnly.disabled = true;
```

```
//将只读按钮设为灰化
```

```
buttonRevise.disabled = false;
```

```
//将编辑按钮设为可用
```

在 `buttonRevise` 的 `onClick` 方法中添加代码:

```
dataset1.setReadOnly(false);
```

```
//将数据源设为可编辑
```

```
buttonReadOnly.disabled = false;
```

```
//将只读按钮设为可用
```

```
buttonRevise.disabled = true;
```

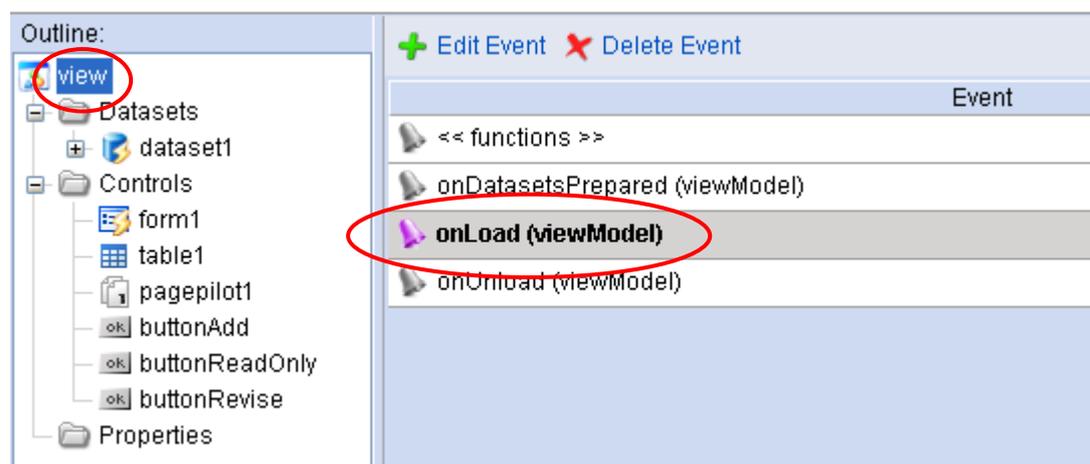
```
//将编辑按钮设为灰化
```

在 `ViewModel` 的实现类的 `onLoad` 方法中添加代码:

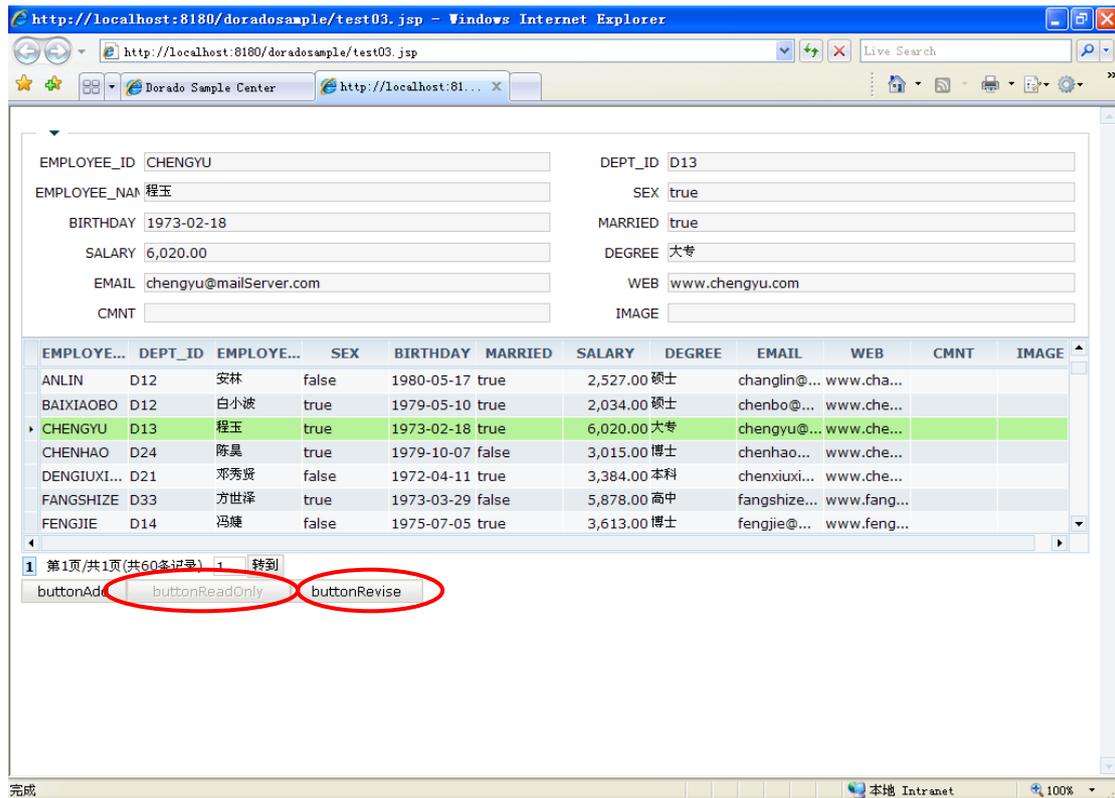
```
buttonReadOnly.disabled = dataset1.isReadOnly();
```

```
buttonRevise.disabled = !dataset1.isReadOnly();
```

```
//初始化时根据数据源的是否只读来设置只读按钮与编辑按钮的灰化与可用状态
```



第三步, 将 `buttonReadOnly` 与 `buttonRevise` 添加到 JSP 中, 浏览页面。



点击按钮 `buttonReadOnly` 将页面设为只读，点击按钮 `buttonRevise` 将页面设为可编辑。

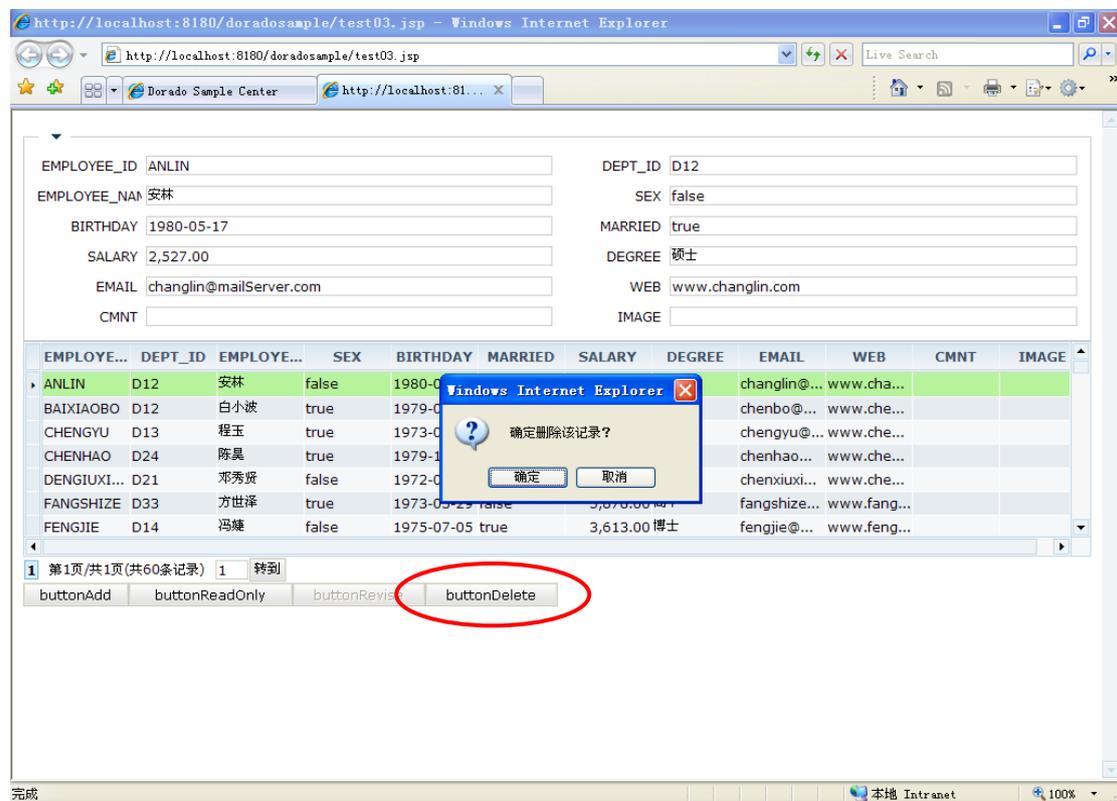
## 4.4. 删除记录

第一步，添加 1 个按钮 `buttonDelete`。

第二步，在 `buttonDelete` 的 `onClick` 方法中添加代码：

```
if (confirm("确定删除该记录? "))  
dataset1.deleteRecord()
```

第三步，将 `buttonDelete` 添加到 JSP 中，浏览页面。



