

基于J2EE技术实现J2ME嵌入式系统访问的网络数据库

张 军

(山东体育学院 基础理论系,山东 济南 250063)

摘 要: 随着手机功能的增强,通过无线网络访问企业数据库的移动数据库应用必然会进一步发展以提供更多的功能,同时其开发也会变的越来越复杂。本文通过研究J2ME和J2EE技术架构的特点,提出一套完整的移动设备访问数据库解决方案,详细解析了移动客户端和服务端的设计方案,对移动查询系统的架构、J2ME客户端的实现和使用Hibernate连接数据库进行了完整的剖析。

关键词: J2ME J2EE 嵌入 Hibernate

1.前言

数据库操作是现在项目开发的根本。现在不管用什么语言编写程序,一般都会涉及数据库的连接,那么在使用J2ME开发平台编写手机应用程序的过程中也避免不了要涉及数据库的连接。J2EE是一种用于建立服务器应用程序的系统平台,它被定义为开发、部署、管理多层以网络和组件技术为基础、以服务器端为中心的企业级应用提供的开放的标准平台。在这里本文主要用到了Hibernate。Hibernate是一个高性能的ORM持久性和查询服务。Hibernate是Java应用和关系数据库之间的桥梁,它负责Java对象和关系数据之间的映射。Hibernate内部封装了通过JDBC访问数据库的操作,向上层应用提供了面向对象的数据访问的API。以下是在开发J2ME过程中的一点体会,希望能得到大家的指点。

2.系统设计思路

在这样的中大型项目中,首先要设计完善的公共类,包括可继承的控件基类、资源类、Hibernate业务逻辑类、Hibernate配置文件等。公共类要提供各种所需的API让程序调用。Hibernate业务逻辑类主要提供访问数据库中指定行列数据、保存指定记录到数据库、排序现有记录等的各种API。公共类还要求高度独立,可以移植到其它类似数据席的应用中去。其次,设计统一的应用层架构,使不同的开发人员在相同的模式下进行开发,便于管理和查错。

在本文讨论的项目中,描述一个最简单流程为:用户在手机端进入某个查询界面。通过选择和录入等方式设置查询条件。查询条件被发送至服务器。在服务器端有相应Servlet类接收手机发送的数据,调用Hibernate业务逻辑类操作数据库,并将返回的结果集返回到手机客户端并显示。流程如下图:

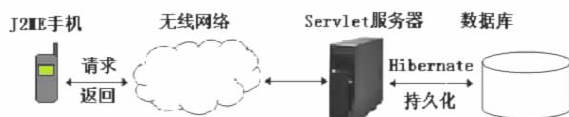


图1 手机与服务器交互流程图

3.手机通过Servlet与Web服务器通信

图2表示了手机客户端程序Midlet通过与Web服务器上Servlet的交互完成访问网络数据库示意图:



图2 手机访问网上数据库示意图

基于J2ME平台开发手机客户端程序,支持打开网络连接、传送数据、访问远程服务器但,并不支持直接访问数据库的能力。因此,我们需要编写一套服务器端程序来接受手机客户端的请求指令,通过解析请求、调用Hibernate业务逻辑类、响应结果的方式把数据再传回给手机。

4.JAVA手机访问数据库的实现

4.1实例描述

下面是一个J2ME与Servlet结合实现仿真手机查询计算机考试成绩的例子。用户在手机端输入要查询的学生学号、密码, Midlet 程序将参数传送Web服务器端Servlet,Servlet程序根据传来的学号、密码,通过Hibernate连接数据库服务器mysql,将读取的学生成绩信息以UF8格式编码传送给手机端Midlet程序。以下为Midlet和Servlet的主要代码。

4.2 开发手机客户端Midlet

```

public void commandAction(Command c, Displayable s){
    Form form1 = new Form("查询结果"); //建立查询结果
    界面
    display.setCurrent(form1);
    StringBuffer sb=new StringBuffer();
    String stid=tf.getString();//获取文本框中所输入的学号
    String url="http://127.0.0.1:8080/servlet/ServletScore?stid="+stid;
    HttpURLConnection conn =null; DataInputStream in=null;
    try{ conn =(HttpURLConnection)Connector.open(url);
        in= conn.openDataInputStream();
        String ok=in.readUTF();
        if(ok.equals("find")){//将查询结果放入sb
            sb.append("学号"+in.readUTF()+"\n");
        }else {sb.append(in.readUTF()+"\n");}
    }catch (Exception e){ sb.append(e.toString());}
    finally{
        try{
            if(in!=null)in.close();
            if(conn!=null)conn.close();
        }catch(java.io.IOException ioe){}
    }
    form1.append(sb.toString());//显示成绩信息
}
    
```

