



上海建桥学院

SHANGHAI JIAN QIAO UNIVERSITY



网络编程

主讲：万永权

学习目标/Target



了解HTTP协议通信简介，能够说出什么是HTTP协议

掌握HttpURLConnection的使用方法，能够使用
HttpURLConnection访问网络

掌握WebView控件的使用方式，能够使用WebView控件加载不同的网页

掌握JSON数据的解析，能够通过不同的方式解析JSON数据

熟悉Handler消息机制的概述，能够归纳Handler消息机制的原理

章节概述 / Summary



在移动互联网时代，手机联网实现信息互通是最基本的功能体验。例如，在上下班的途中或旅行时，只要有时间人们就会拿出手机上网，通过手机接收新资讯、搜索网络资源。Android作为智能手机市场中主流的操作系统，它的强大离不开其对网络功能的支持。Android系统提供了多种实现网络通信的方式。接下来，我们从**最基础的HTTP协议开始**，到Android中原生的HttpURLConnection、WebView控件的使用以及网络数据的解析进行详细讲解。





目录/Contents



01

通过HTTP访问网络

02

使用WebView进行网络开发

03

JSON数据解析

04

Handler消息机制



9.1

通过HTTP访问网络



9.1 通过HTTP访问网络



先定一个小
目标！

- 了解HTTP协议通信简介，能够说出什么是HTTP协议
- 掌握HttpURLConnection的使用方法，能够使用HttpURLConnection访问网络



9.1.1 HTTP协议通信简介

HTTP(Hyper Text Transfer Protocol)即超文本传输协议，它规定了浏览器和服务器之间相互通信的规则。

HTTP协议是一种请求/响应式的协议。

- 当客户端与服务器端建立连接后，向服务器端发送的请求，称作HTTP请求。
- 当服务器端接收到请求后会做出响应，称为HTTP响应。



9.1.1 HTTP协议通信简介

HTTP(Hyper Text Transfer Protocol)即超文本传输协议，它规定了浏览器和服务器之间相互通信的规则。

HTTP协议是一种请求/响应式的协议。

- 当客户端与服务器端建立连接后，向服务器端发送的请求，称作HTTP请求。
- 当服务器端接收到请求后会做出响应，称为HTTP响应。

9.1.1 HTTP协议通信简介



使用手机客户端访问百度时，会发送一个HTTP请求，当服务器端接收到请求后，会做出响应并将百度页面（数据）返回给客户端浏览器，这个请求响应的过程就是HTTP通信的过程。



>>> 9.1.2 使用HttpURLConnection访问网络

HttpURLConnection的用法

在URL的构造方法中传入要访问资源的路径

```
URL url = new URL("http://www.itcast.cn");
```

创建HttpURLConnection对象

```
HttpURLConnection conn = (HttpURLConnection)url.openConnection();
```

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

设置请求方式
设置超时时间
获取服务器返回的输入流
关闭HTTP连接

```
conn.setConnectTimeout(5000);
```

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

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

注意：使用HttpURLConnection对象访问网络时，需要设置超时时间，防止连接被阻塞时无响应，影响用户体验。



9.1.2 使用HttpURLConnection访问网络

GET与POST请求

(1) GET方式

GET方式是以实体的方式得到由请求URL所指向的资源信息，它向服务器提交的参数跟在请求URL后面。使用GET方式访问网络URL的长度一般要小于1KB。

(2) POST方式

POST方式向服务器发出请求时需要在请求后附加实体。它向服务器提交的参数在请求后的实体中，POST方式对URL的长度是没有限制的。

采用POST方式提交数据时，用户在浏览器中看不到向服务器提交的请求参数，因此POST方式要比GET方式相对安全。



>>> 9.1.2 使用HttpURLConnection访问网络

GET方式提交数据

```
String path = "http://192.168.1.100:8080/web/LoginServlet?username="  
    + URLEncoder.encode("zhangsan")  
    + "&password=" + URLEncoder.encode("123");
```

将用户名和密码拼在指定资源路径后面，并对用户名和密码进行编码

```
URL url = new URL(path);
```

```
HttpURLConnection conn = (HttpURLConnection)url.openConnection();
```

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

```
conn.setConnectTimeout(5000);
```

```
int responseCode = conn.getResponseCode();
```

获取到状态码

```
if(responseCode == 200){
```

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

状态码为200，表示访问成功
获取返回内容的输入流

```
}
```



9.1.2 使用HttpURLConnection访问网络

POST方式提交数据

```
URL url = new URL("http://192.168.1.100:8080/web/LoginServlet");
HttpURLConnection conn = (HttpURLConnection) url.openConnection();
conn.setConnectTimeout(5000);
conn.setRequestMethod("POST");
String data = "username=" + URLEncoder.encode("zhangsan")
    + "&password=" + URLEncoder.encode("123");
conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
conn.setRequestProperty("Content-Length", data.length() + "");
conn.setDoOutput(true);
OutputStream os = conn.getOutputStream();
os.write(data.getBytes());
int code = conn.getResponseCode();
if (code == 200) {
    InputStream is = conn.getInputStream();
}
```

准备数据并发送

设置请求头数据提交方式以及提交数据的长度，这里是以form表单的方式提交

以流的形式将数据写到服务器上

以流的形式将数据写到服务器上

»»» 9.1.2 使用HttpURLConnection访问网络



注意

在实际开发中，手机端与服务器端进行交互的过程中避免不了要提交中文到服务器，这时就会出现[中文乱码的情况](#)。无论是GET方式还是POST方式提交参数时都要给参数进行编码，编码方式必须与服务器解码方式一致。同样在获取服务器返回的中文字符时，也需要用指定格式进行解码。



