

7.1.2 Android Http请求头与响应头的学习

分类 **Android 基础入门教程**

本节引言：

上节中我们对Android涉及的网络编程进行了了解，也学习了下Http的基本概念，而本节我们要学习的是Http的请求头与响应头，当然，可以把也可以把这节看作文档，用到的时候来查查 即可！

1.HTTP请求之消息头：

这里贴下上一节给出的图,根据下面给出的表，大家自己感受下相关请求头的作用吧: PS:第一行是请求行:请求方式 + 资源名称 + HTTP协议版本号，另外请求头只是给服务端的一个 信息而已或者说一个简单，至于怎么处理，还是由服务端来决定的！

```
849 字节 发送到 127.0.0.1:18186
POST /default2.aspx HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/gif, image/pjpeg,
Referer: http://jiaowu.zhbit.com/
Accept-Language: zh-CN
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; QQDownload 7
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Host: jiaowu.zhbit.com
Content-Length: 199
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: ASP.NET_SessionId=ugkdpbz0aw1m5uhoirrac55
VIEWSTATE=dDwyODE2NTM0OTg7Oz5v%2BDe3E3S3%2B1jsAqm%gIDqdbdODw%3D%3DtxtUserName=110202051015gT
```

HTTP Request Header请求头信息对照表：

Header	解释	示例
Accept	指定客户端能够接收的内容类型	Accept: text/plain, text/html
Accept-Charset	浏览器可以接受的字符编码集。	Accept-Charset: iso-8859-5
Accept-Encoding	指定浏览器可以支持的web服务器返回内容压缩编码类型。	Accept-Encoding: compress, gzip
Accept-Language	浏览器可接受的语言	Accept-Language: en,zh
Accept-Ranges	可以请求网页实体的一个或者多个子范围字段	Accept-Ranges: bytes

Android 基础入门教程(Q群号：153836263)

- 1.0 Android基础入门教程
 - 1.0.1 2015年最新Android基...
- 1.1 背景相关与系统架构分析
- 1.2 开发环境搭建
 - 1.2.1 使用Eclipse + ADT + S...
 - 1.2.2 使用Android Studio开...
- 1.3 SDK更新不了问题解决
- 1.4 Genymotion模拟器安装
- 1.5.1 Git使用教程之本地仓...
- 1.5.2 Git之使用GitHub搭建...
- 1.6 .9(九妹)图片怎么玩
- 1.7 界面原型设计
- 1.8 工程相关解析(各种文件...
- 1.9 Android程序签名打包
- 1.11 反编译APK获取代码&...
- 2.1 View与ViewGroup的概念
 - 2.2.1 LinearLayout(线性布局)
 - 2.2.2 RelativeLayout(相对布...
 - 2.2.3 TableLayout(表格布局)
 - 2.2.4 FrameLayout(帧布局)
 - 2.2.5 GridLayout(网格布局)
 - 2.2.6 AbsoluteLayout(绝对...
- 2.3.1 TextView(文本框)详解
- 2.3.2 EditText(输入框)详解
- 2.3.3 Button(按钮)与ImageB...
- 2.3.4 ImageView(图像视图)
- 2.3.5.RadioButton(单选按钮...
- 2.3.6 开关按钮ToggleButton...
- 2.3.7 ProgressBar(进度条)
- 2.3.8 SeekBar(拖动条)
- 2.3.9 RatingBar(星级评分条)
- 2.4.1 ScrollView(滚动条)

