Researchment and Realization Based on Android Database Application Technology

Ming Xu, XinChun Yin, Jing Rong
College of Information Engineering Yangzhou University
YangZhou, China
e-mail: xuming@yzu.edu.cn

Abstract—**This paper puts forward and realizes local model SQLite and remote model SQL Server two different database processing technology based on Android system according to mobile platform for database technology application deep-demand. Meanwhile, this paper expounds the function of WebService when Android system access SQL Server database. The approach proposed in this paper can be applied to the mobile terminal platform and mobile business management system smoothly, with the better advantage of commodity and technical maneuverability.**

Keywords-component:Android; database technology; SQLite; SQL Server;WebService

I. INTRODUCTION

As the popularization of smart phone, Android platform has been admired by more and more people, for its features of open-source, convenience, etc. The application software development in Android mobile phone is becoming more and more important. Data-processing is one of the important parts in application software development. Data-processing can hardly do without database technology. But because the limit of hardware condition for mobile itself, mobile phone cannot install and run large DBMS like most general PC. Therefore, how to realize the database application technology in Android system environment has become one of the important research subjects in mobile application software development. This paper is very aiming at introducing mobile database application technology.

II. A BRIEF OF ANDROID DATABASE TECHNOLOGY

This paper takes SQLite and SQL Server as examples to introduce two kinds of general database access methods: local access method and remote access method, under Android system environment, and their realization.

A. Local access method and SQLite

The data-processing of mobile phone works in local way due to the expenditure on internet, poor speed and the condition limit by mobile itself.

SQLite is self-contained database management system in Android system. It is a kind of lightweight database systems. Because of little support for system and external demand only, it can be embedded into hardware equipment easily. Less memory support and the ability of data reading/writing directly make it as Android mobile system good local data management tool.

Usually, SQLite database is stored in data/data/< project folder >/databases/

The smart, light features of SQLite will become helpless immediately when it comes to mobile terminal based on Web Internet remote database access support.

B. Remote access method and SQL Server

Today, to realize data remote accessing and data processing is one of indispensable elements in mobile application with the rapid development of Internet technology as well as mobile speed increased and function improved.

SQL Server, due to its ease of use, good scalability and cross-platform compatibility, making it singled out in the field of database. It has become the one of most popular database systems. In addition, internet remote data accessing, Web site data management makes it become widely used data management and maintenance tools.

Because SQL Server occupies large storage space, and consumes too much system resource, it is inconvenient to be applied in mobile system. Using the remote connecting SQL Server database technology, which can be achieved the ability to access database in the mobile terminal timely, at any time, query and update relevant data information dynamically.

Android system cannot connect to SQL Server database directly. It must transfer through the WebService middleware technology in order to realize the database accessing.

III. RELATED WORK

Let’s introduce the realization of local access method and remote access method of Android system from the angle of database connection in the following. Take a simple login application to show how to complete the specific database connection.

A. Local method connect SQLite

Using SQLite connect local database system.

Environment Building: Android4.0.4.

The steps to connect with database system using SQLite as follows:

First, create an object of the SQLiteDatabase.

Second, call the method connectDB(), which used to connect database.

When the first time you login, there is not any data in the database, so we should use the method insert() to insert the data into the database. Notice, annotation the insert() method when you login the second time, or you will insert the data again.

