

利用 ADF 在 10 分钟内创建一个 SQL 查询 Web 客户端

作者：包光磊 (Todd Bao)

<http://www.shoug.info/archives/todd-bao.html>

SH'OUUG

SHANGHAI ORACLE USERS GROUP

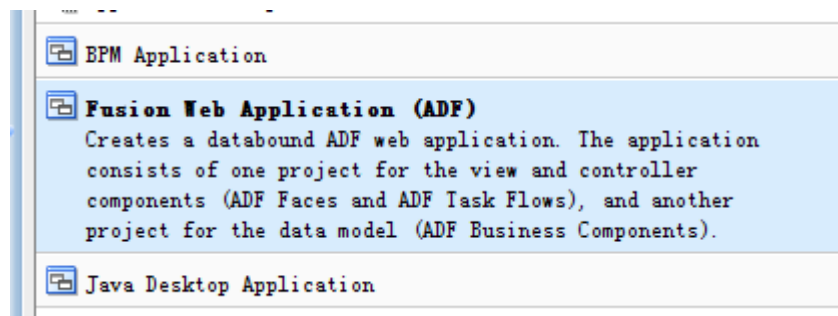
上海 **ORACLE** 用户组

本文适应读者： ADF11g 开发人员

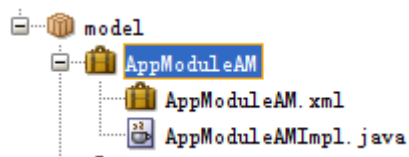
有人怀念 iSQL*Plus 吗？有没有考虑过自己写一个 iSQL*Plus 呢？

本文描述如何通过 ADF 创建一个简单的、执行单一 SQL 查询并展示其结果集的 Web 应用。

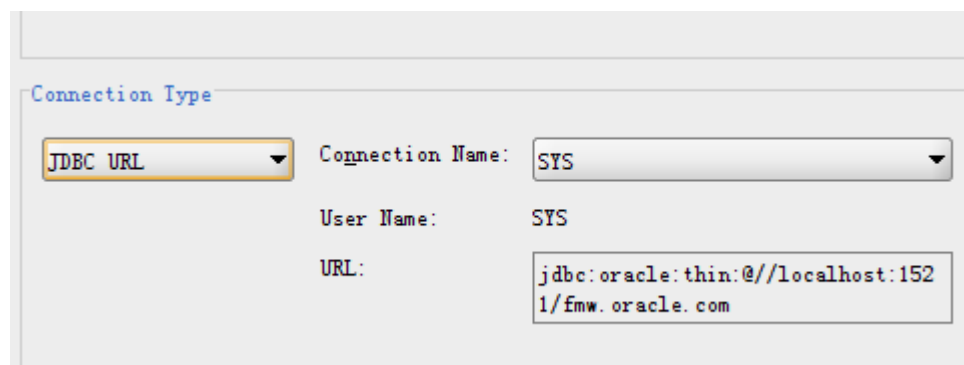
步骤一：创建一个 Fusion Web Application，并建立 Model 和 ViewController 两个项目。



步骤二：在 Model 里创建一个 Application Module，并生成其 Java Class。



步骤三：在 Application Module 的配置里，设置数据库连接为 JDBC URL，这样做可以避免在 WebLogic 上创建显式的 JDBC data source。



步骤四：在 Application Module 的 Java Class 中添加三个方法。描述如下：

1. **clearQuery**: 能够移除一个名为“X”的 ViewObject。你总不愿意在执行了多次查询之后看见 OutOfMemory 冉冉升起吧。此处“X”将是我用来封装任意查询命令的 ViewObject 的名字。

```
public void clearQuery(){
    ViewObject vo = this.findViewObject("X");
    if(vo!=null){
        vo.remove();
    }
}
```