Authorization	HTTP授权的授权证书	Authorization: Basic QWxhZGRpbjpvGVuIHNIc2FtZQ==
Cache-Control	指定请求和响应遵循的缓存机制	Cache-Control: no-cache
Connection	表示是否需要持久连接。 (HTTP 1.1默认进行持久连接)	Connection: close
Cookie	HTTP请求发送时，会把保存在该请求域名下的所有cookie值一起发送给web服务器。	Cookie: \$Version=1; Skin=new;
Content-Length	请求的内容长度	Content-Length: 348
Content-Type	请求的与实体对应的MIME信息	Content-Type: application/x-www-form-urlencoded
Date	请求发送的日期和时间	Date: Tue, 15 Nov 2010 08:12:31 GMT
Expect	请求的特定的服务器行为	Expect: 100-continue
From	发出请求的用户的Email	From: user@email.com
Host	指定请求的服务器的域名和端口号	Host: www.zcmhi.com
If-Match	只有请求内容与实体相匹配才有效	If-Match: "737060cd8c284d8af7ad3082f209582d"
If-Modified-Since	如果请求的部分在指定时间之后被修改则请求成功，未被修改则返回304代码	If-Modified-Since: Sat, 29 Oct 2010 19:43:31 GMT
If-None-Match	如果内容未改变返回304代码，参数为服务器先前发送的Etag，与服务器回应的Etag比较判断是否改变	If-None-Match: "737060cd8c284d8af7ad3082f209582d"
If-Range	如果实体未改变，服务器发送客户端丢失的部分，否则发送整个实体。参数也为Etag	If-Range: "737060cd8c284d8af7ad3082f209582d"
If-Unmodified-Since	只在实体在指定时间之后未被修改才请求成功	If-Unmodified-Since: Sat, 29 Oct 2010 19:43:31 GMT
Max-Forwards	限制信息通过代理和网关传送的时间	Max-Forwards: 10
Pragma	用来包含实现特定的指令	Pragma: no-cache
Proxy-Authorization	连接到代理的授权证书	Proxy-Authorization: Basic QWxhZGRpbjpvGVuIHNIc2FtZQ==
Range	只请求实体的一部分，指定范围	Range: bytes=500-999
Referer	先前网页的地址，当前请求网页紧随其后,即来路	Referer: http://blog.csdn.net/coder_pig
TE	客户端愿意接受的传输编码，并通知服务器接受接受尾加头信息	TE: trailers,deflate;q=0.5
Upgrade	向服务器指定某种传输协议	Upgrade: HTTP/2.0, SHHTTP/1.3, IRC/6.9,

- 2.4.2 Date & Time组件(上)
- 2.4.3 Date & Time组件(下)
- 2.4.4 Adapter基础讲解
- 2.4.5 ListView简单实用
- 2.4.6 BaseAdapter优化
- 2.4.7ListView的焦点问题
- 2.4.8 ListView之checkbox错...
- 2.4.9 ListView的数据更新问题
- 2.5.0 构建一个可复用的自定...
- 2.5.1 ListView Item多布局的...
- 2.5.2 GridView(网格视图)的...
- 2.5.3 Spinner(列表选项框)...
- 2.5.4 AutoCompleteTextVie...
- 2.5.5 ExpandableListView(...
- 2.5.6 ViewPager(翻转视图)...
- 2.5.7 Toast(吐司)的基本使用
- 2.5.8 Notification(状态栏通...
- 2.5.9 AlertDialog(对话框)详解
- 2.6.0 其他几种常用对话框基...
- 2.6.1 PopupWindow(悬浮框...
- 2.6.2 菜单(Menu)
- 2.6.3 ViewPager的简单使用
- 2.6.4 DrawerLayout(官方侧...
- 3.1.1 基于监听的事件处理机制
- 3.2 基于回调的事件处理机制
- 3.3 Handler消息传递机制浅析
- 3.4 TouchListener PK OnTo...
- 3.5 监听EditText的内容变化
- 3.6 响应系统设置的事件(Co...
- 3.7 AsyncTask异步任务
- 3.8 Gestures(手势)
- 4.1.1 Activity初学乍练
- 4.1.2 Activity初窥门径
- 4.1.3 Activity登堂入室
- 4.2.1 Service初涉
- 4.2.2 Service进阶
- 4.2.3 Service精通
- 4.3.1 BroadcastReceiver牛...
- 4.3.2 BroadcastReceiver庖...
- 4.4.1 ContentProvider初探

	以便服务器进行转换（如果支持）	RTA/x11
User-Agent	User-Agent的内容包含发出请求的用户信息	User-Agent: Mozilla/5.0 (Linux; X11)
Via	通知中间网关或代理服务器地址，通信协议	Via: 1.0 fred, 1.1 nowhere.com (Apache/1.1)
Warning	关于消息实体的警告信息	Warn: 199 Miscellaneous warning

2.HTTP响应之响应头:

同样给出上节的图: PS:第一行依次是:协议版本号 状态码 302表示这里没有，但是另外一个地方有，通过Location页面重定向了

482 字节 接收于 127.0.0.1:61396

HTTP/1.1 302 Found

Cache-Control: no-cache, no-store
Pragma: no-cache, no-cache
Content-Length: 146
Content-Type: text/html; charset=gb2312
Expires: -1
Location: /xs_main.aspx?xh=110202051015
Server: Microsoft-IIS/7.0
X-AspNet-Version: 1.1.4322
P3P: CP=CAO PSA OUR
X-Powered-By: ASP.NET
Date: Mon, 01 Jun 2015 13:44:27 GMT

<html><head><title>Object moved</title></head><body><h2>Object moved to here.</h2></body></html>