The code realized as follows:

```
   public class SQLiteActivity extends Activity{
     SQLiteDatabase sld;
     @Override
     public void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    final Button okButton =
(Button)findViewById(R.id.Button01);
    final Button cancelButton =
(Button)findViewById(R.id.Button02);
    final EditText name =
(EditText)findViewById(R.id.EditText01);
    final EditText password =
(EditText)findViewById(R.id.EditText02);
   connectDB(); //connect database
   okButton.setOnClickListener(new
View.OnClickListener() {
   public void onClick(View v) {
   String nameStr = name.getText().toString();
   String pawStr = password.getText().toString();
if(nameStr==null||pawStr==null||nameStr.equals("")||pawStr.
equals("")) {
        Toast.makeText(SQLiteActivity.this, "name or
password can not be empty",
Toast.LENGTH_SHORT).show();}
   else {insert();//before the first login,you should insert the
name and password to the table.
   boolean flag = query(nameStr, pawStr);
   if(flag) {setContentView(R.layout.lifeencyclopedia);}
   else {Toast.makeText(SQLiteActivity.this, "name or
password is wrong.", Toast.LENGTH_SHORT).show();
   }}}});
   cancelButton.setOnClickListener(
     new View.OnClickListener() {@Override
       public void onClick(View v) {
        name.setText("");
        password.setText("");}});
   }
   public void connectDB(){
   try{sld=SQLiteDatabase.openDatabase(
               "/data/data/com.yzu.activity/mydb", //the
path of the database mydb
        null, //CursorFactory
        SQLiteDatabase.OPEN_READWRITE|SQLiteData
base.CREATE_IF_NECESSARY //read,write and create if
not exsists.);
   String sql="create table if not exists users(name
varchar(20), password varchar(20))";
   sld.execSQL(sql);
   }catch(Exception e){
     Toast.makeText(this, "There's something wrong in the
database:"+e.toString(), Toast.LENGTH_SHORT).show();}}
     //query the table
   public boolean query(String name, String
password){ boolean flag = false;
   try{
   String sql = "select * from users where name='"+name+"'
" + "and password='"+password+"' " ;
   Cursor cur=sld.rawQuery(sql, null);
   if(cur.moveToNext()){flag = true;}
   cur.close();
   } catch (Exception e){
    Toast.makeText(this, "There's something wrong in the
database:"+e.toString(), Toast.LENGTH_SHORT).show();}
    return flag; }
    //insert data into the table
   public void insert() {
   try{
   String sql="insert into users values('molly','123')";
    sld.execSQL(sql);
   }catch(Exception e){
   Toast.makeText(this, " There's something wrong about
the database"+e.toString(),
   Toast.LENGTH_SHORT).show();
    }
  }
}
```

B. *Android system remote connect SQL Server database*

Android system can access SQL Server database with intermediate agent WebService. WebService will return the data in XML format. Than obtain the information of database through analyze these XML data.

WebService is a remote procedure call standard based on Simple Object Access Protocol(SOAP). It is a kind of software functions provided at a network address over the Web or the cloud. WebService can be described by XML file. This file is called as WSDL. WebService can integrate different operating system platforms, different programming language and different technology together.

There is not WebService library to call provided by Android SDK directly, We can use the SDK library by a third party called as KSOAP. KSOAP can be downloaded online as follows:

ksoap2-android-assembly-2.4-jar-with-dependencies.jar.

The steps to connect with database system using SQL Server as follows:

Development Environment: Android4.0.4, SQL Server2008, visual studio 2008(WebService code written by C#).

First create a database in SQL Server and corresponding table. Still take the Login program for example.

The table is designed as follow:

TABLE I. LOGIN TABLE

Column Name	Data Type	Length	Is NULL
id	int	4	
name	varchar	50	
password	varchar	50	

a. Example for Login Table.

WebService can created through Visual Studio 2008.

First, We should create a class to manage the database, we named the class DBOperation, we can create, delete, update and read the data in the database. In this article, we only show you how to read the data.

The DBOperation code to realize for managing database be omitted here.

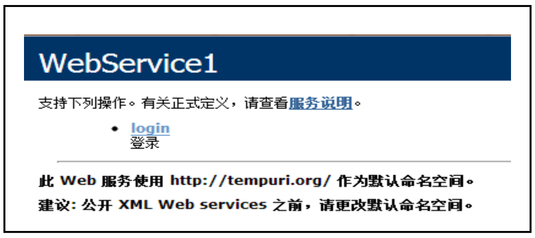
Then, create a WebService in order to read or update the database through the WebService.

Create an instance dbOperation by DBOperation class, which we can call the method of it. Notice the method login, it has two parameters, the name and the password, we can call the method login in the class DBOperation.

The code to create WebService be omitted here.

The running result can be displayed as follows:

Figure 1. The WebService running result.



WebService1

支持下列操作。有关正式定义，请查看服务说明。

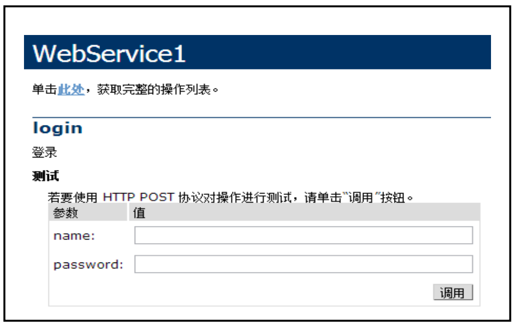
- login
  登录

此 Web 服务使用 http://tempuri.org/ 作为默认命名空间。

建议：公开 XML Web services 之前，请更改默认命名空间。

Remember the URL of the two pages:

Figure 2. The WebService for Login interface.



