



## 《使用 HTTP 协议访问网络》习题

### 一、填空题

1. Android 系统提供了多种网络通信方式，包括\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
2. 当客户端与服务器端建立连接后，想服务器端发送的请求，被称\_\_\_\_\_。
3. Android 客户端访问网络发送 HTTP 请求的方式一般有两种，和\_\_\_\_\_。
4. 与服务器交互过程中，最常用的两种数据提交方式是和\_\_\_\_\_。
5. 为了根据下载进度实时更新 UI 界面，需要用到 Handle 消息机制来实现\_\_\_\_\_。

### 二、问答题

1. 什么是 Http 通信？什么是 Socket 通信？两者的区别又是什么？
2. 请简述使用 HTTP 协议访问网络的几种方法。
3. 请简述使用 HttpURLConnection 的步骤。
4. 请简述使用 OkHttp 的步骤。



## 《使用 HTTP 协议访问网络》习题答案

### 一、填空题

1. Android 系统提供了多种网络通信方式，包括 Socket 通信、HTTP 通信、URL 通信和 WebView。
2. 当客户端与服务器端建立连接后，想服务器端发送的请求，被称 HTTP 请求。
3. Android 客户端访问网络发送 HTTP 请求的方式一般有两种，HttpURLConnection和 HttpClient。
4. 与服务器交互过程中，最常用的两种数据提交方式是 GET和 POST。
5. 为了根据下载进度实时更新 UI 界面，需要用到 Handle 消息机制来实现 线程间通信。

### 二、问答题

#### 1. 答案

Android 与服务器的通信方式主要有两种，一是 Http 通信，一是 Socket 通信。

Http (Hypertext Transfer Protocol)，是 Web 互连网的基础，它是建立在 TCP 协议之上的一种客户端/服务器端协议，于 1990 年被提出，经过多年的使用与发展，得到不断的完善和扩展。Http 连接的工作过程是客户端发送的 Http 请求，等待服务器接收请求回送响应，客户端解析、处理返回的信息。在请求结束后，服务器处理完客户端的请求，就会主动释放连接。从建立连接到关闭连接的过程称为“一次连接”。由于 Http 协议简单，使得 Http 服务器的程序规模小，因而通信速度很快。

Socket 是一种抽象层，通信两端通过它来发送和接收数据，使用 Socket 可以将应用程序添加到网络中，与处于同一网络中的其他应用程序进行通信。Socket 有两种传输模式：面向连接和无连接。面向连接模式使用 TCP 协议，在通信两端建立通信链路，依次发送和接收数据。无连接模式使用 UDP 协议，将数据进行打包发送，与面向连接模式不同的是，接收端接收数据包顺序与发送端发送的顺序是不一样的，而且数据安全性欠佳。



Http 通信和 Socket 通信两者的最大差异在于, Http 连接使用的是“请求—响应方式”,即在请求时建立连接通道,当客户端向服务器发送请求后,服务器端才能向客户端返回数据。而 Socket 通信则是在双方建立起连接后就可以直接进行数据的传输,在连接时可实现信息的主动推送,而不需要每次由客户端向服务器发送请求。

## 2. 答案

用 HttpURLConnection 及 OkHttpClient 访问网络。

## 3. 答案

第一步、获取 HttpURLConnection 的实例:

```
URL url = new URL("http://www.baidu.com");  
  
HttpURLConnection conn  
= (HttpURLConnection) url.openConnection();
```

第二步、设置 HTTP 请求所使用的方法: GET 和 POST

GET 表示从服务器获取数据, POST 表示提交数据给服务器。

```
conn.setRequestMethod("GET");
```

第三步、进行自由定制:

```
conn.setConnectTimeout(5000); //连接超时
```

```
conn.setReadTimeout(5000); //读取超时
```

第四步、调用 `getInputStream()`; 方法获取到服务器返回的输入流

```
InputStream is = conn.getInputStream();
```

对输入流进行读取。

第五步、调用 `disconnect()` 方法将这个 HTTP 连接关闭

```
conn.disconnect();
```

## 4. 答案

第一步、在项目中添加 OkHttpClient 库的依赖。

编辑 `app/build.gradle` 文件

```
dependencies {
```

```
...
```



```
compile 'com.android.support:appcompat-v7:25.0.1'  
testCompile 'junit:junit:4.12'  
compile 'com.squareup.okhttp3:okhttp:3.4.1'  
}
```

第二步、创建一个 OkHttpClient 实例。

```
OkHttpClient client=new OkHttpClient();
```

第三步、创建 Request 对象。

```
Request request=new  
Request.Builder().url("http://www.baidu.com").build();
```

通过 url() 方法来设置目标的网络地址，Request 是 OkHttpClient 中访问的请求，Builder 是辅助类。

第四步、创建 Call 对象。

```
Response response=client.newCall(request).execute();
```

通过 OkHttpClient 的 newCall() 方法创建 Call 对象，execute() 方法来发送请求并获取服务器返回的数据

第五步、得到返回的具体内容。

```
String responseData=response.body().string();
```

response.body() 返回 ResponseBody 类可以方便的获取 string