4.3 开发服务器端响应servlet

Servlet的doPost方法代码如下:

```

DataOutputStream out = new DataOutputStream(new
    BufferedOutputStream(response.getOutputStream()));
String stid=request.getParameter("stid");
TestUser tu = new TestUser();
User user = new User();
User=tu.getUser(stid);
If(user != null){
    String tycj =user.getTycj();
    String whcj =user.getWhcj();
    String ok="find";
    out.writeUTF(ok);
}
    
```

```
out.writeUTF(tycj);
out.writeUTF(whcj);
}
```

4.4 Hibernate连接MySQL数据库的配置

Hibernate配置文件主要用于配置数据库连接和Hibernate运行时所需的各种属性,这个配置文件应该位于应用程序或Web程序的类文件夹classes中。Hibernate配置文件支持两种形式,一种是xml格式的配置文件,另一种是Java属性文件格式的配置文件,采用“键=值”的形式。建议采用xml格式的配置文件。xml配置文件可以直接对映射文件进行配置,并由Hibernate自动加载,而prop-erties文件则必须在程序中通过编码加载映射文件。两种配置方式的使用方法类似,本文主要以第一种方式为主详细介绍如下:

Hibernate默认的xml格式的配置文件名称为hibernate.cfg.xml。下面是连接MySQL的Hibernate的xml配置文件。

```
<session-factory>
<property name="connection.driver_class">
com.mysql.jdbc.Driver
</property>
<property name="connection.url">
jdbc:mysql://localhost:3306/stumanage
</property>
<property name="connection.username">root</property>
<property name="connection.password">root</property>
<property name="dialect">
org.hibernate.dialect.MySQLDialect
</property>
<property name="hibernate.transaction.factory_class">
org.hibernate.transaction.JDBCTransactionFactory
</property>
<property name="show_sql">true</property>
<mapping resource="UserForm.hbm.xml"/>
</session-factory>
```

在上面的代码中,"localhost"代表本地MySQL服务器,如果想连接其他服务器可以修改为要连接的MySQL服务器的名称;"stumanage"为要连接的数据库名称;"root"为登录用户名;"root"表示用户密码;"org.hibernate.transaction.JDBCTransaction

"Factory"表示采用JDBC的事务处理机制;"UserForm.hbm.xml"为持久化类对应的映射文件名称。

4.5 hibernate-MySQL配置文件及简单应用(CRUD)

User.hbm.xml 配置如下:

```
<hibernate-mapping>
<class name="test.hibernate.User" table="user">
<id name="id" column="id" type="int">
<generator class="increment" />
</id>
<property name="name" column="name" type="string"
not-null="true" />
<property name="password" column="password" type="
string" not-null="true" />
```

```
</class>
</hibernate-mapping>
```

DAO类核心代码如下:

```
public User getUser(String User_id){
Session session = sessionFactory.openSession();
Transaction tx = null; User user = null;
try{
tx = session.beginTransaction();
Query query = session.createQuery("from User where id =
" + User_id);
List<User> list = query.list();
Iterator<User> it = list.iterator();
while (it.hasNext()){ user = it.next();}
tx.commit();
} catch (Exception e) { tx.rollback();
} finally{ session.close();}
return user;
}
```

4.6 程序运行结果

编写编译servlet程序并放到相应的servlet目录。启动Web服务器,启动mysql数据库服务器,运行J2MEWTK的ktoolbar,建立项目编写手机端代码,编译后选择defaultphone手机模拟器,运行界面如图3所示,在第一个界面输入要查的学号,点击OK,即可访问服务器端ServletScore,获得查询结果。



图3 模拟手机运行效果图

5. 结语

目前基于J2ME手机的移动信息管理和数据应用都比较简单,但随着网络和硬件功能的增强,出现类似于台式机数据库系统复杂度的移动应用将会越来越多。本文提出了一种轻量级的移动电子商务开发架构,充分利用J2ME和J2EE技术各自的特点,将两者有机结合起来,开发出轻便的J2ME客户端和J2EE服务器端,从而来满足企业级商业平台开发的需要,对移动数据信息集成具有一定的实用参考意义。

参考文献:

- [1]唐润华,章璐,刘外喜.基于J2ME-J2EE移动集成查询系统的设计与实现.科学技术与工程,2008,8,(1):87-89.
- [2]唐自仰,张克君.基于J2ME/Servlet的手机聊天系统.计算机工程,2009,35,(10):262-263.
- [3]宋汉增,沈琳.利用Hibernate对象持久化服务简化Java数据库访问.计算机应用,2003,23,(12):135-137.