WebService1

单击此处，获取完整的操作列表。

login

登录

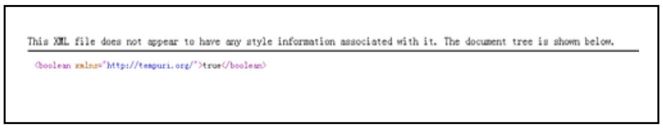
测试

若要使用 HTTP POST 协议对操作进行测试，请单击"调用"按钮。

参数	值
name:	
password:	

Input name and password, click CALL button, the XML information display as follows:

Figure 3. The XML information.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<boolean xmlns="http://tempuri.org/">true</boolean>
```

We can use remote call server when finish WebService created and a series of setups. WebService can be called as follow steps:

- Instantiate SoapObject

Use the method SoapObject (String namespace, String name), we can find the first parameter "namespace" in the Figure 1, http://tempuri.org/, the second parameter name is the method name, clearly it is "login".

- set parameters of the called method

Use the method of SOAP class:

addProperty(java.lang.String name, java.lang.Object value)

Here we need two parameters, the name and the password, so we should use the method addProperty two times.

- set the SOAP request message

Use the class SoapSerializationEnvelope to serialize the request, send the request message to the server.

SoapSerializationEnvelope(int Version), the parameter Version should fit the version in the Web Service.

- create HttpTransportsSE object

HttpTransportsSE(String url), the url is the url of the Figure 1 : http://localhost:49495/WebService1.asmx, therefore we should change the localhost to 10.0.2.2 when we use it.

- call WeService method

void call(String soapAction, SoapEnvelop envelop)

the first parameter soapAction is the namespace string attach the method string.

- obtain WeService method and return result

Use the method getResponse().

In the Android mobile point, create a new project, import ksoap package. The total code written as follows:

```
public class TestActivity extends Activity {
        private Button okButton;
        private Button cancelButton;
        private EditText name;
        private EditText password;
        @Override
            public void onCreate(Bundle
savedInstanceState){
                super.onCreate(savedInstanceState);
            setContentView(R.layout.main);
okButton = (Button) findViewById(R.id.ok_Btn);
cancelButton = (Button) findViewById(R.id.cancel_Btn);
    name = (EditText) findViewById(R.id.name_edit);
password = (EditText) findViewById(R.id.paw_Edit);
    okButton.setOnClickListener(new
Button.OnClickListener() {
    public void onClick(View v) {
    String nameStr = name.getText().toString();
    String pwdStr = password.getText().toString();
    if(nameStr==null||pwdStr==null||nameStr.equals("")||pw
dStr.equals("")) {
        Toast.makeText(TestActivity.this, "name or
password can not be empty",
Toast.LENGTH_SHORT).show();
    } else {boolean bool = login(nameStr, pwdStr);
        if(bool){Toast.makeText(TestActivity.this,
"Successfully!",Toast.LENGTH_SHORT).show();} else {
        Toast.makeText(TestActivity.this,
"Failed!",Toast.LENGTH_SHORT).show();}}}});
cancelButton.setOnClickListener(new
View.OnClickListener(){
    public void onClick(View v) {
      name.setText("");
      password.setText("");} });
    }
    private static final String NAMESPACE =
"http://tempuri.org/";
    private static String URL =
"http://10.0.2.2:49495/WebService1.asmx";
    private static final String METHOD_NAME = "login";
    private static String SOAP_ACTION =
"http://tempuri.org/login";
```

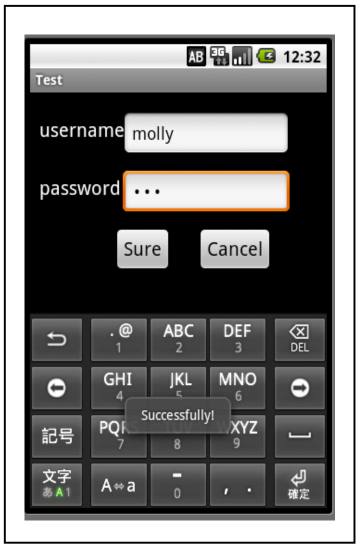
```
    private Object detail;
    public Boolean login(String name, String password){
      boolean flag = false;
      try{
         SoapObject rpc = new SoapObject(NAMESPACE,METHOD_NAME);
         rpc.addProperty("name", name);
         rpc.addProperty("password", password);
         SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(SoapEnvelope.VER10);
         envelope.bodyOut = rpc;
         envelope.dotNet = true;
         envelope.setOutputSoapObject(rpc);
         HttpTransportSE ht = new HttpTransportSE(URL);
         ht.debug = true;
         ht.call(SOAP_ACTION, envelope);
         detail = (Object) envelope.getResponse();
         if(detail.toString().equals("true")){
           flag = true;} else {flag = false;}
    } catch(Exception e){
       e.printStackTrace();}
    return flag;
    }
}
```

C. *An example for login interface using Android database*

This is an example for login interface. The figure shows login application simulating mobile under Android system environment.

The application has connected to database and accessed the data from database. So the mobile screen give the information for accessed the data successfully as follows:

Figure 4. Login application running results



IV. Conclusion

This paper states the two technologies to access database: local and remote access to realize in the mobile with Android system and comparative analyze them. Come to the conclusion, the two models must use the two different technologies, to correspond different application scenario. Take the most popular SQLite in Android system and the most widely used SQL Server in Internet as the examples. Using SQLite for local model and SQL Server for remote model is easy and reasonable model. The strengths of two technologies are different but complementary. To take different processing technologies according to different actual situations.

These examples in this paper, background provided by mobile phone property management system of the city Housing Management Bureau. This database application technologies and realize methods can be extended to mobile phone business management application project in a general way. The succeeding works to develop database application of this business management system are more and more improve and perfect at present.

References

[1] Margaret Butler, “Android: Changing the Mobile Landscape, IEEE pervasive computing”, Vol.10 January 2011, pp.4–7/doi:10.1109/ MPRV.2011.1

[2] Nicholas Palmer, Emilian Miron, Roelof Kemp, Thilo Kielmann, Henri Bal, “Towards Collaborative Editing Of Structured Data On Mobile Devices”, IEEE International Conference on Mobile Data Management (MDM 2011), pp.194-199

[3] Chun-mao, Liu Yun-gang, Zhang Hong-ying, Wang, “Linux-based remote monitoring and control of CNC machine tool of SQLite,”International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT 2011).Vol.1,2011,pp.9-12

[4] Ajila, S.A. Al-Asaad, A., “Mobile databases - Synchronization &amp; conflict resolution strategies using SQL server”,IEEE International Conference on Information Reuse and Integration (IRI 2011),pp.487-489//doi:10.1109/IRI.2011.6009598

[5] Weiqi Song, Tao Tao, Tiegang Gao , “Performance Optimization for Flash Memory Database in Mobile Embedded System”, Education Technology and Computer Science, March 2010, pp. 35-39

[6] Chuangwei Zhang, Xu Yin, “Design and implementation of single-service multi-function Webservice”, Computer Science and Service System (CSSS 2011), 2011,PP.3912 – 3915/doi:10.1109/ CSSS.2011.5974771

[7] Damianos Gavalas, Daphne Economou, “Development Platforms for Mobile Applications: Status and Trends”,IEEE Software,Vol.28 January 2011, pp.77-86//doi:10.1109/MS.2010.155

[8] Weider D. Yu, Hongbin Yuan, “An Approach to Explore Mobile Software Engineering Advances in Cloud Computing Environment”, Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual,July 2011,pp. 292 - 297// doi:10.1109/COMPSACW.2011.98

[9] Huang, Chung-Lin; Huang, Chung-Chi; Huang, Cong-Hui; Liu, Cheng-Wei, “Development of Cloud Computing Based Intelligent Integrated Manufacturing System”, Advanced Science Letters, Vol.9, April 2012 , pp. 16-23//doi:10.1166/ASL.2012.2544

[10] Li Ye-bai, Zhang Bin, Wang Hai-bin, “Study and Design on Data Management Model of SQL Server CE For Mobile Application”, 2010 International Conference on e-Education, e-Business, e-Management and e-Learning (IC4E 2010)//doi:10.1109/IC4E.2010.73