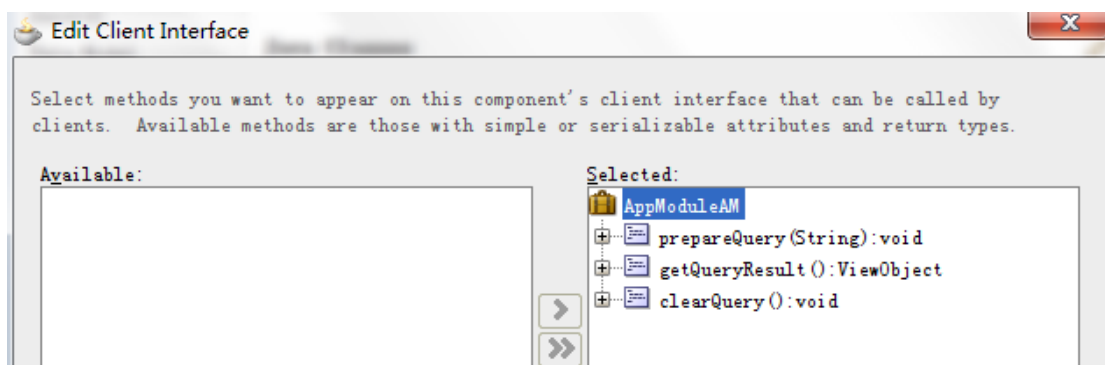
2. **prepareQuery**: 实为创建一个名为“X”的 ViewObject, 此处我们并不需要执行真正的查询, 将来应该利用 ADF Binding 的 Iterator 自动执行。该方法中, 查询作为形参传入。

```
public void prepareQuery(String query) {
    this.clearQuery();
    this.createViewObjectFromQueryStmt("X", query);
}
```

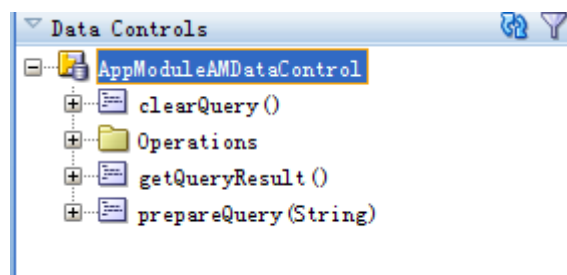
3. **getQueryResult**: 返回查询的结果。

```
public ViewObject getQueryResult(){
    return this.findViewObject("X");
}
```

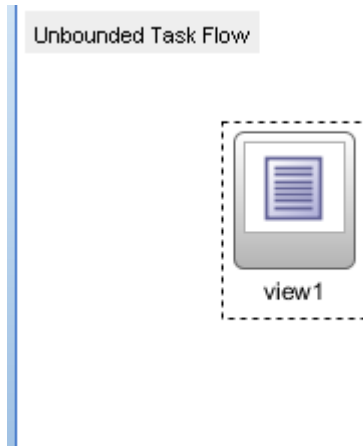
步骤五：将紧前步骤的三个方法全部暴露为 Application Module 的客户端接口。



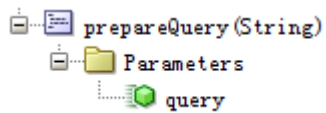
此时你的 Data Control 应该变成这样了：



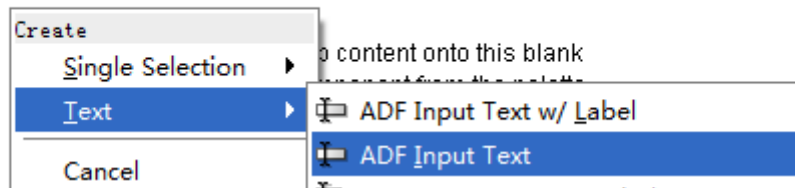
步骤六：在 **ViewController** 项目中创建一个新页面 **view1**。



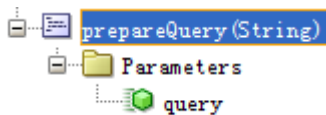
步骤七：将 **prepareQuery** 的形参 **query** 抛入 **view1**。



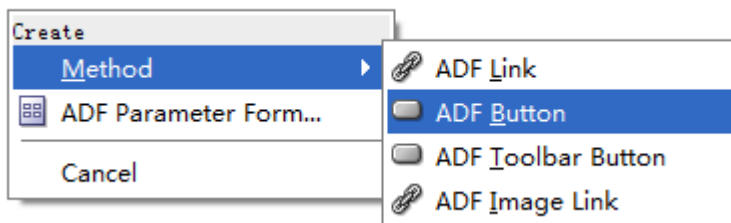
作为 Input Text:



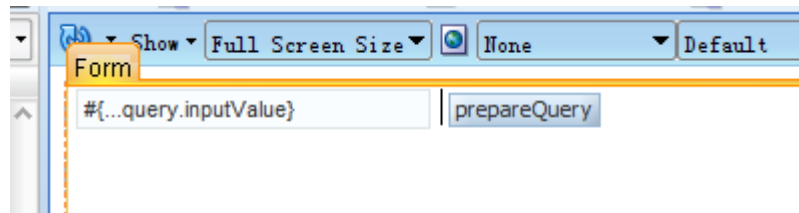
步骤八：将 **prepareQuery** 的方法自身抛入 **view1**。



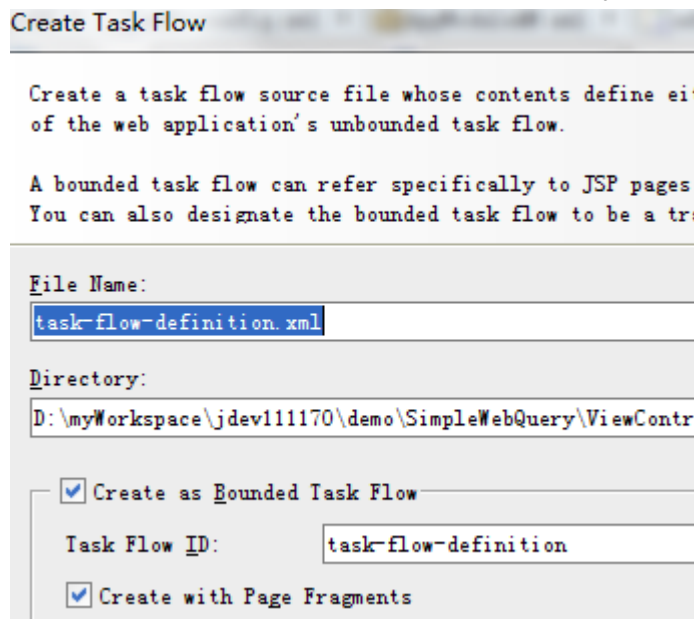
作为 Button:



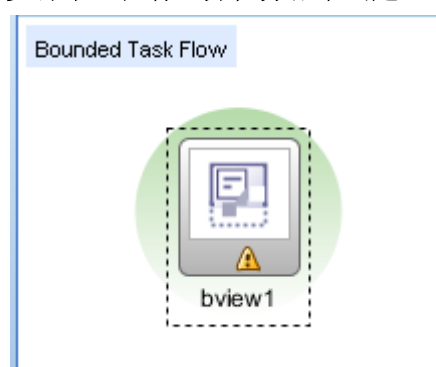
至此，页面看上去是这样的：



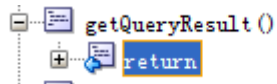
步骤九：创建一个有边界的任务流，页面类型为 jsff。



步骤十：在有边界任务流中创建一个新页面名为 bview1。



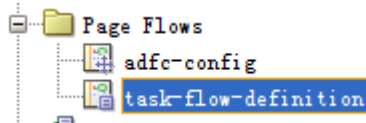
步骤十一：将 `getQueryResult` 的方法返回 `return` 抛入 `bview1`。



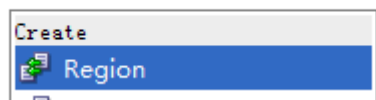
自动作为一个动态表，代码可以自动生成：

```
<af:table rows="#{bindi
    fetchSize="#{
    emptyText="#{
    var="row" row
    value="#{bindi
    selectedRowKe
    selectionList
    rowSelection=
    <af:forEach items="#{
        <af:column headerTe
            sortable
        <af:outputText ve
    </af:column>
    </af:forEach>
</af:table>
```