9.2

使用WebView进行网络开发



9.2 使用WebView进行网络开发



先定一个小
目标！

- 掌握**WebView**的使用方式，能够使用**WebView**浏览不同网页、执行HTML代码和支持JavaScript



9.2.1 使用WebView浏览网页

在Android程序中，**WebView控件可以在XML布局文件中使用<WebView>标签来添加，也可以在Java文件中通过new关键字来创建。**

通常会采用**在XML布局文件中添加<WebView>标签的形式**，具体代码如下。

```
<WebView
```

```
    android:id="@+id/webView" → WebView控件的id
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent" />
```



9.2.1 使用WebView浏览网页

<WebView>控件的常用方法如下表所示。

属性名称	功能描述
loadUrl(String url)	用于加载指定URL对应的网页
loadData(String data, String mimeType, String encoding)	用于将指定的字符串数据加载到浏览器中
loadDataWithBaseUrl(String baseUrl, String data, String mimeType, String encoding, String historyUrl)	基于URL加载指定的数据
capturePicture()	用于创建当前屏幕的快照
goBack()	用于执行后退操作，相当于浏览器上后退按钮的功能
goForward()	用于执行前进操作，相当于浏览器上前进按钮的功能
stopLoading()	用于停止加载当前页面
reload()	用于刷新当前页面



9.2.1 使用WebView浏览网页

接下来通过一个案例来演示如何使用WebView控件加载网页，本案例的界面效果如下图所示。

① 创建名为WebView的程序

1 创建程序： ② 指定包名为com.<yourname>.webview

2 放置界面控件： 放置1个WebView控件

① 通过findViewById()方法获取WebView控件

3 编写界面交互代码： ② 调用loadUrl()方法来加载指定的网页





9.2.1 使用WebView浏览网页

在MainActivity中实现WebView控件浏览网页的功能，通过WebView控件的loadUrl()方法来加载指定的网页，主要代码如下。

```
WebView webview=(WebView)findViewById(R.id.webView);  
webview.loadUrl("http://www.itheima.com/");
```

通过加载网页地址加载网页

需要在清单文件 (AndroidManifest.xml) 的< manifest>标签中添加允许访问网络资源的权限。

```
<uses-permission android:name="android.permission.INTERNET"/>
```



9.2.1 使用WebView浏览网页



注意

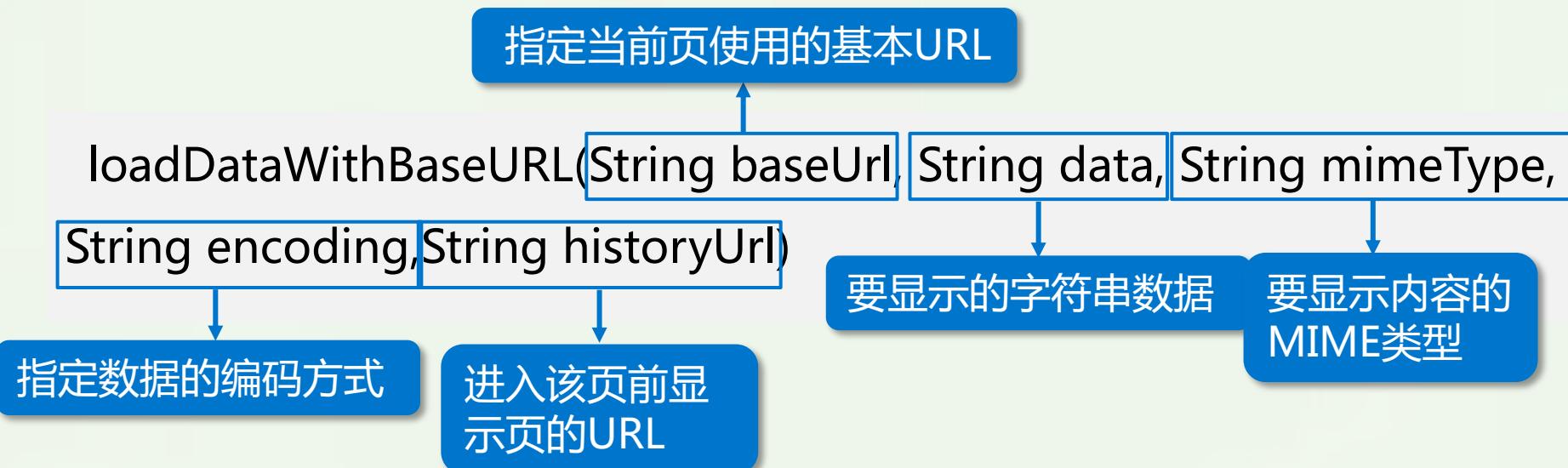
如果想让上述WebView控件具备放大和缩小网页的功能，则需要对该控件进行如下设置：

```
//设置WebView控件支持使用屏幕控件或手势进行缩放  
webview.getSettings().setSupportZoom(true);  
  
//设置WebView控件使用其内置的变焦机制，该机制集合屏幕缩放控件使用  
webview.getSettings().setBuiltInZoomControls(true);
```



9.2.2 使用WebView执行HTML代码

WebView类提供了**loadData()**和**loadDataWithBaseUrl()**方法加载HTML代码。当使用**loadData()**方法来加载带中文的HTML内容时会产生乱码，但是使用**loadDataWithBaseUrl()**方法就不会出现这种情况。**loadDataWithBaseUrl()**方法的定义方式如下：





9.2.2 使用WebView执行HTML代码

接下来通过一个案例来演示如何使用WebView控件加载HTML代码，本案例的界面效果如下图所示。

1

创建程序： ① 创建名为WebViewHtml的程序
② 指定包名为com.<yourname>.webviewhtml

2

实现加载HTML的功能： 在MainActivity中实现WebView控件
加载HTML代码的功能

WebViewHtml

请选择您要学习的课程：

- 新媒体课程
- 大数据课程
- 人工智能课程



9.2.3 设置WebView支持JavaScript

为了解决WebView控件在默认情况下不支持JavaScript代码的问题，我们需要通过setJavaScriptEnabled()方法来设置WebView控件，使其可以支持JavaScript代码。

```
WebSettings settings= webview.getSettings(); // 获取WebSettings对象
```

```
settings.setJavaScriptEnabled(true); → 设置JavaScript可用
```

```
webview.setWebChromeClient(new WebChromeClient());
```