协议版本

各种头

Http回应的内容

空行

返回的内容

http://blog.csdn.net/coder_pig

HTTP Responses Header 响应头信息对照表：

Header	解释	示例
Accept-Ranges	表明服务器是否支持指定范围请求及哪种类型的分段请求	Accept-Ranges: bytes
Age	从原始服务器到代理缓存形成的估算时间（以秒计，非负）	Age: 12
Allow	对某网络资源的有效的请求行为，不允许则返回405	Allow: GET, HEAD
Cache-Control	告诉所有的缓存机制是否可以缓存及哪种类型	Cache-Control: no-cache
Content-Encoding	web服务器支持的返回内容压缩编码类型	Content-Encoding: gzip
Content-Language	响应体的语言	Content-Language: en,zh
Content-Length	响应体的长度	Content-Length: 348
Content-Location	请求资源可替代的备用的另一地址	Content-Location: /index.htm
Content-MD5	返回资源的MD5校验值	Content-MD5: Q2hIY2sgSW50ZWdyaXR5IQ==
Content-Range	在整个返回体中本部分的字节位置	Content-Range: bytes 21010-47021/47022

- 4.4.2 ContentProvider再探...
- 4.5.1 Intent的基本使用
- 4.5.2 Intent之复杂数据的传递
- 5.1 Fragment基本概述
- 5.2.1 Fragment实例精讲—...
- 5.2.2 Fragment实例精讲—...
- 5.2.3 Fragment实例精讲—...
- 5.2.4 Fragment实例精讲—...
- 5.2.5 Fragment实例精讲—...
- 6.1 数据存储与访问之——文...
- 6.2 数据存储与访问之——S...
- 6.3.1 数据存储与访问之——...
- 6.3.2 数据存储与访问之——...
- 7.1.1 Android网络编程要学...
- 7.1.2 Android Http请求头与...
- 7.1.3 Android HTTP请求方...
- 7.1.4 Android HTTP请求方...
- 7.2.1 Android XML数据解析
- 7.2.2 Android JSON数据解析
- 7.3.1 Android 文件上传
- 7.3.2 Android 文件下载（1）
- 7.3.3 Android 文件下载（2）
- 7.4 Android 调用 WebService
- 7.5.1 WebView(网页视图)基...
- 7.5.2 WebView和JavaScrip...
- 7.5.3 Android 4.4后WebVie...
- 7.5.4 WebView文件下载
- 7.5.5 WebView缓存问题
- 7.5.6 WebView处理网页返...
- 7.6.1 Socket学习网络基础准备
- 7.6.2 基于TCP协议的Socket...
- 7.6.3 基于TCP协议的Socket...
- 7.6.4 基于UDP协议的Socke...
- 8.1.1 Android中的13种Draw...
- 8.1.2 Android中的13种Draw...
- 8.1.3 Android中的13种Draw...
- 8.2.1 Bitmap(位图)全解析 P...
- 8.2.2 Bitmap引起的OOM问题
- 8.3.1 三个绘图工具类详解
- 8.3.2 绘图类实战示例

Content-Type	返回内容的MIME类型	Content-Type: text/html; charset=utf-8
Date	原始服务器消息发出的时间	Date: Tue, 15 Nov 2010 08:12:31 GMT
ETag	请求变量的实体标签的当前值	ETag: "737060cd8c284d8af7ad3082f209582d"
Expires	响应过期的日期和时间	Expires: Thu, 01 Dec 2010 16:00:00 GMT
Last-Modified	请求资源的最后修改时间	Last-Modified: Tue, 15 Nov 2010 12:45:26 GMT
Location	用来重定向接收方到非请求URL的位置来完成请求或标识新的资源	Location: http://blog.csdn.net/coder_pig
Pragma	包括实现特定的指令，它可应用到响应链上的任何接收方	Pragma: no-cache
Proxy-Authenticate	它指出认证方案和可应用到代理的该URL上的参数	Proxy-Authenticate: Basic

3.代码验证响应头的作用：

好了，看了那么多概念的东西，不动动手怎么行呢？是吧，那我们就写一些简单的代码来验证一些 常用的响应头的作用吧，以便加深我们的了解，这里的话服务端就用最简单的Servlet来实现，如果不会 Java Web的朋友只需将代码拷一拷，配置下web.xml，把Servlet的类名扣上，比如：

```
<servlet>
    <servlet-name>FirstServlet</servlet-name>
    <servlet-class>com.jay.server.FirstServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>FirstServlet</servlet-name>
    <url-pattern>/FirstServlet</url-pattern>
</servlet-mapping>
```