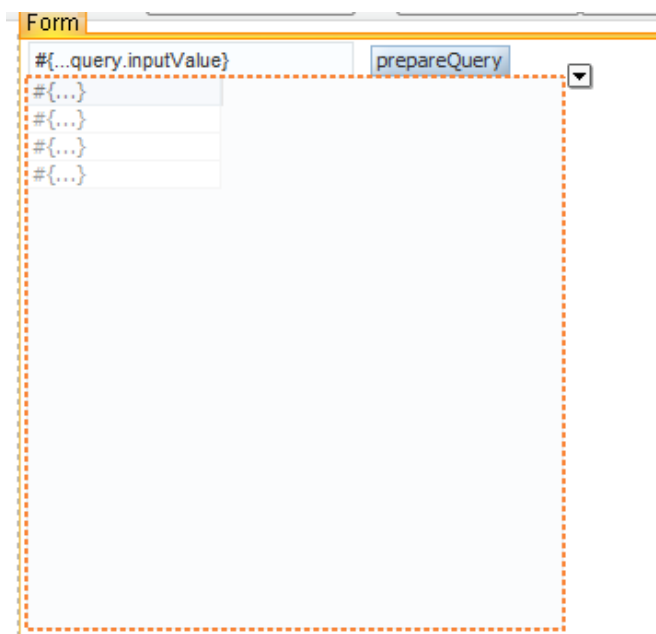
步骤十二：将整个有边界的任务流作为 `Region` 抛入 `view1`。



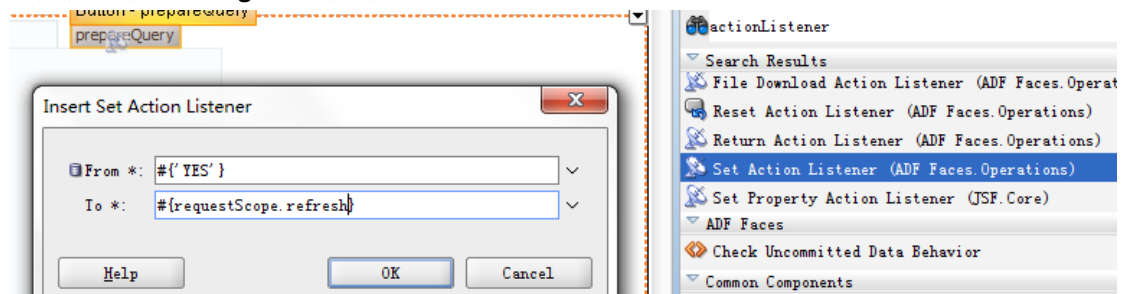
作为 `View1` 中的一个 `Region`，用来展示查询的结果：



此时，你的 `View1` 应该长成这样了：



步骤十三: 为 `prepareQuery` 按钮添加一个 `setActionListener`, 为一个变量赋值, 通过这个变量控制结果显示 `Region` 的刷新。



步骤十四: 在 `view1` 的页面定义文件中找到作为 `Region` 的任务流设置, 添加 `RefreshCondition` 这一属性。

```
<taskFlow id="taskflowdanySimpleType"
RefreshCondition="#{requestScope.refresh=='YES'}"
taskFlowId="/WEB-INF/task-flow-definition.xml#task-flow-definition"
activation="deferred"
xmlns="http://xmlns.oracle.com/adf/controller/binding"/>
```

现在, 你可以运行 `view1` 了。检查一下效果。



输入 `select * from hr.employees`, 点击 `prepareQuery`:

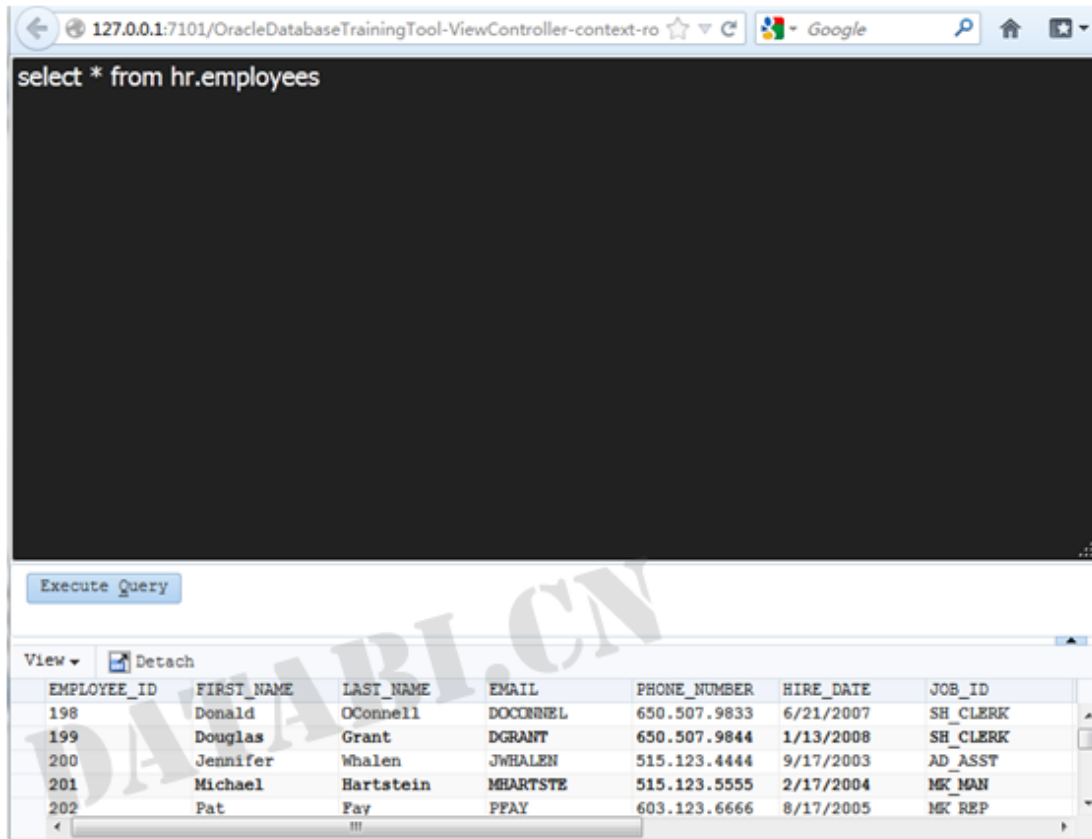
EMPLOYEE_ID	FIRST_NAME	LAST_NAME
198	Donald	OConnell
199	Douglas	Grant
200	Jennifer	Whalen
201	Michael	Hartstein
202	Pat	Fay
203	Susan	Mavris
204	Hermann	Baer
205	Shelley	Higgins
206	William	Gietz
100	Steven	King
101	Neena	Kochhar
102	Lex	De Haan
103	Alexander	Hunold
104	Bruce	Ernst
105	David	Austin
106	Valli	Pataballa

恭喜你，一个简单的 web SQL 查询工具就这样完成了。请看一下手表，过去了几分钟？

当然，我的这个例子是简化到极限的 ADF 例子，甚至没有使用显性声明的 managed bean。还有很多问题需要处理、许多逻辑需要梳理，比如处理大结果集时需要在 ViewObject 上做出调整，实现分页；为了使分页的接口更人性化至少需要 managed bean 的帮助等等。

还有一个彩蛋等着大家，在同一个页面上你第二次执行查询试一下，有什么问题？尝试着解决一下。

最后，稍微和布局控件交涉一下你当然会得到漂亮得多的界面。



例子中虽然暴露了 clearQuery 方法，但是直接调用它的只用 prepareQuery。读者可以参考步骤八通过类似方式，把 clearQuery 作成一个动作控件，随时根据用户的需要清理 ViewObject。

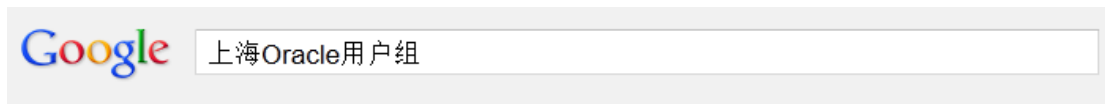
郑重声明：本文仅代表个人观点，不代表 Oracle 公司。

包光磊

新浪微博 <http://weibo.com/toddbao>

我主持的国内 ADF/SOA 论坛：<http://www.databi.cn/forum-57-1.html>

SHOUG 的网址: www.shoug.info 找我们:



Web Images Maps More Search tools

About 234,000 results (0.34 seconds)

[上海Oracle用户组 | SHOUG, 走近全系Oracle技术和数据库专家](#)

www.shoug.info/ Translate this page

5 days ago – SHOUG的全称是ShangHai Oracle Users Group, 中文为上海Oracle用户组。SHOUG的成员仅仅局限于上海地区吗? 上海是国际化大都市, 我们 ...

[SHOUG成员 | 上海Oracle用户组](#)

www.shoug.info/archives/category/shoug成员 Translate this page

2 days ago – 在Oracle数据库领域具有比较扎实的理论基础、较丰富的实践经验、处理过较多的Oracle性能优化和数据恢复案例, 乐于分享, 学习Oracle技术。