使WebView控件显示带有
JavaScript代码的提示框



9.2.2 使用WebView执行HTML代码

接下来通过一个案例来演示如何使用WebView控件支持一个带有JavaScript代码的网页，本案例的界面效果如下图所示。

① 创建名为WebViewJS的程序

1 创建程序： ② 指定包名为com.<yourname>.webviewjs

将JavaScript文件导入到程序中创建

2 导入JavaScript文件： assets文件夹中

① 放置1个Button控件

3 放置界面控件： ② 放置1个WebView控件

① 导入背景图片

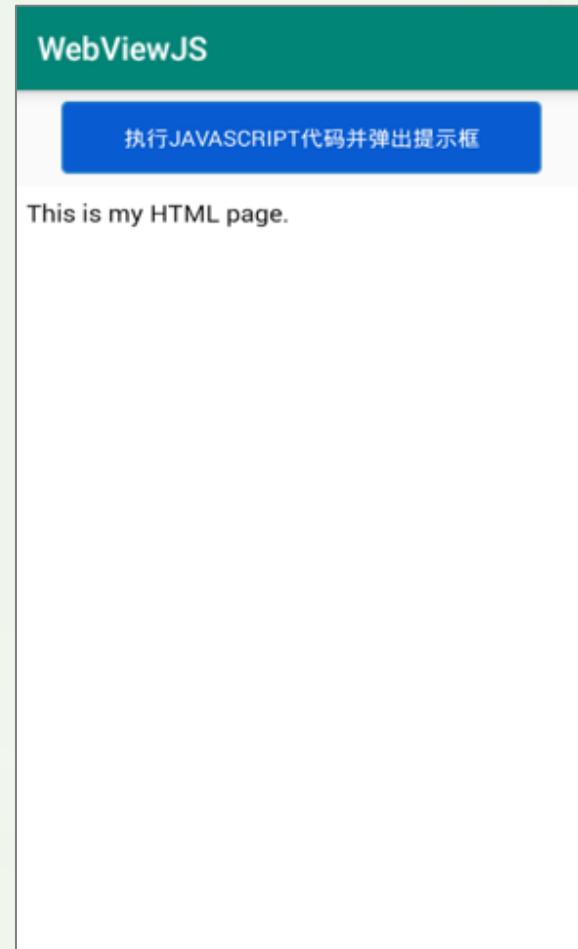
4 创建背景选择器： ② 创建背景选择器btn_dialog_selector.xml

在MainActivity中实现WebView控

5 实现加载JavaScript代码的功能： ② 支持JavaScript代码的功能

① 运行程序

6 运行程序： ② 点击“执行JAVASCRIPT代码并弹出提示框”按钮





9.2.3 设置WebView支持JavaScript





9.3

JSON数据解析



9.3 JSON数据解析



先定一个小
目标！

掌握JSON数据的解析，能够通过不同的方式解析
JSON数据

9.3.1 JSON数据



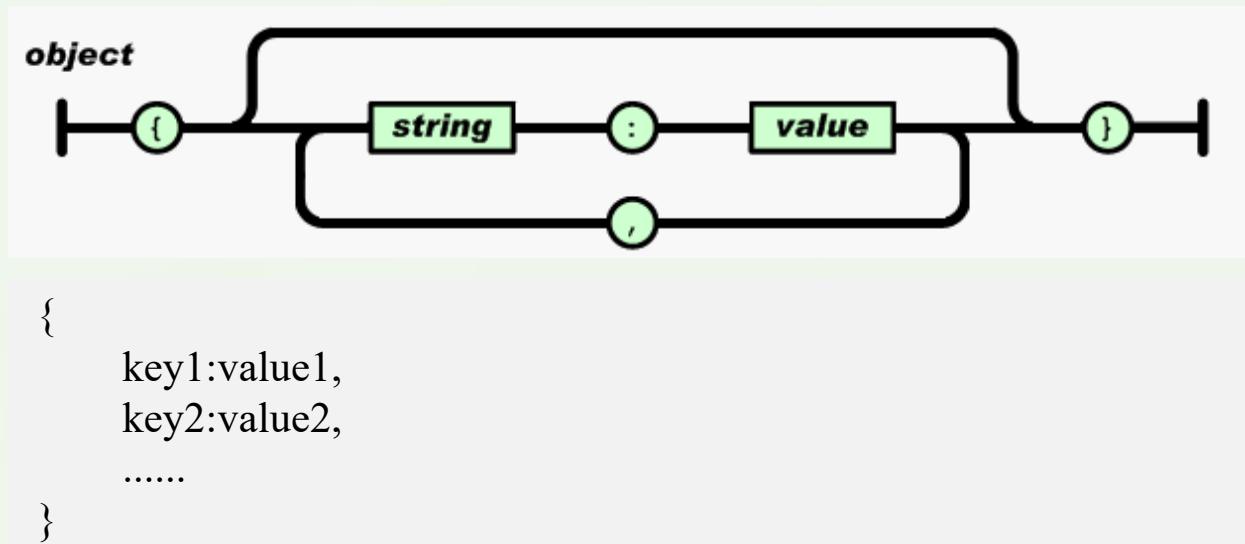
JSON数据的特点

- (1) JSON即JavaScript Object Notation（对象表示法），是一种轻量级的数据交换格式。
- (2) JSON是基于纯文本的数据格式，它可以传输String、Number、Boolean类型的数据，也可以传输数组或者Object对象。
- (3) JSON文件的扩展名为.json。
- (4) JSON分为JSON对象和JSON数组两种数据结构。