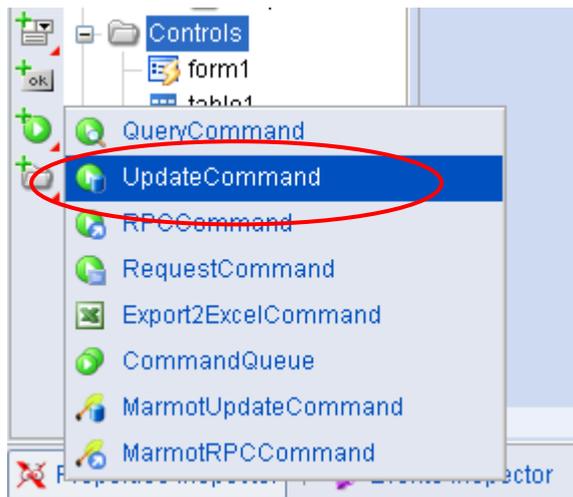
点击按钮 `buttonDelete` 后，弹出对话框点击按钮“确定”后，删除当前记录。如果不需要弹出对话框确认，在 `buttonDelete` 的 `onClick` 方法中添加的代码为：

```
dataset1.deleteRecord()
```

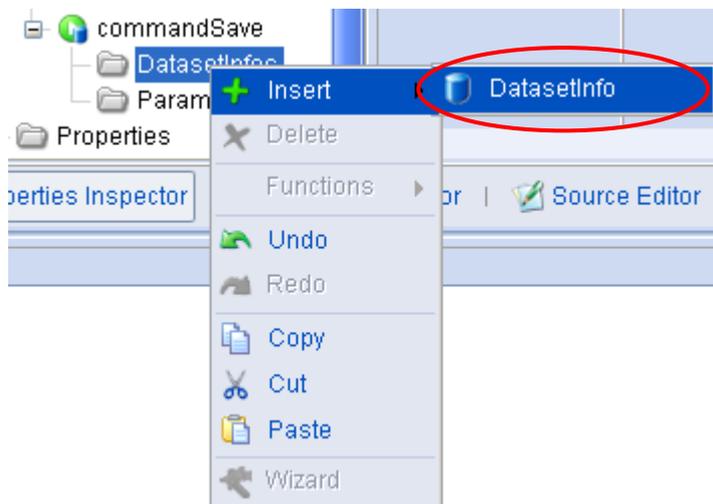
## 4.5. 保存修改

第一步，添加 1 个按钮 `buttonSave`。

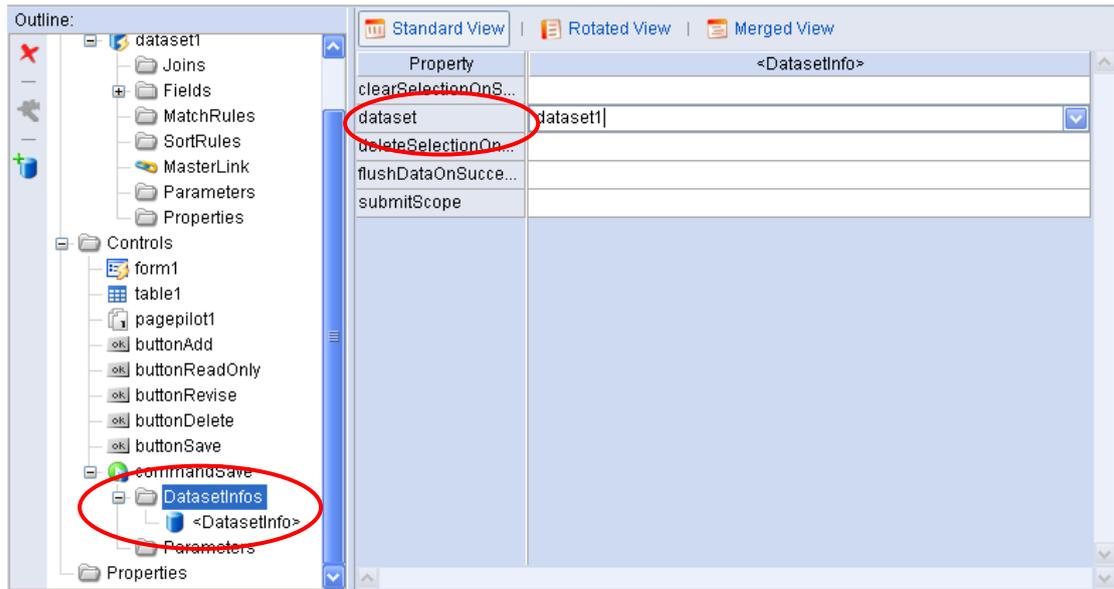
第二步，添加 1 个更新命令 `UpdateCommand`，重命名为 `commandSave`。



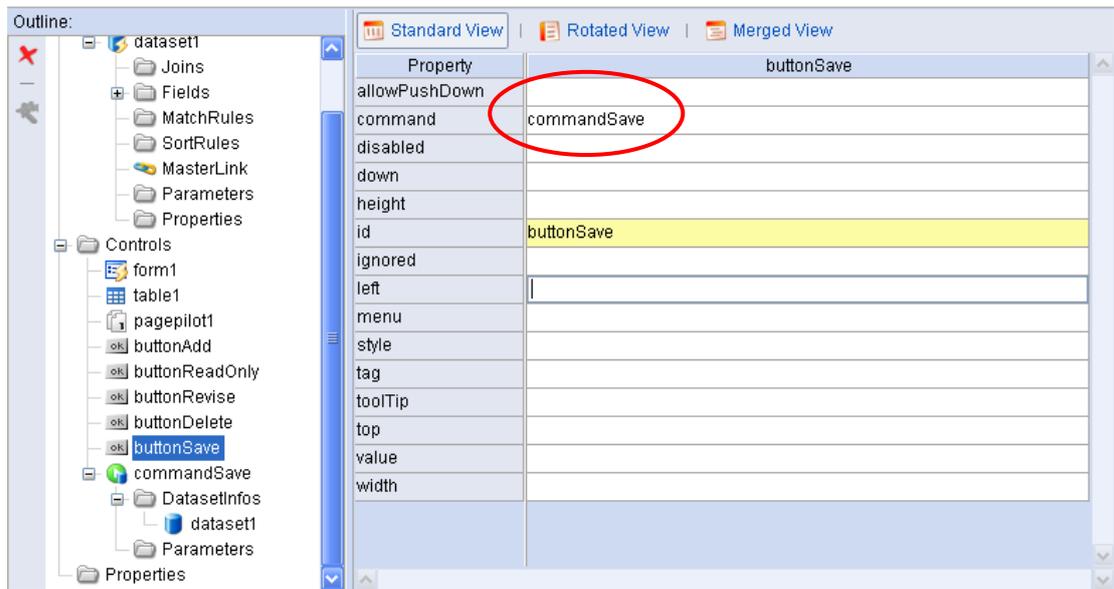
第三步，`commandSave` 节点下有两个子项分别是 `DatasetInfos` 与 `Parameters`。在 `DatasetInfos` 下添加一个 `DatasetInfo`。



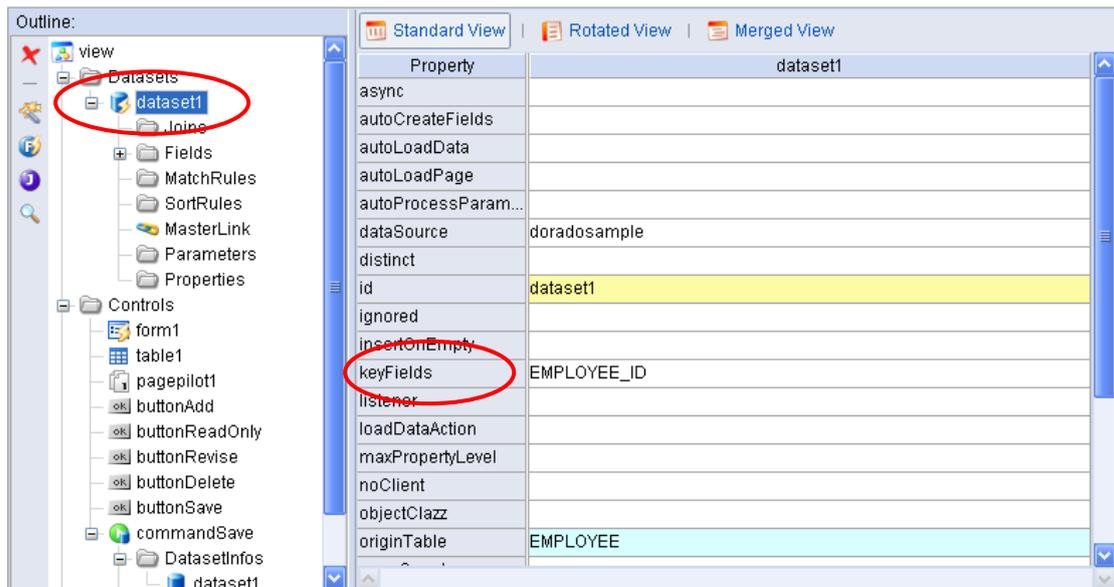
将添加的 `DatasetInfo` 的 `dataset` 属性设为 `dataset1`。



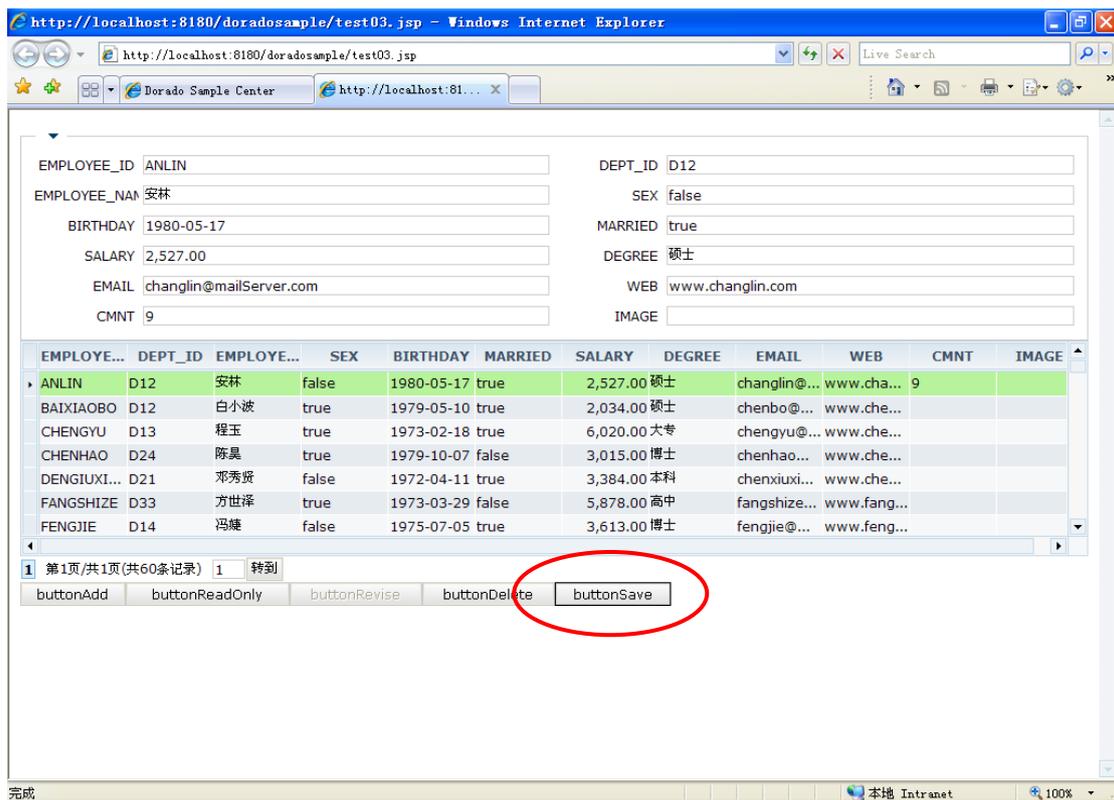
第四步，在 `buttonSave` 的 `command` 属性中选择 `commandSave`，实现按钮与命令的绑定。



第五步，将 `dataset1` 的 `keyFields` 属性设为 `EMPLOYEE_ID`。



将 `buttonSave` 添加到 JSP 中，保存全部，浏览页面。



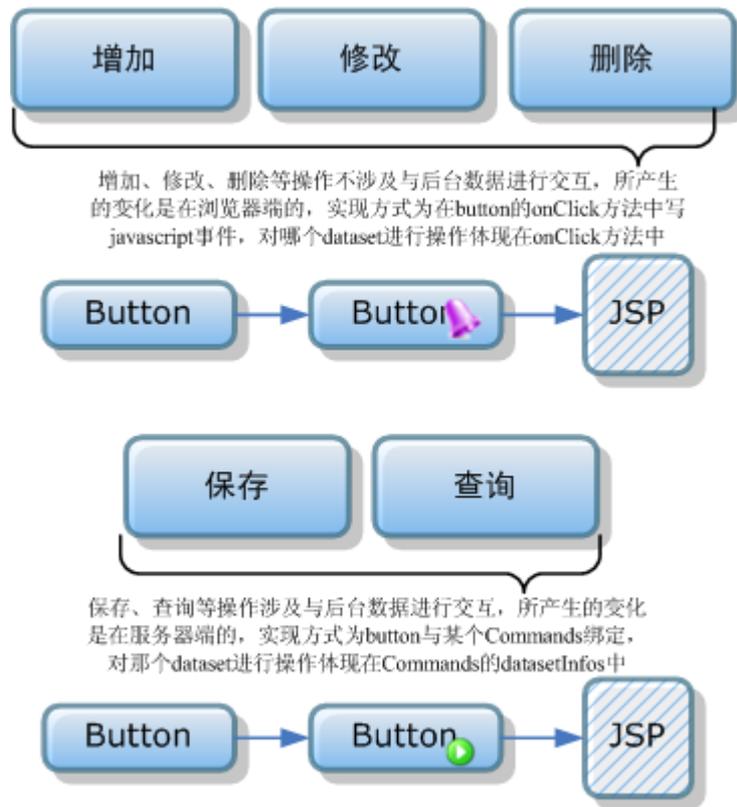
随意做几处修改，点击按钮 `buttonSave`，会有一对话框一闪而过，这个过程即是保存过程。刷新页面，会看到所做的修改都做了保存了。