改成对应的类名即可！

1) 通过Location实现页面重定向

实现代码：

```
package com.jay.http.test;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

- 8.3.3 Paint API之—— Mask...
- 8.3.4 Paint API之—— Xferm...
- 8.3.5 Paint API之—— Xferm...
- 8.3.6 Paint API之—— Xferm...
- 8.3.7 Paint API之—— Xferm...
- 8.3.8 Paint API之—— Xferm...
- 8.3.9 Paint API之—— Color...
- 8.3.10 Paint API之—— Colo...
- 8.3.11 Paint API之—— Colo...
- 8.3.12 Paint API之—— Path...
- 8.3.13 Paint API之—— Sha...
- 8.3.14 Paint几个枚举/常量值...
- 8.3.15 Paint API之——Type...
- 8.3.16 Canvas API详解(Part 1)
- 8.3.17 Canvas API详解(Part...
- 8.3.18 Canvas API详解(Part...
- 8.4.1 Android动画合集之帧...
- 8.4.2 Android动画合集之补...
- 8.4.3 Android动画合集之属...
- 8.4.4 Android动画合集之属...
- 9.1 使用SoundPool播放音...
- 9.2 MediaPlayer播放音频与...
- 9.3 使用Camera拍照
- 9.4 使用MediaRecord录音
- 10.1 TelephonyManager(电...
- 10.2 SmsManager(短信管理...
- 10.3 AudioManager(音频管...
- 10.4 Vibrator(振动器)
- 10.5 AlarmManager(闹钟服务)
- 10.6 PowerManager(电源服...
- 10.7 WindowManager(窗口...
- 10.8 LayoutInflater(布局服务)
- 10.9 WallpaperManager(壁...
- 10.10 传感器专题(1)——相...
- 10.11 传感器专题(2)——方...
- 10.12 传感器专题(3)——加...
- 10.12 传感器专题(4)——其...
- 10.14 Android GPS初涉
- 11.0 《2015最新Android基...

```

public class ServletOne extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        //告诉浏览器响应码, 以及重定向页面
        resp.setStatus(302);
        resp.setHeader("Location", "http://www.baidu.com");
    }
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        this.doGet(req, resp);
    }
}

```

运行结果：

当我们去访问: <http://localhost:8080/HttpTest/ServletOne> 的时候, 我们会发现页面跳转到了百度, 接着我们用FireFox的开发者工具: 可以看到我们发出的HTTP的内容:

响应头：

```

Content-Length: 0
Date: Tue, 02 Jun 2015 03:08:02 GMT
Location: http://www.baidu.com
Server: Apache-Coyote/1.1

```

2) 通过Content-Encoding告诉浏览器数据的压缩格式

实现代码:

```

package com.jay.http.test;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.zip.GZIPOutputStream;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ServletTwo extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        String data = "Fresh air and sunshine can have an amazing effect on our feelings. "
            + "Sometimes when we are feeling down, all that we need to do is simply to go "
            + "outside and breathe. Movement

```



```

t and exercise is also a fantastic way to feel better. "
        + "Positive emotions can be generated by motion. So if we start to feel down,"
        + " take some deep breathes, go outside, feel the fresh air, "
        + "let the sun hit our face, go for a hike, a walk, a bike ride, "
        + "a swim, a run, whatever. We will feel better if we do this.";
        System.out.println("原始数据长度:" + data.getBytes().length);
        // 对数据进行压缩:
        ByteArrayOutputStream bout = new ByteArrayOutputStream();
        GZIPOutputStream gout = new GZIPOutputStream(bout);

        gout.write(data.getBytes());
        gout.close();
        // 得到压缩后的数据
        byte gdata[] = bout.toByteArray();
        resp.setHeader("Content-Encoding", "gzip");
        resp.setHeader("Content-Length", gdata.length + "");

        resp.getOutputStream().write(gdata);

    }

    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        doGet(req, resp);
    }
}

```

运行结果：

控制台输出:

```
原始数据长度:478
```

浏览器输出:

```
Fresh air and sunshine can have
exercise is also a fantastic way
air, let the sun hit our face, g
```

再看看我们的HTTP内容:

响应头：

```
Content-Encoding: gzip
Content-Length: 285
Date: Tue, 02 Jun 2015 03:50:27 GMT
Server: Apache-Coyote/1.1
```

```
1 Fresh air and sunshine can have an amaz
```

这个gzip压缩字符串对于简单的字符串压缩，效率不高，比如小猪本来写的是一个一首静夜诗的字符串，后来发现压缩过后的大小，竟然比原先的还要大==...

3) 通过content-type , 设置返回的数据类型

服务端返回的有时可能是一个text/html, 有时也可能是一个image/jpeg, 又或者是一段视频video/avi 浏览器可以根据这个对应的数据类型, 然后以不同的方式将数据显示出来! 好吧, 这里我们弄一个读PDF的

实现代码 :