9.3.1 JSON数据

对象结构的JSON数据

以“{”开始，以“}”结束。中间部分由0个或多个以“,”分隔的key:value对构成，注意关键字和值之间以“：“分隔。

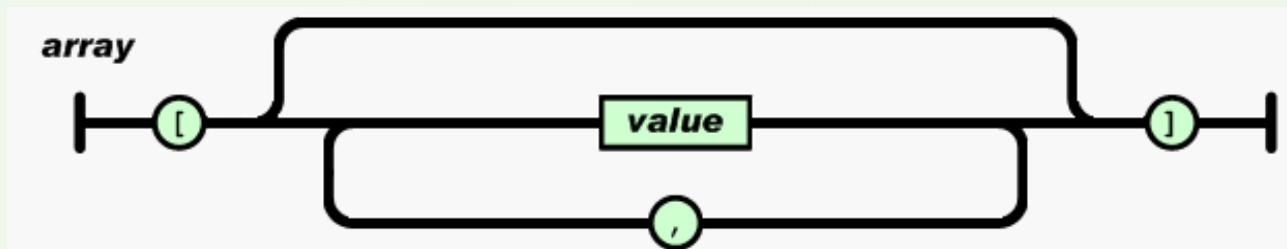


关键字key必须为String类型，值value可以是String、Number、Object、Array等数据类型。

9.3.1 JSON数据

数组结构的JSON数据

以“[”开始，以“]”结束。中间部分由0个或多个以“，”分隔的值的列表组成。



```
[  
    value1,  
    value2,  
    .....  
]
```

值value可以是String、Number、Boolean、null等数据类型。



9.3.1 JSON数据

对象包含对象

```
{  
    "name": "zhangsan",  
    "address":{  
        "city":"Beijing",  
        "street":"Xisanqi",  
        "postcode":100096  
    }  
}
```

对象包含数组

```
{  
    "name": "zhangsan",  
    "hobby":["篮球","羽毛球","游泳"]  
}
```

注意：使用JSON存储单个数据（如“abc”），一定使用数组结构，
因为对象结构必须是由“key:value”的形式构成。



9.3.2 JSON解析

两种解析方式

org.json

Android SDK中为开发者提供的，通过使用[JSONObject](#)和[JSONArray](#)两个类完成对JSON数据的解析。

Gson

由Google公司提供的，在使用[Gson库](#)之前，首先需要将gson.jar添加到项目中，然后才能调用其提供的方法。



9.3.2 JSON解析

解析JSON对象

例如，要解析的JSON数据如下：

```
{ "name": "zhangsan", "age": 27, "married":true }      //json1 一个json对象  
[{"name": "lisi", "age": 25}, {"name": "Jason", "age": 20}] //json2 一个json数组
```

使用[JSONObject](#)解析JSON对象：

```
JSONObject jsonObj = new JSONObject(json1);  
String name = jsonObj.optString("name");  
int age = jsonObj.optInt("age");  
boolean married = jsonObj.optBoolean("married");
```

`optXXX()`方法在解析数据时比`getXXX()`方法更安全，如果对应字段不存在，`optXXX()`方法会返回空值或者0，而`getXXX()`方法会抛出异常。



9.3.2 JSON解析

解析JSON数组

使用JSONArray解析JSON数组：

```
JSONArray jsonArray = new JSONArray(json2);
for(int i = 0; i < jsonArray.length(); i++) {
    JSONObject jsonObj = jsonArray.getJSONObject(i);
    String name = jsonObj.optString("name");
    int age = jsonObj.optInt("age");
}
```

数组的解析方法和对象类似，只是将key值替换为数组中的下标。



9.3.2 JSON解析

Gson库解析JSON数据

例如，要解析的JSON数据如下（与org.json解析数据相同）：

```
{ "name": "zhangsan", "age": 27, "married":true }      //json1 一个json对象  
[{"name": "lisi", "age": 25}, {"name": "Jason", "age": 20}] //json2 一个json数组
```

使用[Gson库](#)前，首先需要将[gson.jar](#)添加到项目中，并且创建JSON数据对应的实体类Person1与Person2，需要注意的是，实体类中的成员名称要与JSON数据中的key值一致。



9.3.2 JSON解析

Gson库解析JSON数据

使用Gson解析JSON对象

将JSON数据转换成对象

```
Gson gson = new Gson();  
Person person1 = gson.fromJson(json1, Person1.class);
```

使用Gson解析JSON数组

TypeToken是Google提供的一个解析JSON数据的类

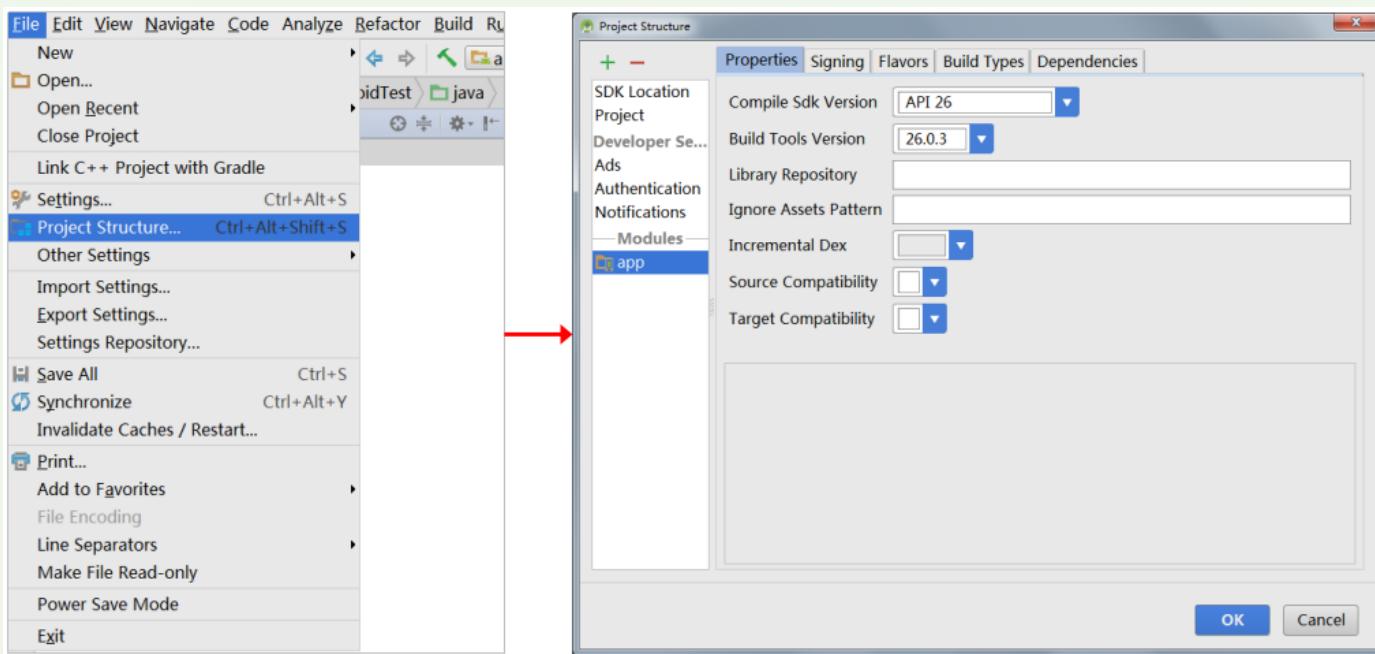
```
Gson gson = new Gson();  
Type listType = new TypeToken<List<Person2>>(){}.getType();  
List<Person2> person2 = gson.fromJson(json2, listType);
```



