**Android 网络编程**

1. 使用socket通信

/\*\*

\* 服务端应该采用多线程处理客户端Socket请求，

\* 一个子线程维护一个与客户端Socket的链接，

\* 此处为了方便，未使用多线程

\*/

public void myServerSocket() throws IOException {

//指定30000端口创建ServerSocket

ServerSocket serverSocket = new ServerSocket(30000);

//循环接收客户端请求

while (true){

//接收到客户端Socket连接请求后，产生一个与之对应的Socket

Socket socket = serverSocket.accept();

OutputStream outputStream = socket.getOutputStream();

outputStream.write("This is a ServerSocket!".getBytes("UTF-8"));

outputStream.close();

socket.close();

}

}

/\*\*

\* 由于Socket是阻塞式的，

\* 客户端应该创建子线程处理网络请求，

\* 此处为了方便，未使用子线程

\*/

public void myClientSocket() throws IOException {

//创建Socket到指定主机（192.168.2.23）指定端口(30000)

Socket socket = new Socket("192.168.2.23", 30000);

BufferedReader bufferedReader = new BufferedReader(

new InputStreamReader(socket.getInputStream()));

String serverData = bufferedReader.readLine();

bufferedReader.close();

socket.close();

}

1. 使用URL通信
2. URL

public void myURL() throws IOException {

URL url = new URL("http://www.oschina.net");

InputStream inputStream = url.openStream();

//省略inputStream相关处理过程

inputStream.close();

}

1. URLConnection
2. GET请求

/\*\*

\* GET请求，直接调用URLConnection.connect()方法

\* 即可获取网络数据

\*/

public void myURLConnection() throws IOException {

URL url = new URL("http://www.oschina.net");

URLConnection urlConnection = url.openConnection();

urlConnection.connect();

InputStream inputStream = urlConnection.getInputStream();

//省略inputStream相关处理过程

inputStream.close();

}

1. POST请求

/\*\*

\* POST请求，必须调用URLConnection得setDoOutput()和setDoInput()方法,

\* 且setDoOutput()要在setDoInput()之前

\*/

public void myURLConnection() throws IOException {

URL url = new URL("http://www.oschina.net");

URLConnection urlConnection = url.openConnection();

urlConnection.setDoOutput(true);

urlConnection.setDoInput(true);

OutputStream outputStream = urlConnection.getOutputStream();

……

InputStream inputStream = urlConnection.getInputStream();

//省略inputStream相关处理过程

inputStream.close();

}

1. HttpURLConnection（与URLConnection用法相同）

public void myHttpURLConnection() throws IOException {

URL url = new URL("http://www.oschina.net");

HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();

//设置请求方式GET或POST

urlConnection.setRequestMethod("GET");

InputStream inputStream = urlConnection.getInputStream();

//省略inputStream相关处理过程

inputStream.close();

}

1. HttpClient

public void myHttpClient() throws IOException {

HttpClient httpClient = new DefaultHttpClient();

HttpGet httpGet = new HttpGet("http://www.oschina.net");

HttpResponse httpResponse = httpClient.execute(httpGet);

HttpEntity httpEntity = httpResponse.getEntity();

//省略对HttpEntity相关处理过程

}