```
package com.jay.http.test;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ServletThree extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        resp.setHeader("content-type", "application/pdf");

        InputStream in = this.getServletContext().getResourceAsStream("/file/android编码规范.pdf");
        byte buffer[] = new byte[1024];
        int len = 0;
        OutputStream out = resp.getOutputStream();
        while((len = in.read(buffer)) > 0)
        {
            out.write(buffer, 0, len);
        }

        protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
            ServletException, IOException
        {
            doGet(req, resp);
        };
    }
}
```

运行结果 :

在浏览器上输入: <http://localhost:8080/HttpTest/ServletThree>

好啦, 果然可以读到PDF了, 对了, 这个PDF我已经丢在webroot的file目录下, 不然会报空指针哦~:

当然，你也可以试着去播放一段音乐或者视频，只需修改下content-type这个参数而已

下面顺便给出个HTTP Content-type的对照表吧: [HTTP Content-type的对照表](#)



Android 编码规范

http://blog.csdn.net/coder_pig

4) 通过refresh响应头，让浏览器隔几秒后跳转至别的页面

恩呢，一般我们可能有这样的需求，比如每隔几秒刷新一次页面，又或者加载某个页面几秒后又跳转至另一个页面，那么refresh可以满足你的需要~

实现代码：

```
package com.jay.http.test;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ServletFour extends HttpServlet {
    public int second = 0;
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        //1.浏览器每隔2秒定时刷新页面
        resp.setHeader("refresh", "2");
        // resp.getWriter().write(++second + "");
        // System.out.println("doGet方法被调用~");

        //2.进入页面5s后，然页跳到百度~
        resp.setHeader("refresh", "5;url='http://www.baidu.com'");
        resp.getWriter().write("HE HE DA~");
    }

    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        {
            doGet(req, resp);
        }
    }
}
```



```
};  
}
```

运行结果：

1的话每隔2秒刷新一次页面，我们可以看到显示的数字是递增的，另外doGet方法也一直被调用，说明页面真的是刷新的！

2的话进入页面后5s，就自己跳转到百度了~

5) 通过content-dispostion响应头，让浏览器下载文件

这个很简单，我们只需把③中设置Content-type的一行去掉，然后加上：**resp.setHeader("content-disposition", "attachment;filename=Android.pdf");**

实现代码：

```
package com.jay.http.test;  
  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.OutputStream;  
  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
  
public class ServletFive extends HttpServlet {  
  
    @Override  
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)  
        throws ServletException, IOException {  
        resp.setHeader("content-disposition", "attachment;filename=Android.pdf");  
        InputStream in = this.getServletContext().getResourceAsStream("/file/android编码规范.pdf");  
        byte buffer[] = new byte[1024];  
        int len = 0;  
        OutputStream out = resp.getOutputStream();  
        while((len = in.read(buffer)) > 0)  
        {  
            out.write(buffer, 0, len);  
        }  
    }  
  
    @Override  
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
```



反馈

```
throws ServletException, IOException {  
    doGet(req, resp);  
}  
}
```

运行结果：



本节小结：

本节给大家介绍了Http中的请求头和响应头，也写了几个关于响应头调浏览器的一些示例，相信经过本章，大家对于Http协议更加了解了，下节我们来学习Android给我们提供的Http 请求方式:HttpURLConnection！好的，本节就到这里，谢谢~ 对了，本节demo下载：[下载 HttpTest.zip](#)

坚持一个月, 看美剧不用字幕

每天只要 5分钟, 坚持30天, 效果真的不一样!

立即行动

在线实例

- HTML 实例
- CSS 实例
- JavaScript 实例
- Ajax 实例
- jQuery 实例

字符集&工具

- HTML 字符集设置
- HTML ASCII 字符集
- HTML ISO-8859-1
- HTML 实体符号

最新更新

- C语言- 打...
- Win7和Linux下的...
- Linux下 Nginx+To...
- linux安装svn服...

站点信息

- 意见反馈
- 免责声明
- 关于我们
- 文章归档

<ul style="list-style-type: none">· XML 实例· Java 实例	<ul style="list-style-type: none">· HTML 拾色器· JSON 格式化工具	<ul style="list-style-type: none">· PHP 换行符 PHP_EOL· PHP 7 新特性· Linux Crontab ...
--	---	---

关注微信



Copyright © 2013-2015 菜鸟教程 **runoob.com** All Rights Reserved. 备案号：闽ICP备15012807号-1