Android Studio添加库文件

在Android程序中添加库文件进行讲解，具体操作步骤如下：

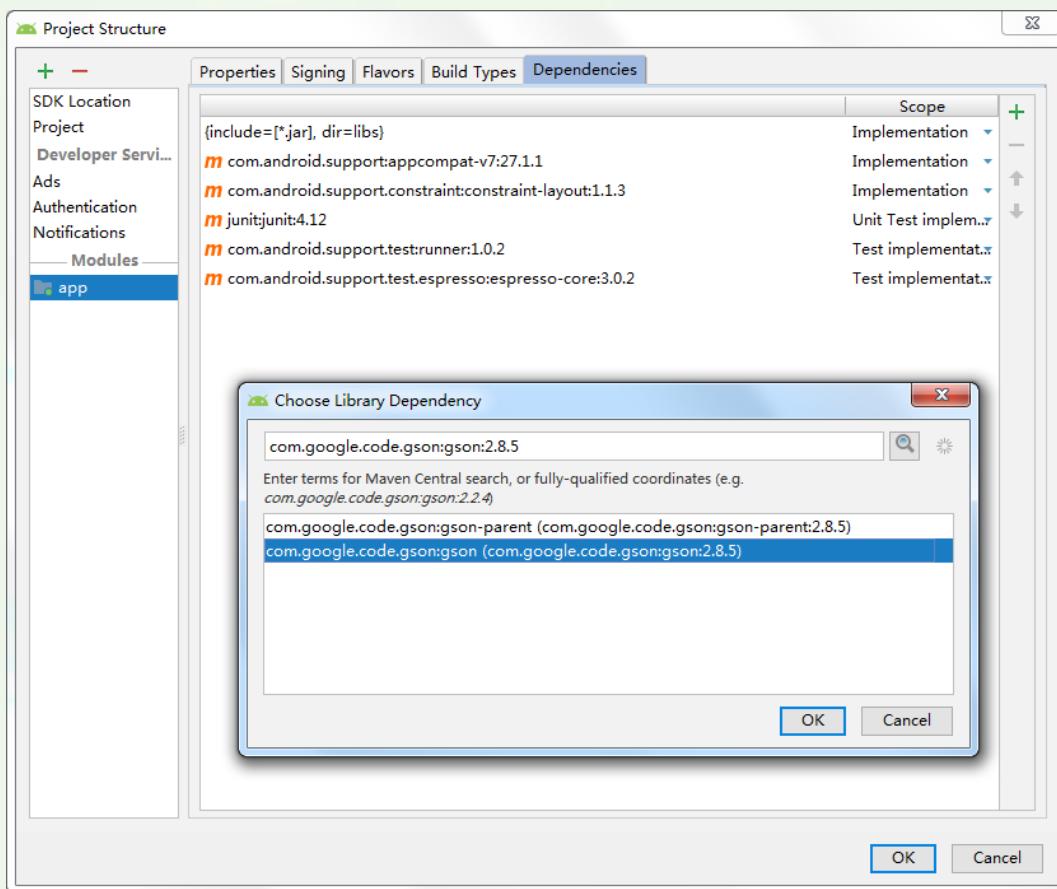
1. 在Android Studio中，选择【File】→【Project Structure...】选项，此时会弹出一个Project Structure窗口，如下图所示。





Android Studio添加库文件

2. 选中Project Structure窗口中的【Dependencies】选项卡，接着单击该窗口右上角的“+”，选择Library dependency选项，此时会弹出一个Choose Library Dependency窗口，在该窗口中找到Gson库com.google.code.gson:gson:2.8.5并选中，如下图所示。



9.3.3 实战演练—仿拼多多砍价界面

本节我们将通过仿拼多多砍价界面的案例来演示如何解析JSON数据并将数据显示到界面上。本案例的界面效果如下图所示。

1

搭建界面布局：

- ① 创建名为Pinduoduo的程序
- ② 导入界面图片与添加recyclerview-v7库
- ③ 放置界面控件
- ④ 搭建商品列表的条目布局
- ⑤ 修改默认标题栏的名称

2

实现界面功能：

- ① 添加okhttp库、json库和glide-3.7.0.jar库
- ② 准备商品信息数据
- ③ 封装商品信息的实体类
- ④ 编写商品列表的适配器
- ⑤ 实现显示商品列表数据的功能