## dorado 原理：dorado 中的 commands



### dorado 原理：dorado 中的 commands

dorado 中的 commands 用于浏览器与服务器进行数据交互，如提交数据等，需要与 1 个或多个 dataset 绑定。



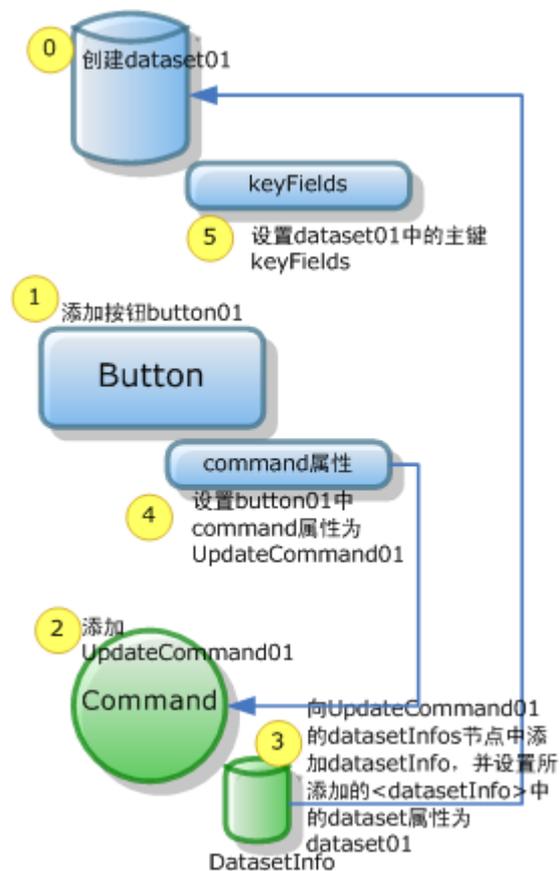
在这个例子中，增加、修改、删除，这 3 个操作不涉及与后台数据进行交互，所产生的变化是在浏览器端的，保存以及查询操作涉及与后台数据进行交互，所产生的变化是在服务器端的。因而，增加、修改、删除，这 3 个操作是在 button 的 onClick 方式中写 javascript 事件，对哪个 dataset 进行操作体现在 onClick 方法中；保存以及查询操作是将 button 的 command 属性与某个 UpdateCommand 或 QueryCommand 绑定，对那个 dataset 进行操作体现在 Commands 的 datasetInfos 中。

## dorado 原理：dorado 中的保存



### dorado 原理：dorado 中的保存

Dorado 中的保存或称之为向后台提交数据，是通过更新命令 `UpdateCommand` 来实现的。`UpdateCommand` 需要设置的关键属性是其子项 `DatasetInfo` 中的 `dataset`，即告诉 `UpdateCommand` 是作用于哪一个或几个 `dataset`。`UpdateCommand` 属于不可见控件，需要与一个可见控件相绑定，一般选用按钮 `Button`。此外还要设置被提效的 `dataset` 中的主键 (`keyFields`) 属性。



具体步骤是，

第 0 步准备工作，创建 `dataset01`。

第 1 步添加按钮 `button01`。

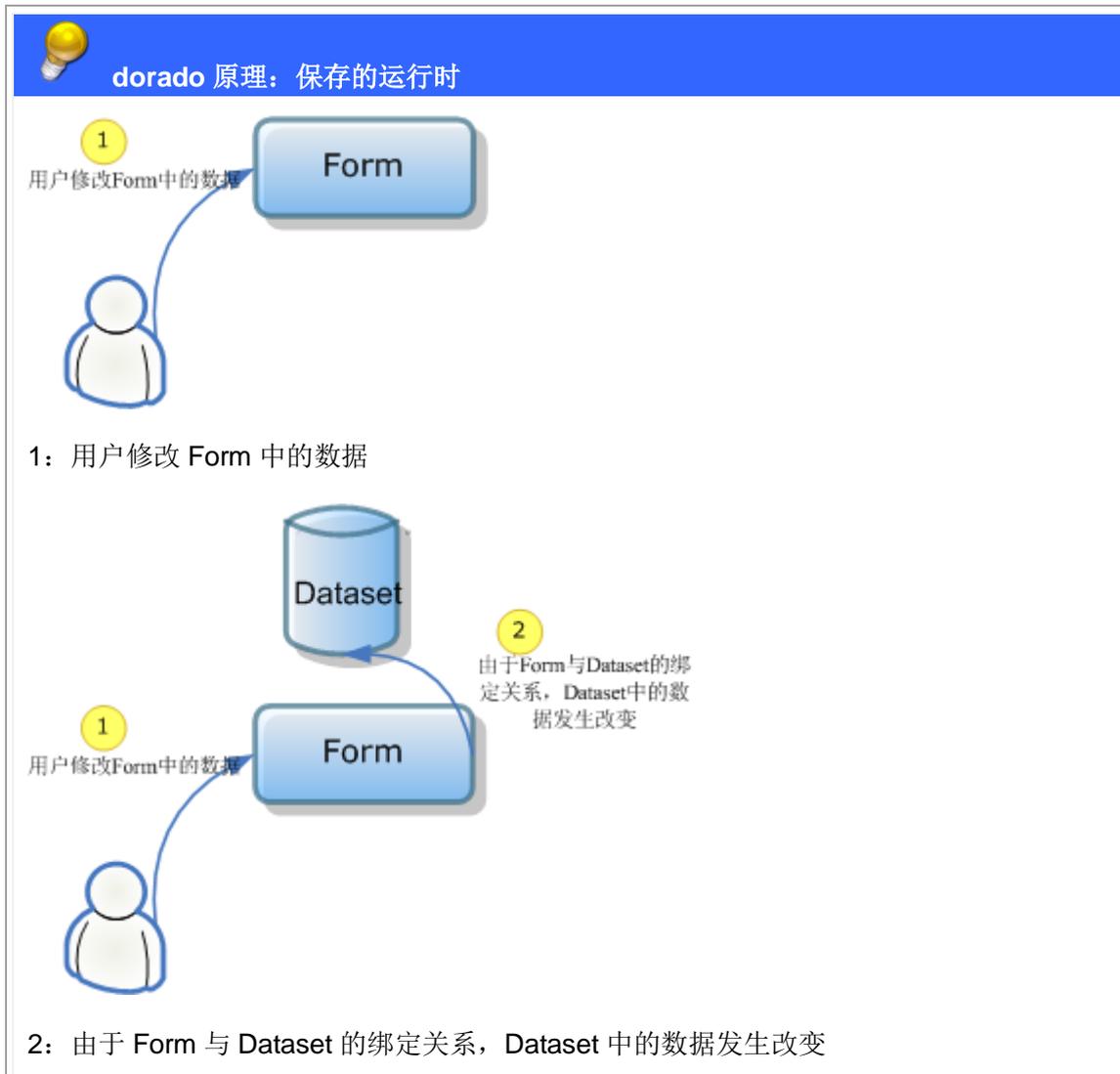
第 2 步添加 `UpdateCommand01`。

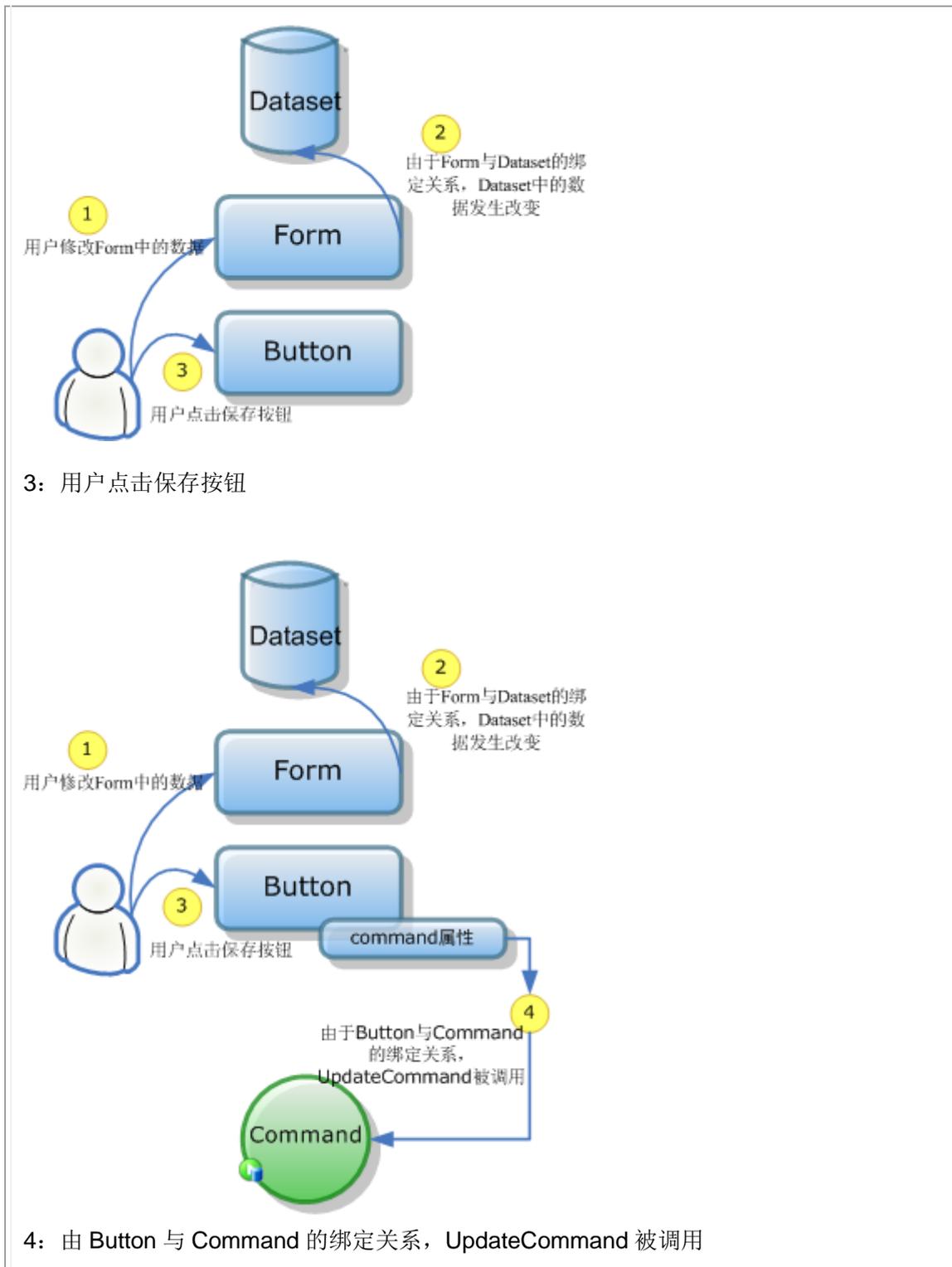
第 3 步在 `UpdateComman01` 的 `datasetInfos` 节点中添加 `datasetInfo`，并设置所添加的 `<datasetInfo>` 中的 `dataset` 属性为 `dataset01`。

