



Java 核心技术(进阶)

第六章 Java 网络编程

第五节 Java HTTP 编程 (HttpClient)

华东师范大学 陈良育



HttpClient

- JDK HTTP Client (JDK自带, 从9开始)
- Apache HttpComponents的HttpClient (Apache出品)



JDK HttpClient

- JDK 9 新增, JDK10更新, JDK11正式发布
- java.net.http 包
- 取代URLConnection
- 支持HTTP/1.1和HTTP/2
- 实现大部分HTTP方法
- 主要类
 - HttpClient
 - HttpRequest
 - HttpResponse



HttpComponents

- hc.apache.org, Apache 出品
- 从HttpClient进化而来
- 是一个集成的Java HTTP工具包
 - 实现所有HTTP方法: get/post/put/delete
 - 支持自动转向
 - 支持https协议
 - 支持代理服务器等

进一步延伸



- 爬虫
- 自动刷票机器人
- 各类Web监控
- Web回归测试
- ○ ○ ○

总结



- 了解JDK 11自带的Java HTTP Client 包
- 了解Apache Http Components 包
- 了解Java网络编程的进一步应用

代码(1) JDKHttpClientGetTest.java



```
public class JDKHttpClientGetTest {  
  
    public static void main(String[] args) throws IOException, InterruptedException {  
        doGet();  
    }  
  
    public static void doGet() {  
        try{  
            HttpClient client = HttpClient.newHttpClient();  
            HttpRequest request = HttpRequest.newBuilder(URI.create("http://www.baidu.com")).build();  
            HttpResponse response = client.send(request, HttpResponse.BodyHandlers.ofString());  
            System.out.println(response.body());  
        }  
        catch(Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

代码(2) JDKHttpClientPostTest.java



```
public class JDKHttpClientPostTest {  
  
    public static void main(String[] args) throws IOException, InterruptedException {  
        doPost();  
    }  
  
    public static void doPost() {  
        try {  
            HttpClient client = HttpClient.newBuilder().build();  
            HttpRequest request = HttpRequest.newBuilder()  
                .uri(URI.create("https://tools.usps.com/go/ZipLookupAction.action"))  
                //.header("Content-Type", "application/x-www-form-urlencoded")  
                .header("User-Agent", "HTTPie/0.9.2")  
                .header("Content-Type", "application/x-www-form-urlencoded; charset=utf-8")  
                .method("POST", HttpRequest.BodyPublishers.ofString("tAddress=1 Market S  
                //.version(Version.HTTP_1_1)
```


代码(3) JDKHttpClientPostTest.java



```
.POST(HttpRequest.BodyPublishers.ofString("tAddress="
    + URLEncoder.encode("1 Market Street", "UTF-8")
    + "&tCity=" + URLEncoder.encode("San Francisco", "UTF-8") + "&sState=CA"))
//.POST(HttpRequest.BodyPublishers.ofString("tAddress=" + URLEncoder.encode("1 Market Street", "UTF-8")
    .build());
HttpResponse response = client.send(request, HttpResponse.BodyHandlers.ofString());
System.out.println(response.statusCode());
System.out.println(response.headers());
System.out.println(response.body().toString());
}
catch(Exception e) {
    e.printStackTrace();
}
}
```

代码(4) HttpComponentsGetTest.java



```
public class HttpComponentsGetTest {  
  
    public final static void main(String[] args) throws Exception {  
  
        CloseableHttpClient httpClient = HttpClient.createDefault();  
        RequestConfig requestConfig = RequestConfig.custom()  
            .setConnectTimeout(5000) //设置连接超时时间  
            .setConnectionRequestTimeout(5000) // 设置请求超时时间  
            .setSocketTimeout(5000)  
            .setRedirectsEnabled(true) //默认允许自动重定向  
            .build();  
  
        HttpGet httpGet = new HttpGet("http://www.baidu.com");  
        httpGet.setConfig(requestConfig);  
        String srtResult = "";
```

代码(5) HttpClientGetTest.java



```
try {
    HttpResponse httpResponse = httpClient.execute(httpGet);
    if(httpResponse.getStatusLine().getStatusCode() == 200){
        srtResult = EntityUtils.toString(httpResponse.getEntity(), "UTF-8");//获得返回的结果
        System.out.println(srtResult);
    }else
    {
        //异常处理
    }
} catch (IOException e) {
    e.printStackTrace();
}finally {
    try {
        httpClient.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}
```


代码(6) HttpComponentsPostTest.java



```
public class HttpComponentsPostTest {  
  
    public final static void main(String[] args) throws Exception {  
  
        //获取可关闭的 httpCilent  
        //CloseableHttpClient httpClient = HttpClientBuilder.create().setRedirectStrategy(new LaxRedirectStrategy()).build();  
        CloseableHttpClient httpClient = HttpClientBuilder.create().setRedirectStrategy(new LaxRedirectStrategy()).build();  
        //配置超时时间  
        RequestConfig requestConfig = RequestConfig.custom().  
            setConnectTimeout(10000).setConnectionRequestTimeout(10000)  
            .setSocketTimeout(10000).setRedirectsEnabled(false).build();  
  
        HttpPost httpPost = new HttpPost("https://tools.usps.com/go/ZipLookupAction.action");  
        //设置超时时间  
        httpPost.setConfig(requestConfig);  
    }  
}
```

代码(7) HttpComponentsPostTest.java



//装配post请求参数

```
List<BasicNameValuePair> list = new ArrayList<BasicNameValuePair>();
```

```
list.add(new BasicNameValuePair("tAddress", URLEncoder.encode("1 Market Street", "UTF-8"))); //请求参数
```

```
list.add(new BasicNameValuePair("tCity", URLEncoder.encode("San Francisco", "UTF-8"))); //请求参数
```

```
list.add(new BasicNameValuePair("sState", "CA")); //请求参数
```

```
try {
```

```
    UrlEncodedFormEntity entity = new UrlEncodedFormEntity(list, "UTF-8");
```

```
    //设置post请求参数
```

```
    httpPost.setEntity(entity);
```

```
    httpPost.setHeader("User-Agent", "HTTPie/0.9.2");
```

```
    //httpPost.setHeader("Content-Type", "application/form-data");
```

```
    HttpResponse httpResponse = httpClient.execute(httpPost);
```

```
    String strResult = "";
```


代码(8) HttpClientPostTest.java



```
        if(httpResponse != null){
            System.out.println(httpResponse.getStatusLine().getStatusCode());
            if (httpResponse.getStatusLine().getStatusCode() == 200) {
                strResult = EntityUtils.toString(httpResponse.getEntity());
            }
            else {
                strResult = "Error Response: " + httpResponse.getStatusLine().toString();
            }
        }else{
        }
        System.out.println(strResult);
    } catch (Exception e) {
        e.printStackTrace();
    }finally {
        try {
            if(httpClient != null){
                httpClient.close(); //释放资源
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```



谢谢!