3

运行程序：将程序运行到模拟器





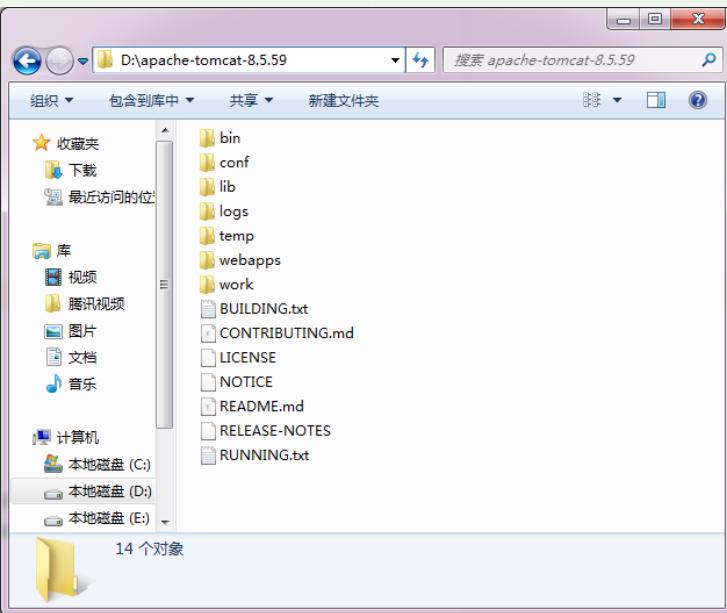
安装配置Tomcat服务器



Tomcat运行稳定、可靠、效率高，不仅可以和目前大部分主流的Web服务器（如Apache、IIS服务器）一起工作，还可以作为独立的Web服务器软件。

1. 下载Tomcat

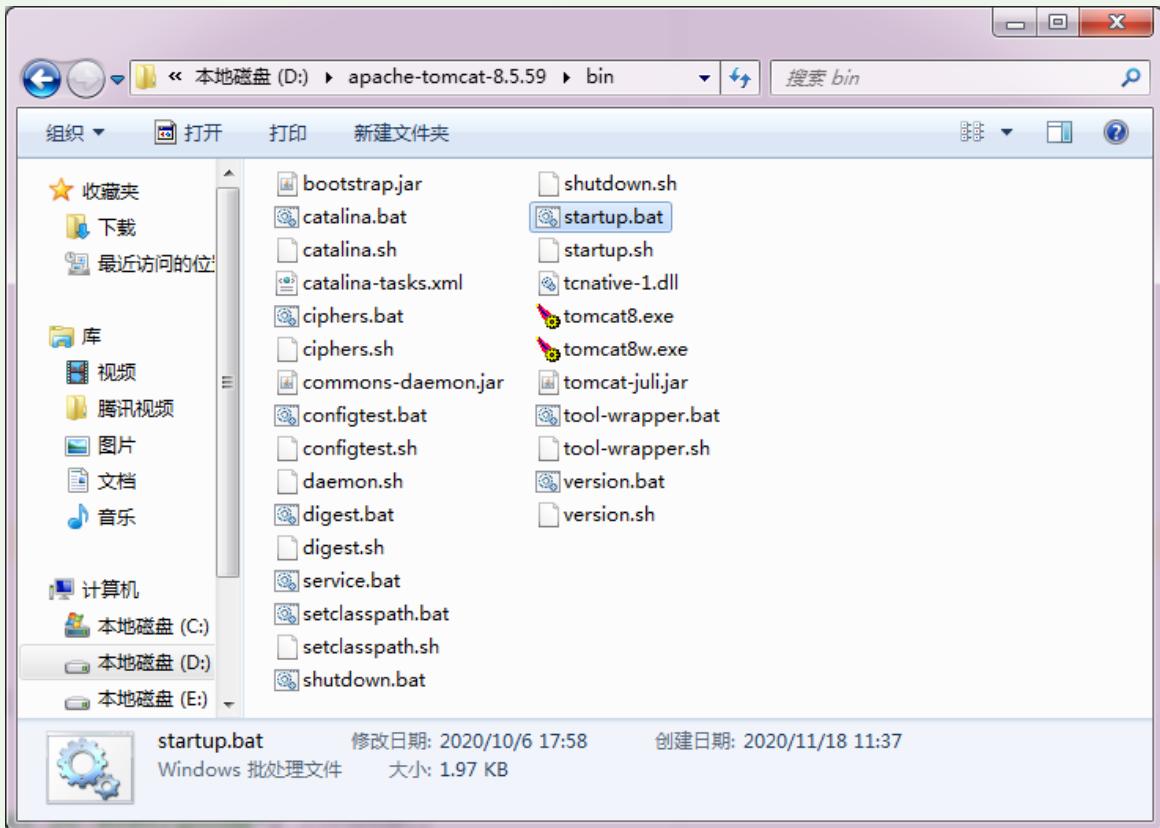
在Tomcat官网上下载apache-tomcat-8.5.59-windows-x64.zip文件，解压该文件可以看到Tomcat的目录结构，如下图所示。