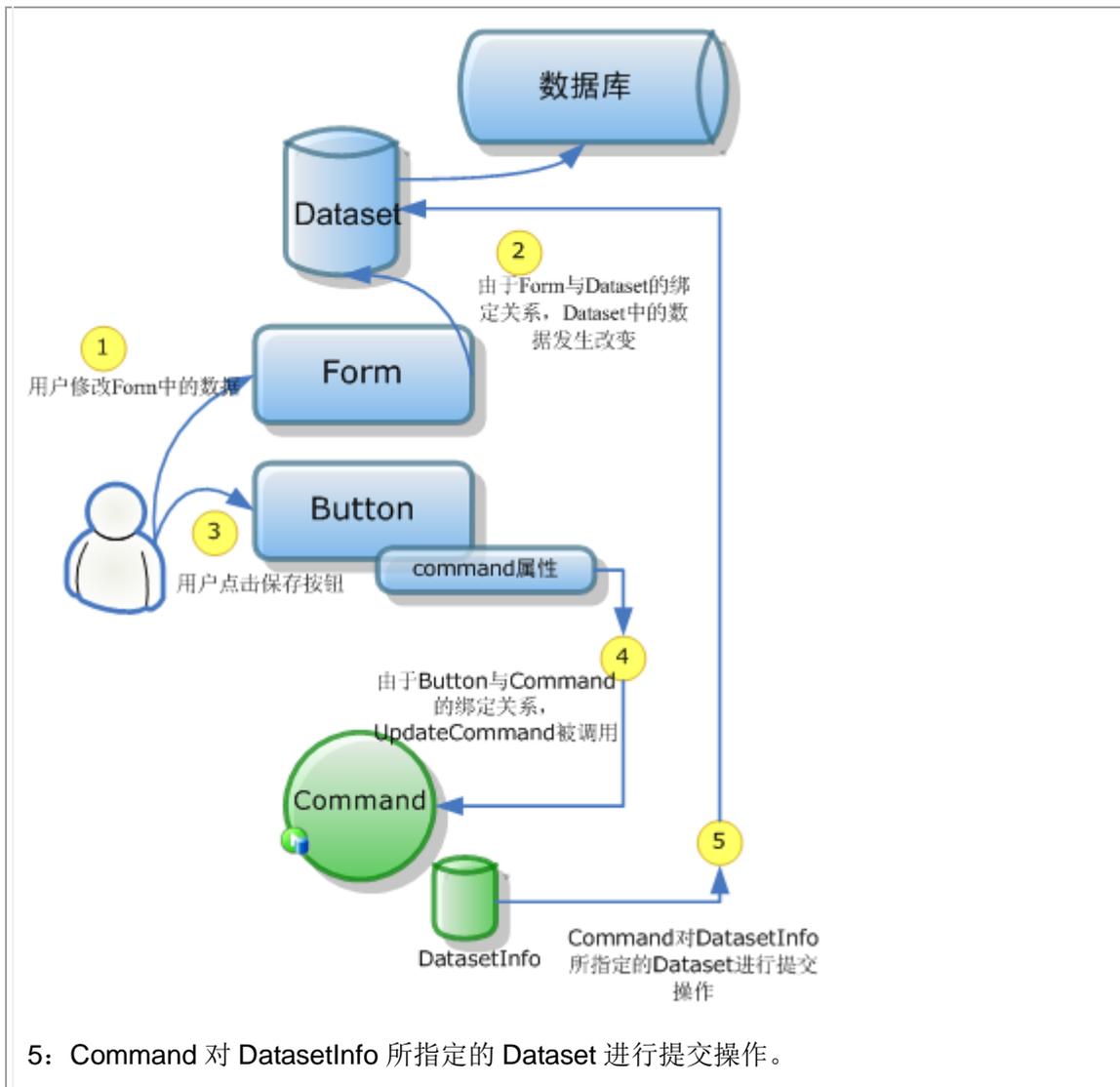
第 4 步设置 `button01` 中 `command` 属性为 `UpdateCommand01`。

第 5 步设置 dataset01 中的主键 keyFields，在这个例子中可以是任意字段。

## dorado 原理：保存的运行时







## 5. 第四杯茶：单表的查询

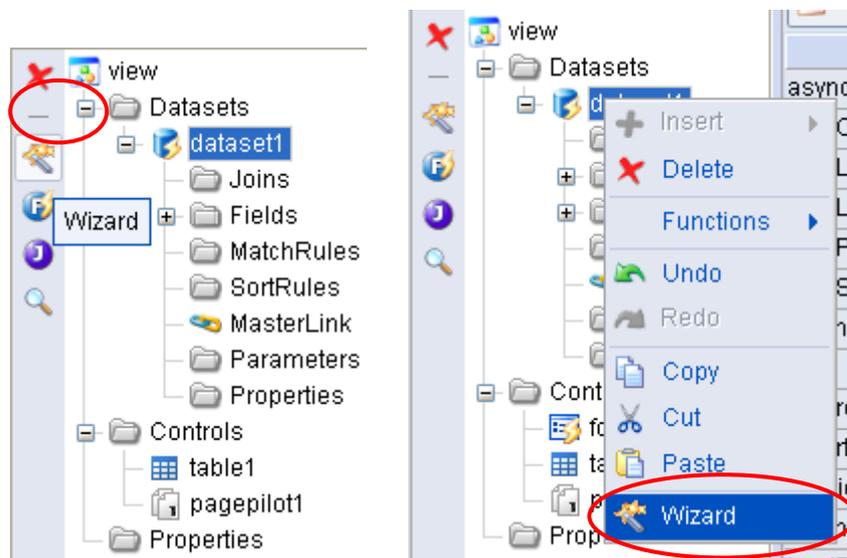
### 5.1. 准备工作

创建 TEST04 文件夹，在其中创建视图模型 Test04，添加一个 dataset1 (AutoSqlDataset) 数据来自 EMPLOYEE 表，再添加一个 table1 (DataTable)，一个 form1 (Autoform)，一个 datapilot1 (DataPilot)，全部与 dataset1 绑定。

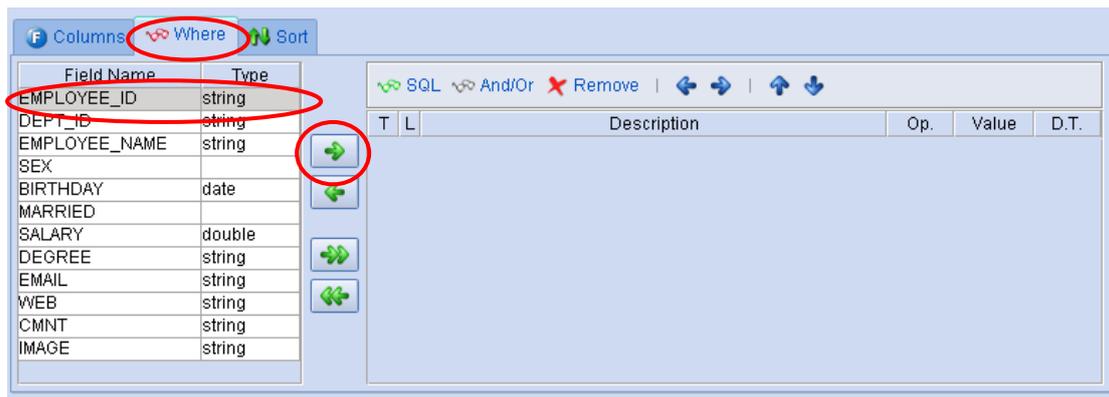
### 5.2. 设置匹配条件

第一步使用向导 (Wizard) 增加 dataset1 的匹配条件。点击左侧的向导 (Wizard)  图标，或者在 dataset1 的右键菜单中选择“Wizard”菜单项。

AutoSqlDataset 的设置查询条件过程实际上是一个拼 SQL 语句的过程。如果不设置任何查询条件，所生成的 SQL 语句相当于“SELECT \*”。



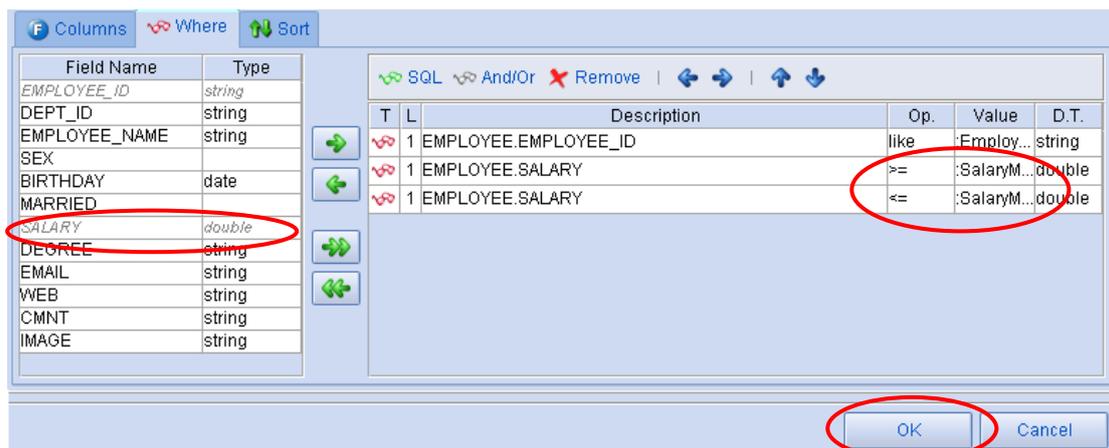
在 AutoSqlDataset Wizard 中，切换到 Where 标签页，选中 EMPLOYEE\_ID 再点击  右键头。



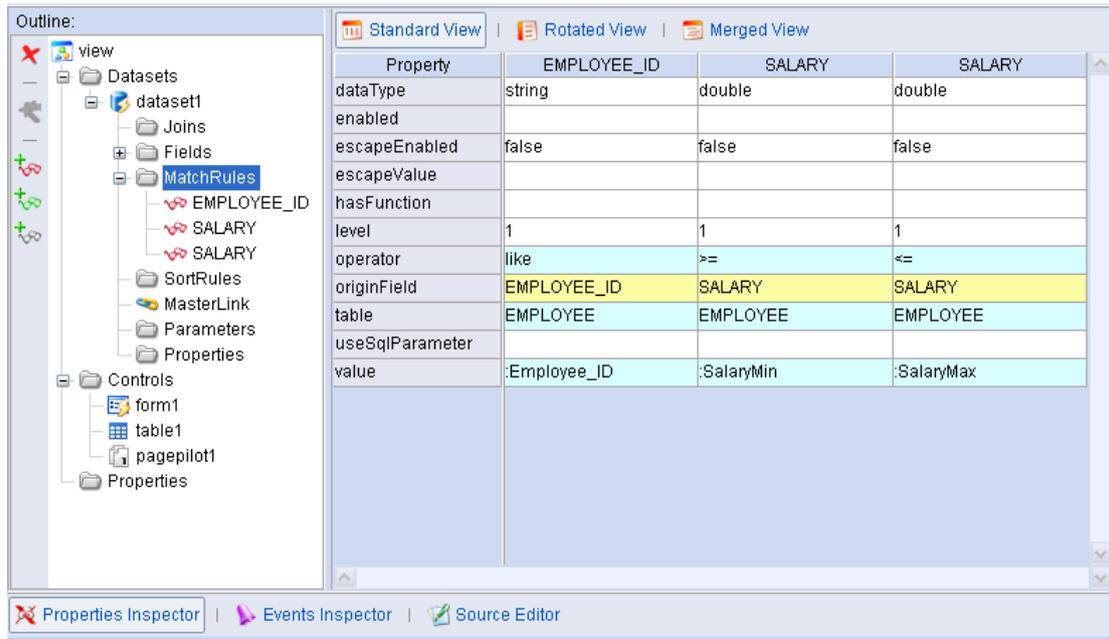
在下拉框 Op. 中选择 “like”，在 Value 中输入 “:Employee\_ID”。



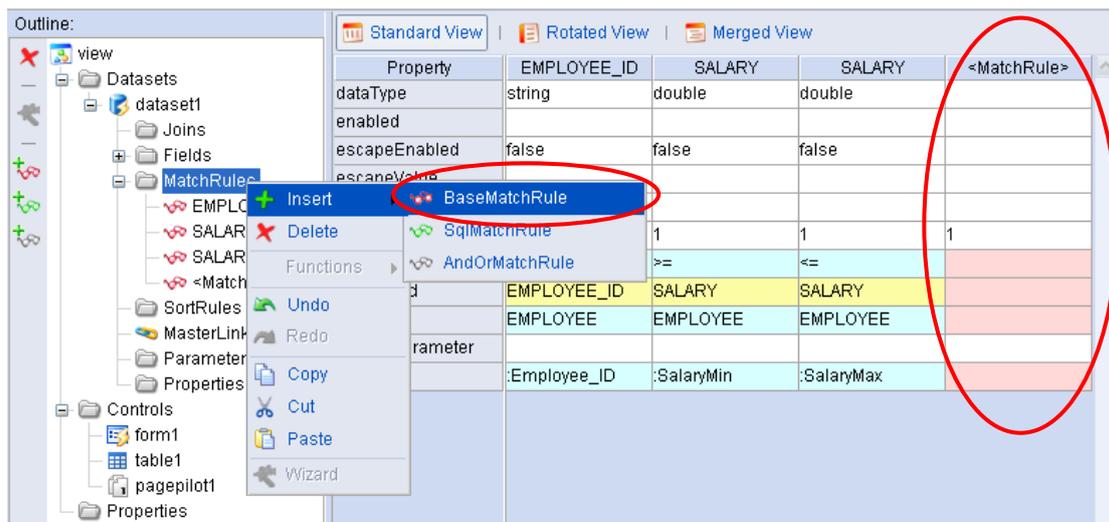