安装配置Tomcat服务器

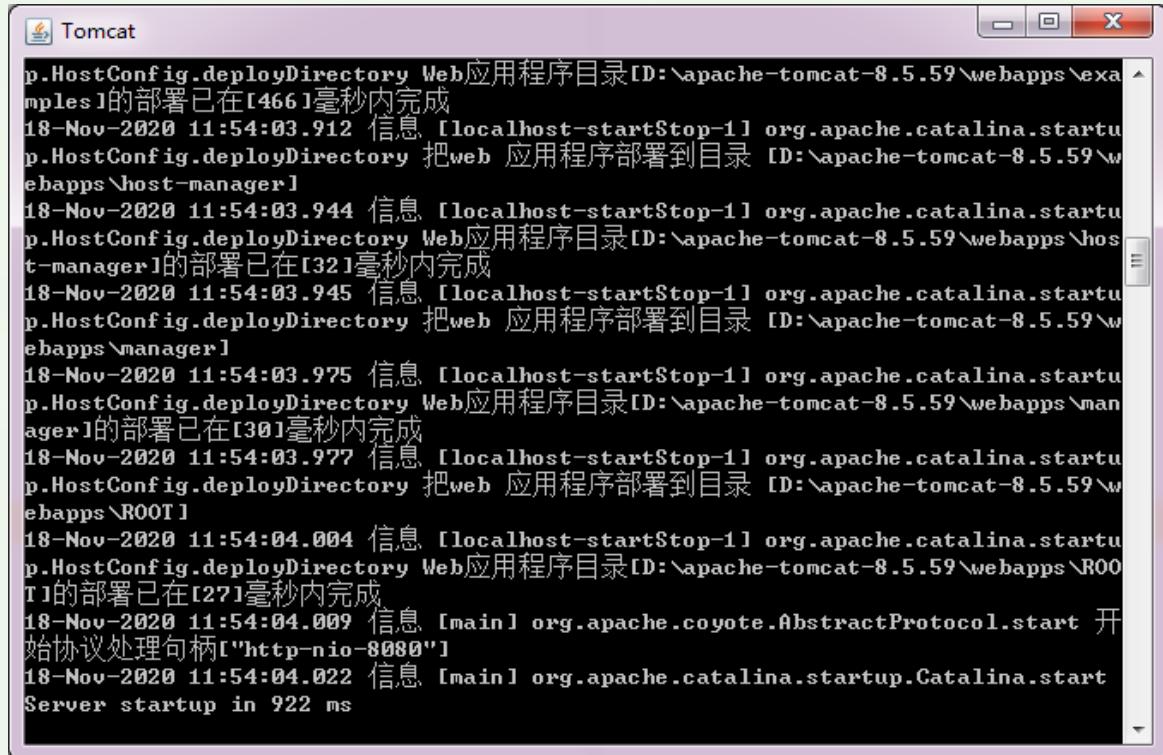
2. 启动Tomcat

在Tomcat安装目录的bin目录下，存放了许多脚本文件，其中[startup.bat](#)就是启动Tomcat的脚本文件，如下图所示。

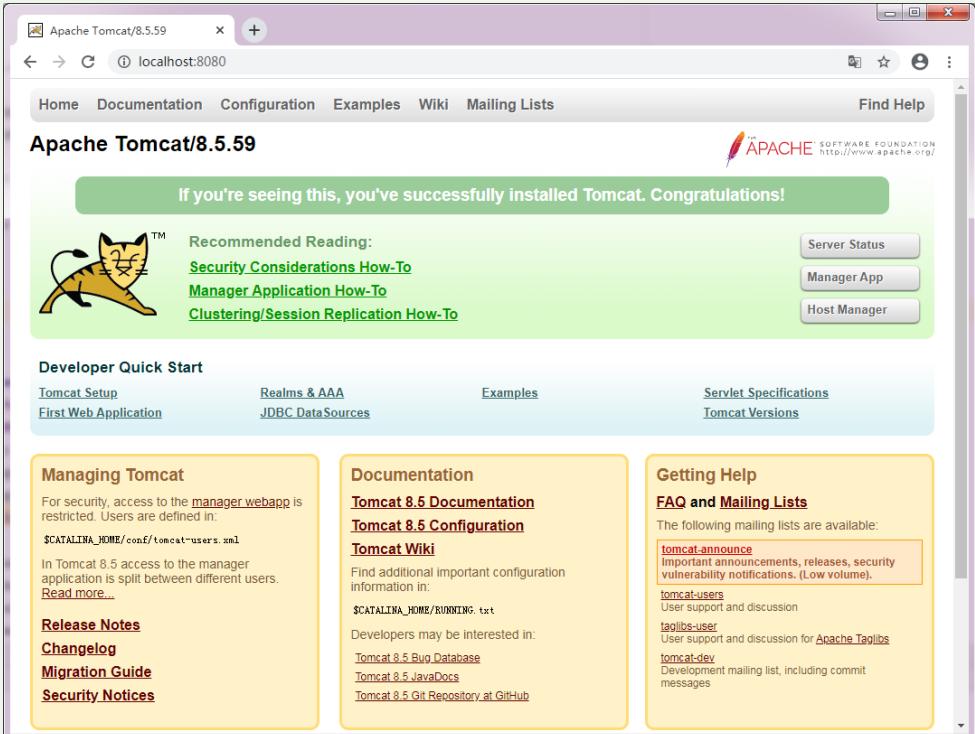


安装配置Tomcat服务器

双击startup.bat文件，便会启动Tomcat服务器，Tomcat启动信息窗口如下图所示。



Tomcat服务器启动后，在浏览器的地址栏中输入`http://localhost:8080`访问Tomcat服务器，如果浏览器中的显示Tomcat页面如下图所示，则说明Tomcat 服务器安装部署成功了。



3. 关闭Tomcat

在Tomcat根目录下的bin文件夹中，运行`shutdown.bat`脚本文件即可关闭Tomcat或者直接关闭Tomcat启动信息窗口。

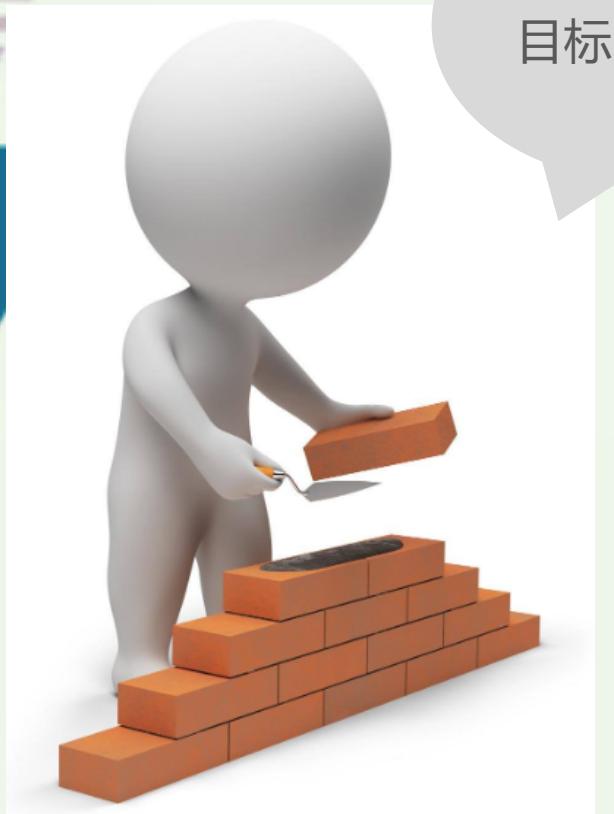


9.4

Handler消息机制



9.4 Handler消息机制



先定一个小
目标！

熟悉Handler消息机制的概述，能够归纳Handler消息
机制的原理



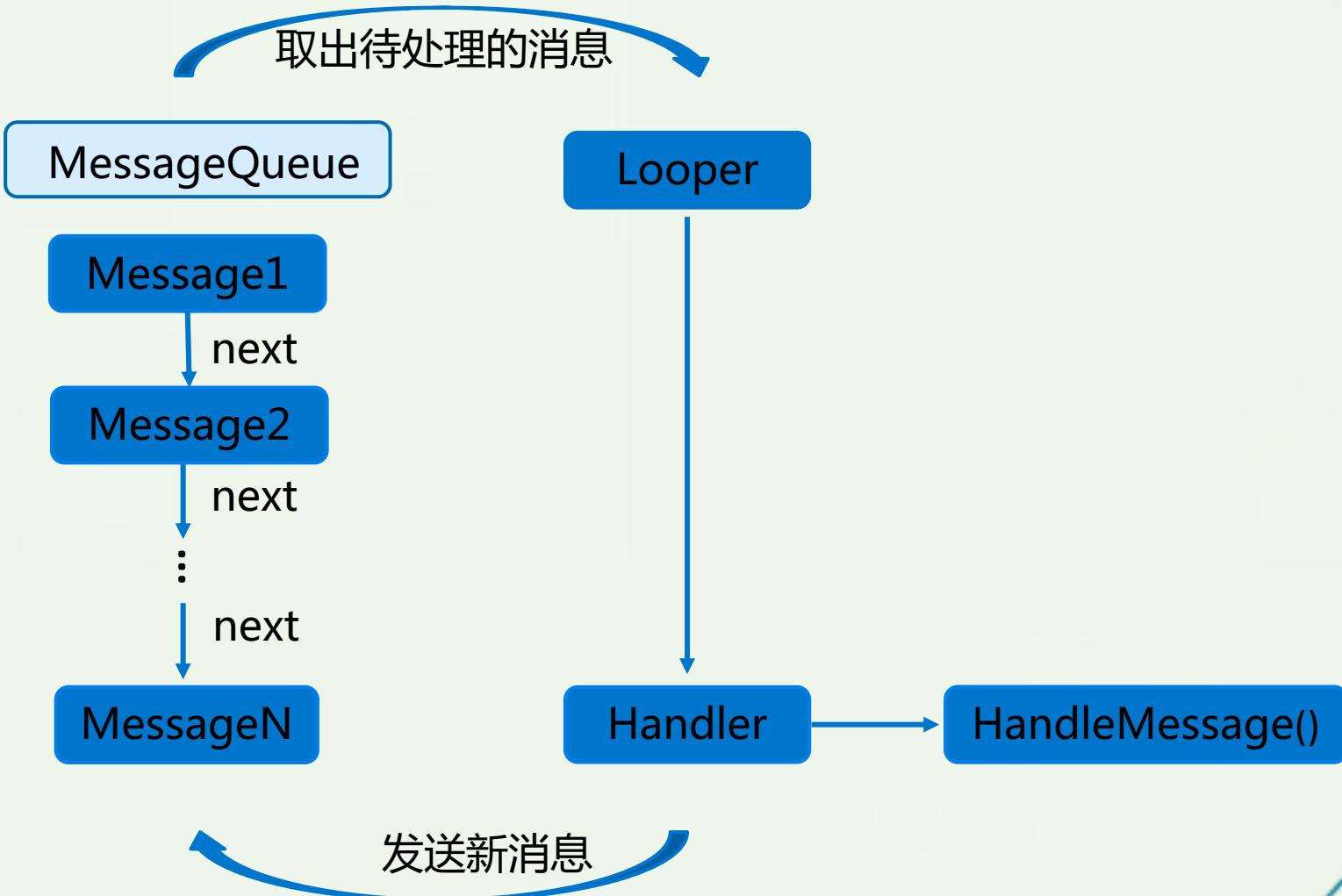
9.4 Handler消息机制

Handler是一种**异步回调机制**，主要**负责与子线程进行通信**。

Handler机制主要包括四个关键对象：

- **Message**：是在线程之间传递的消息，它可以在内部携带少量的信息，用于在不同线程之间交换数据。
- **Handler**：是处理者的意思，它主要用于发送消息和处理消息。
- **MessageQueue**：是消息队列的意思，它主要用来存放通过Handler发送的消息。通过Handler发送的消息会存在MessageQueue中等待处理，每个线程中只会有一个MessageQueue对象。
- **Looper**：是每个线程中的MessageQueue的管家。调用Looper的loop()方法后，就会进入到一个无限循环中。每当发现MessageQueue中存在一条消息，就会将它取出，并传递到Handler的handleMessage()方法中。

9.4 Handler消息机制





本章小结

本 章 小 结

本章详细地讲解了Android中的网络编程。包括HTTP协议、如何使用HttpURLConnection访问网络、提交数据的方式、使用WebView控件浏览网页、WebView控件执行HTML代码、WebView控件支持JavaScript代码、解析JSON数据以及Handler消息机制。在实际开发中大多数应用程序中都需要联网与解析数据，因此希望读者可以熟练掌握本章内容，能更有效率地进行客户端与服务端之间的通信。