以此类推，再添加两个条件字段选择 SALARY，Op. 与 Value 分别是  $\geq$  :SalaryMin 和  $\leq$  :SalaryMax。点击按钮 OK 保存设置。



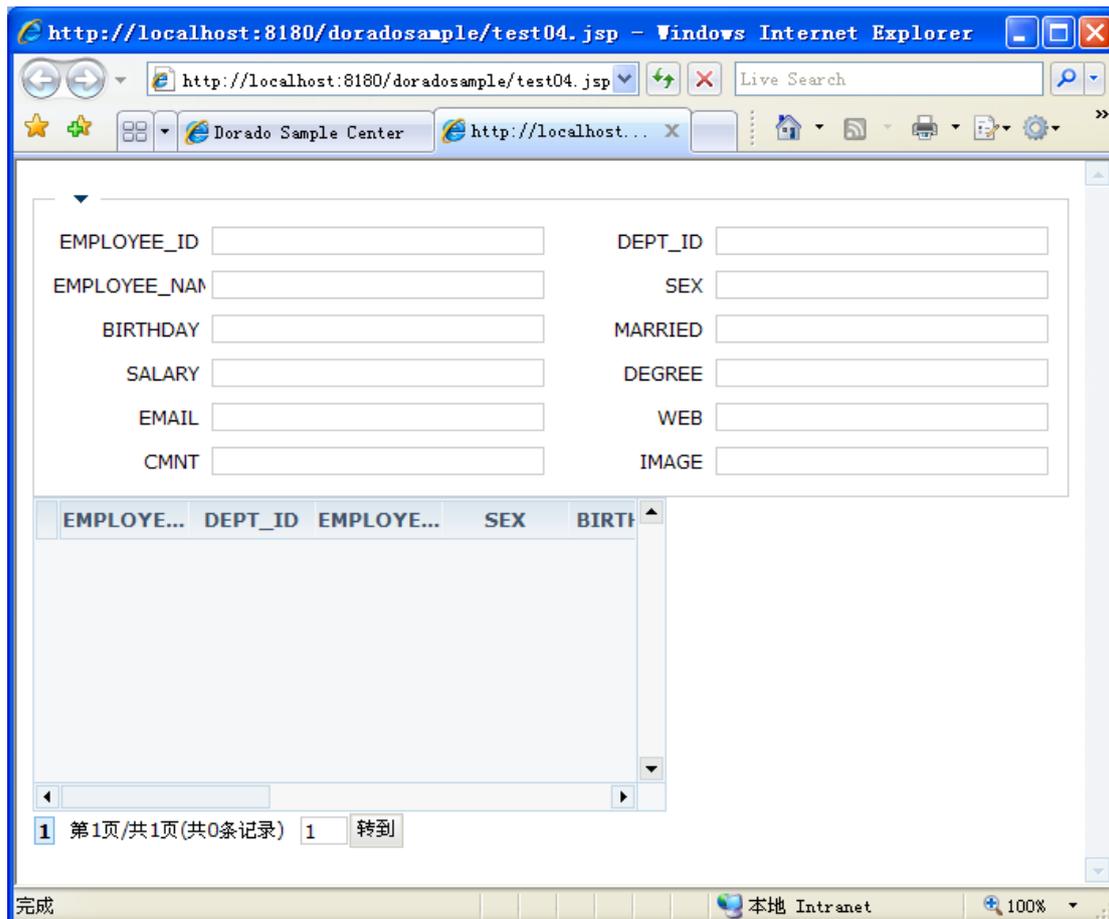
这时，再点击 dataset1 子节点 MatchRules 时，会看到多了 3 个匹配条件 MatchRule ，仔细观察其属性 datatype、operator、originField、table、value 等就是在向导中设置的。



使用向导与使用 MatchRules 的右键菜单（Insert->BaseMatchRule）是相效的。

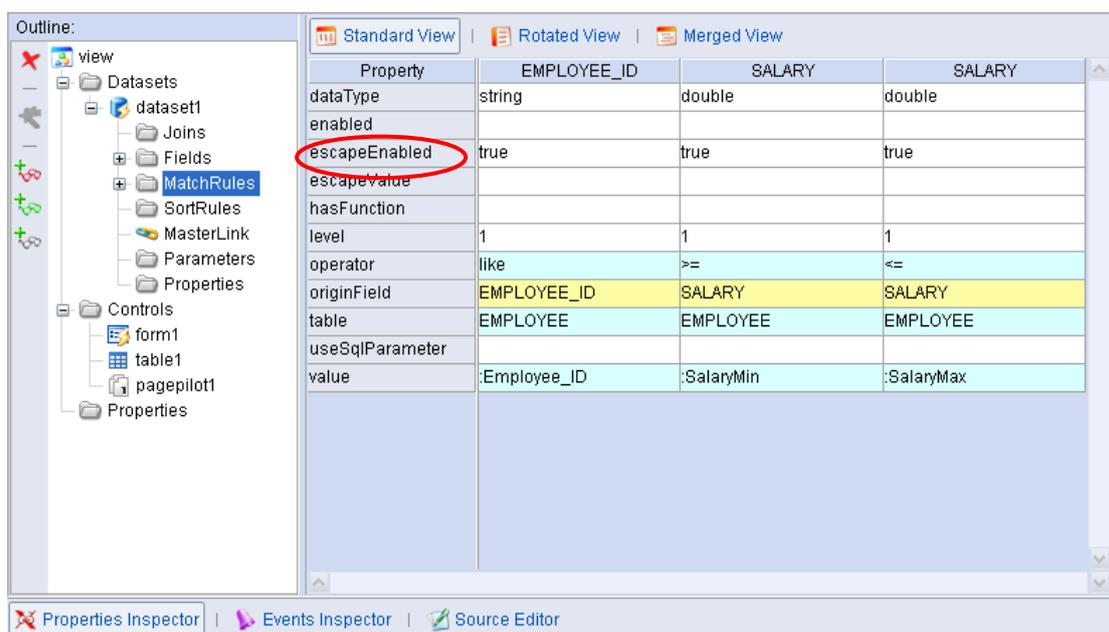


这时生成 JSP 进行浏览，会看到没有数据输出。

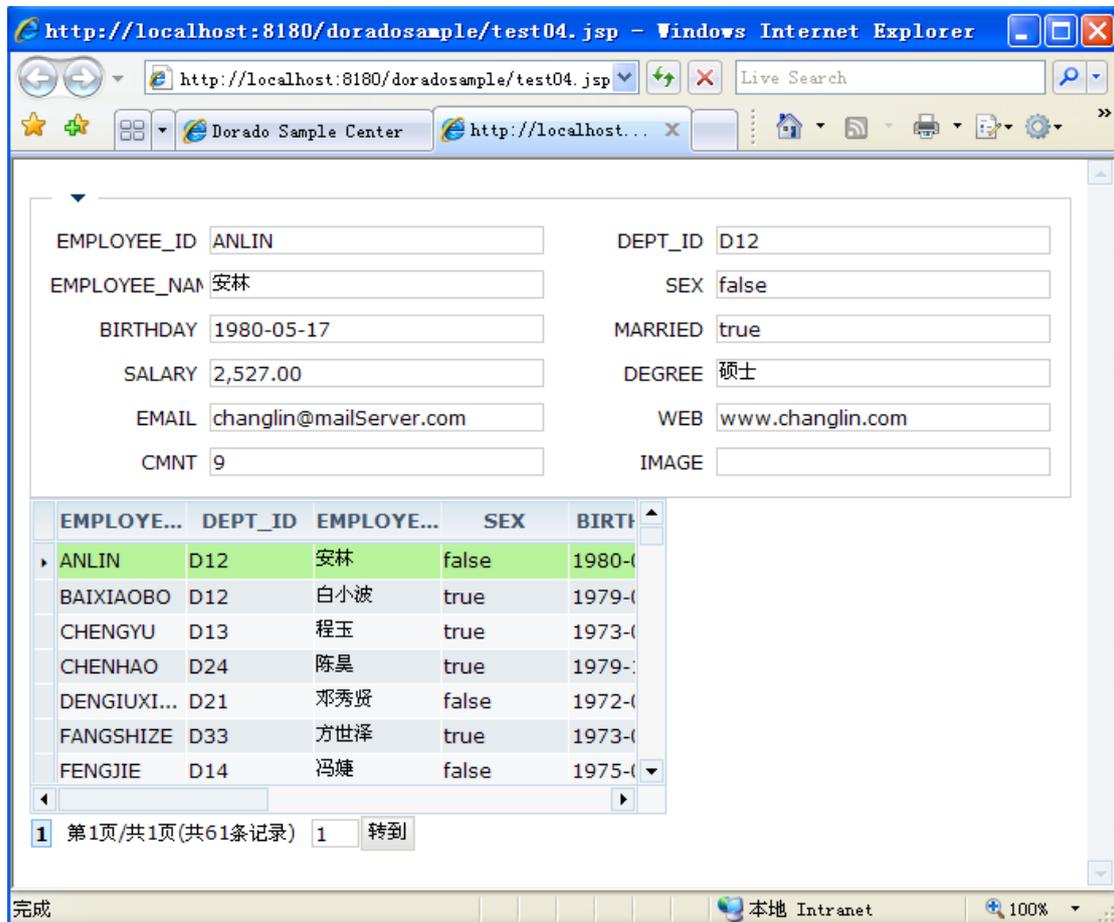


在 MatchRules 中将 `escapeEnabled` 属性设为 `true`。

`escapeEnabled` 设为 `true` 表示允许逃逸，即当该条件没有值时，就不生成相应的 SQL 语句，相当于动态查询。

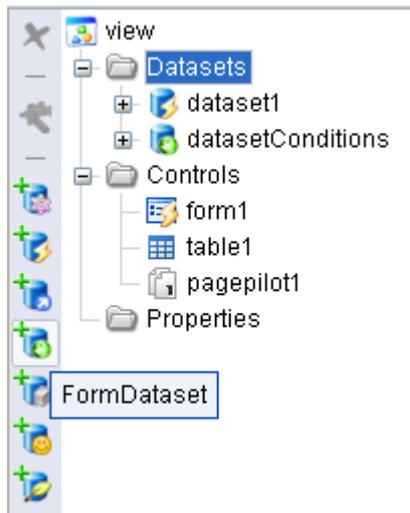


这时再浏览页面，就会看到数据呈现了。



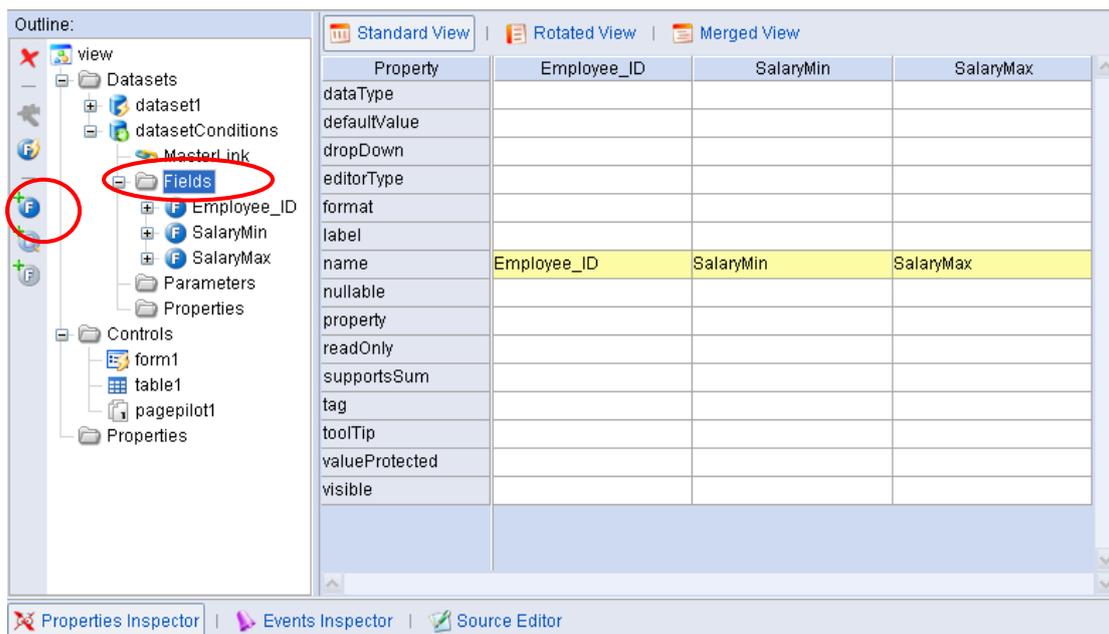
### 5.3. 添加条件数据集

第二步添加一个 FormDataset , 重命名为 datasetConditions。



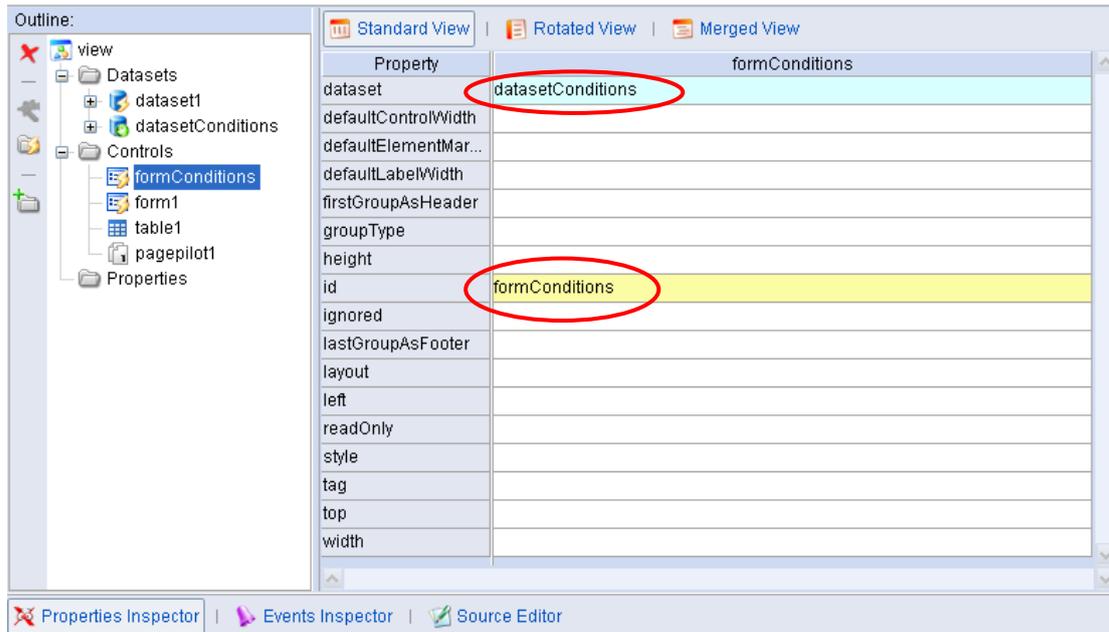
在 `datasetConditions` 子节点 `Fields` 添加 3 个 `Field`，重命名为 `Employee_ID`，`SalaryMin`，`SalaryMax`。

`datasetConditions` 中 `Field` 的命名最好与 `dataset1` 的匹配条件中的 `Value` 的命名相同。不相同也可以，但需要额外添一些做转换的手工代码，这里先不介绍。



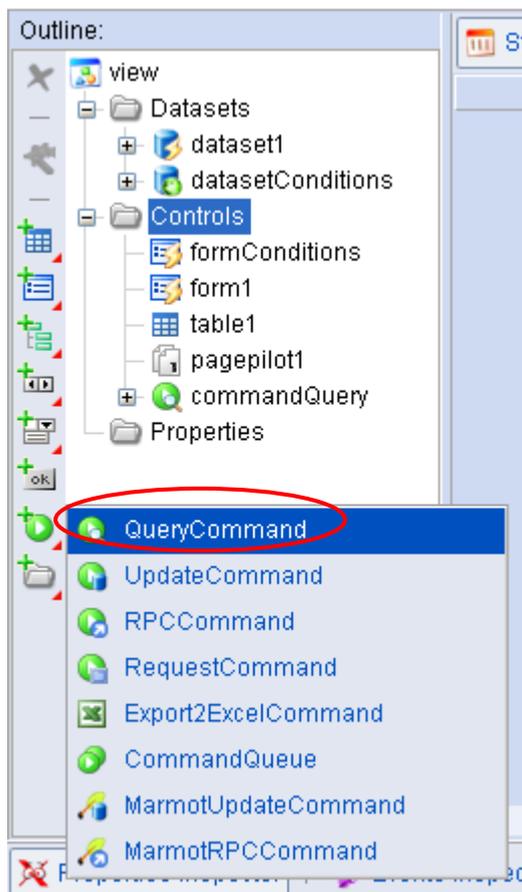
## 5.4. 添加用于条件输入的表单

在 `Controls` 中添加一个 `AutoForm`，重命名为 `formConditions`，设置其 `dataset` 属性为 `datasetConditions`。

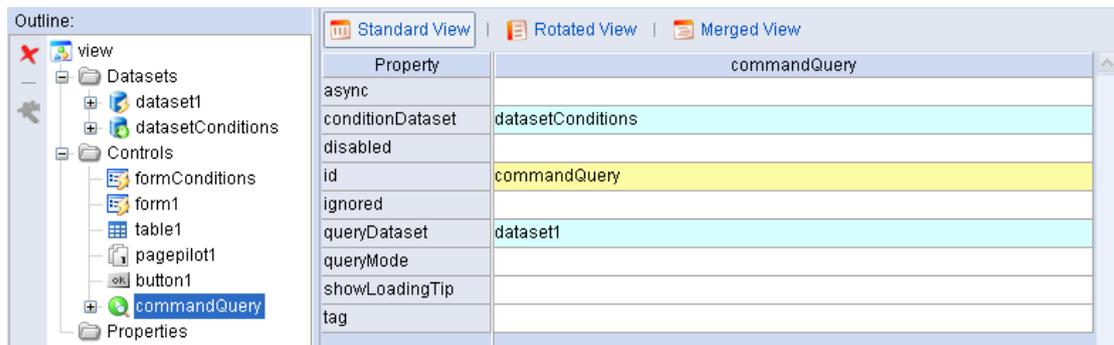


## 5.5. 添加查询命令

添加一个 QueryCommand，重命名为 commandQuery。

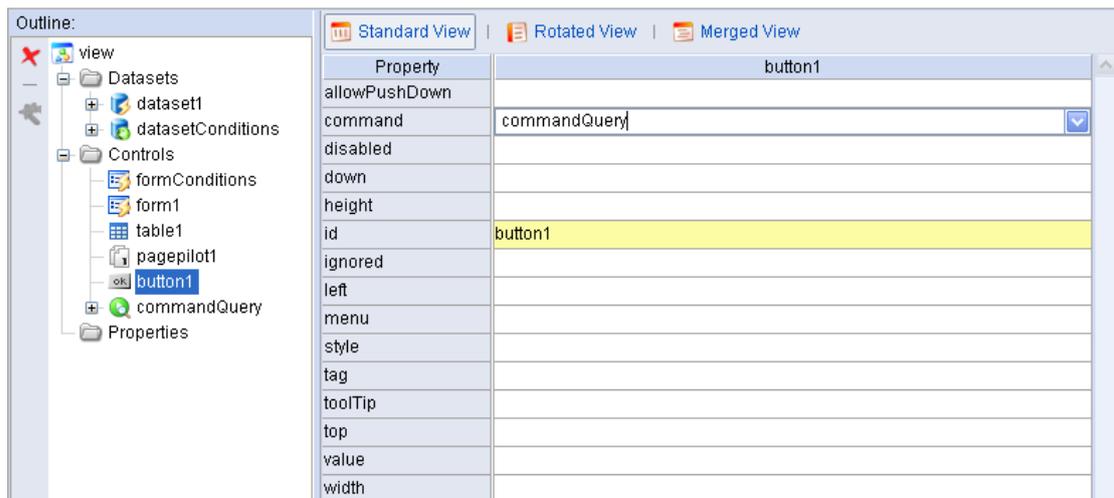


设置 `commandQuery` 的 `queryDataset` 为 `dataset1`, `conditionDataset` 为 `datasetConditions`。



## 5.6. 添加查询按钮

添加一个按钮，设置 `button1` 的 `command` 属性为 `commandQuery`。



## 5.7. 改进

将 `button1` 输出到 JSP, 浏览页面, 在 `Employ_ID` 文本框中输入“%AN%”, 在 `SalaryMin` 与 `SalaryMax` 中分别输入 1000 与 5000, 点击 `button1`, 就得到工号中包含“AN”, 并且 `Salary` 大于等于 1000, 并且 `Salary` 小于等于 5000 的记录。



做一处改进，一个是在 `commandQuery` 的 `beforeExecute` 方法中添加代码。



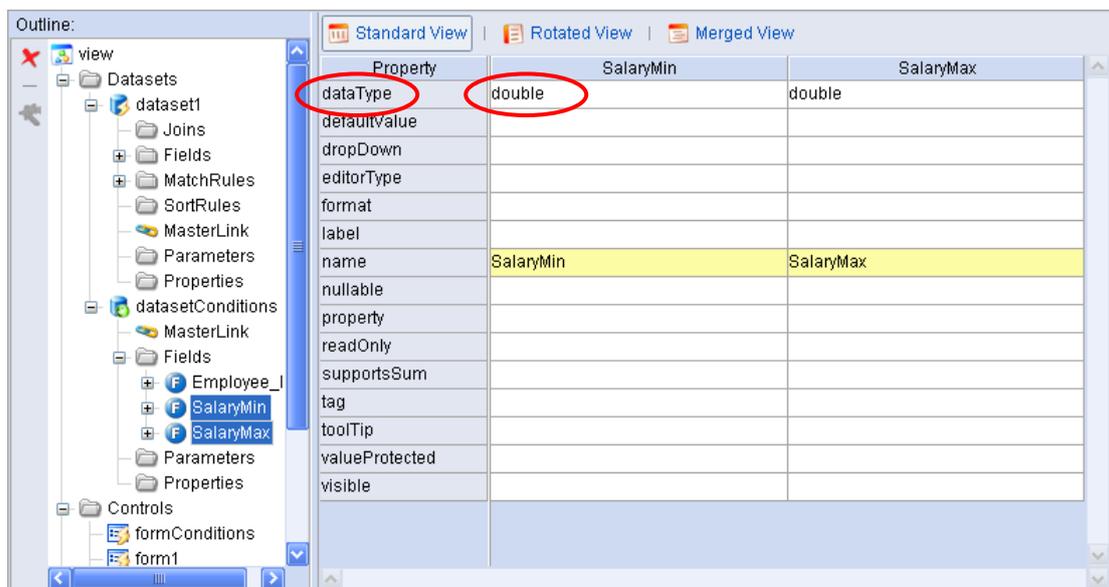
```

var Employee_ID = datasetConditions.getValue("Employee_ID");
if (Employee_ID) {
    command.parameters().setValue("Employee_ID", "%" + Employee_ID + "%");
}
else {
    command.parameters().setValue("Employee_ID", "");
}

```

这样在 Employee\_ID 中输入“AN”即可，不用在前后特别加上 2 个通配符“%”了。

再做一处改进，在 datasetConditions 的 Fields 中的 SalaryMin 与 SalaryMax 的 dataType 属性设为 double。这样在页面上 SalaryMin 与 SalaryMax 就只接入数字型的输入了。

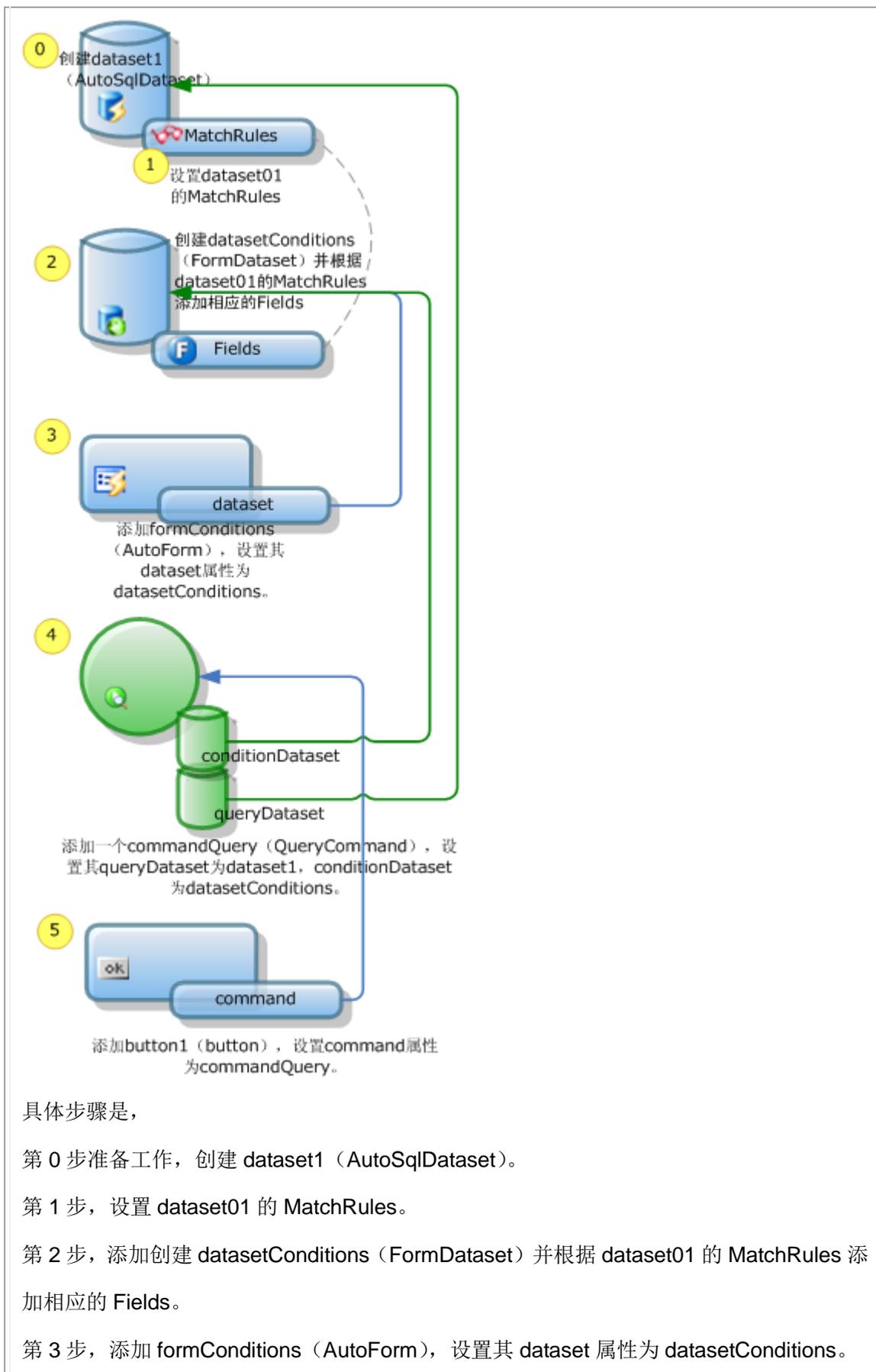


## dorado 原理：dorado 中的查询



### dorado 原理：dorado 中的查询

dorado 中的查询通常需要两个 dataset，一个用于承载查询结果，一个用于承载查询条件。用于承载查询条件的 dataset 通常使用 FormDataset，即其数据是来自于前台页面的，而不是如 AutoSqlDataset 等是来自于后台业务逻辑的。相对于 FormDataset 的 Controls 通常是 AutoForm。查询是与后台发生数据交互的，因此是 button+QueryCommand 的操作方式。



具体步骤是，

第 0 步准备工作，创建 dataset1 (AutoSqlDataset)。

第 1 步，设置 dataset01 的 MatchRules。

第 2 步，添加创建 datasetConditions (FormDataset) 并根据 dataset01 的 MatchRules 添加相应的 Fields。

第 3 步，添加 formConditions (AutoForm)，设置其 dataset 属性为 datasetConditions。

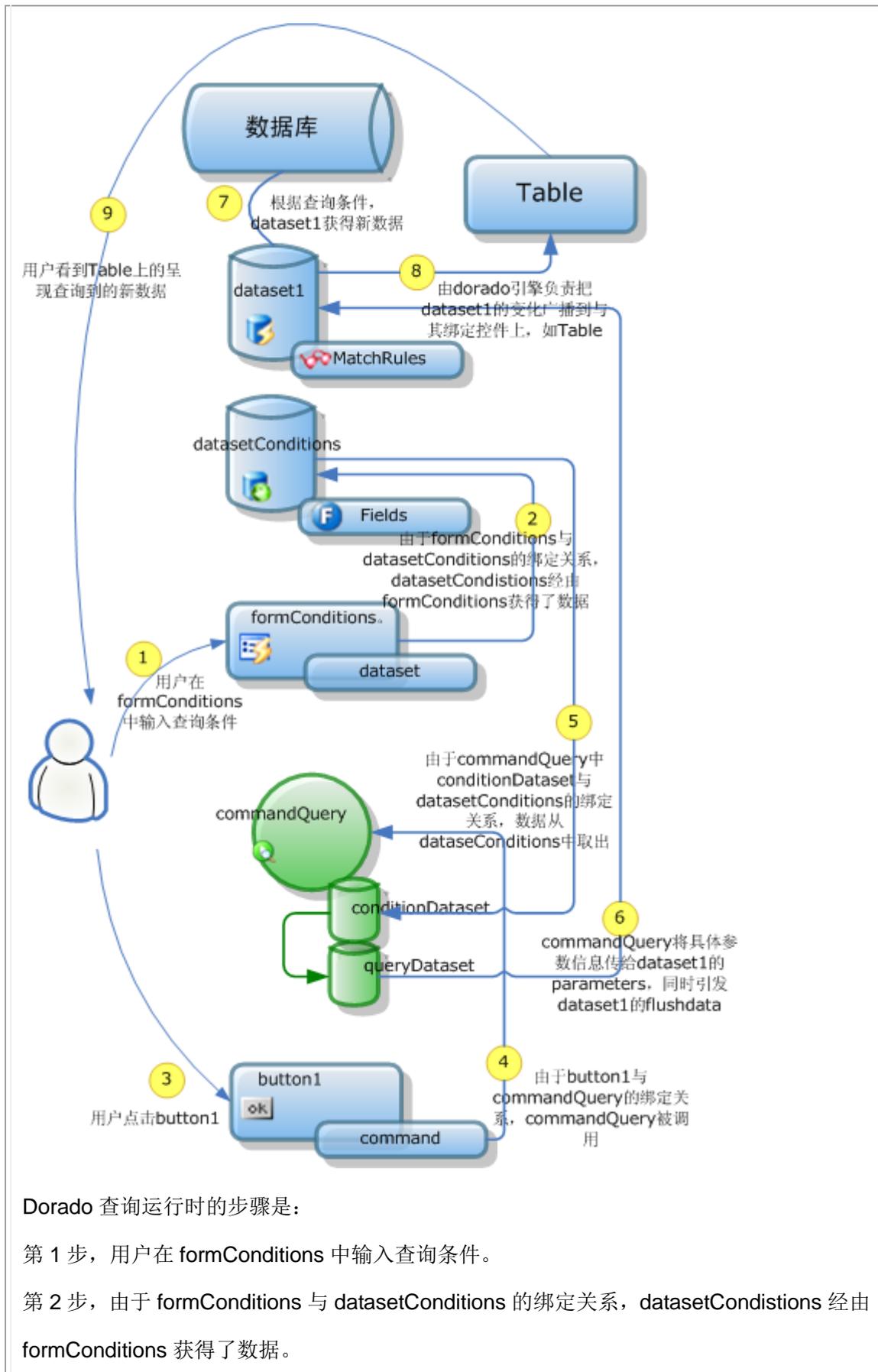
第 4 步, 添加一个 `commandQuery` (`QueryCommand`), 设置其 `queryDataset` 为 `dataset1`, `conditionDataset` 为 `datasetConditions`。

第 5 步, 添加 `button1` (`button`), 设置 `command` 属性为 `commandQuery`。

## dorado 原理: dorado 查询的运行时态



dorado 原理: dorado 查询的运行时态



Dorado 查询运行时的步骤是:

第 1 步, 用户在 formConditions 中输入查询条件。

第 2 步, 由于 formConditions 与 datasetConditions 的绑定关系, datasetConditions 经由 formConditions 获得了数据。

第 3 步，用户点击 `button1`。

第 4 步，由于 `button1` 与 `commandQuery` 的绑定关系，`commandQuery` 被调用。

第 5 步，由于 `commandQuery` 中 `conditionDataset` 与 `datasetConditions` 的绑定关系，数据从 `datasetConditions` 中取出。

第 6 步，`commandQuery` 将具体参数信息传给 `dataset1` 的 `parameters`，同时引发 `dataset1` 的 `flushdata`。

第 7 步，根据查询条件，`dataset1` 获得新数据。

第 8 步，由 `dorado` 引擎负责把 `dataset1` 的变化广播到与其绑定控件上，如 `Table`。

第 9 步，用户看到 `Table` 上的呈现查询到的新数据。



## 文档信息

文档中文名称	喝喝茶 编编程——使用 dorado 基础技术进行 Web 应用开发		
文档英文名称	Master dorado in A Tea Break		
文档内容简介	属于《dorado 飞速入门》范围的技术文档		
文档分享范围	公开文档		
日期	作者	版本	变更说明
2007 年 10 月 20 日	<a href="mailto:Thomas.ding@bstek.com">Thomas.ding@bstek.com</a>	V0.1	创建
2007 年 11 月 14 日	<a href="mailto:Thomas.ding@bstek.com">Thomas.ding@bstek.com</a>	V0.2	增加运行时图序