



# Java 核心技术(进阶)

第六章 Java 网络编程

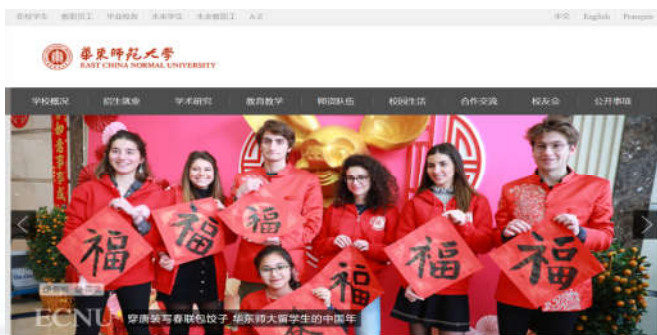
第四节 Java HTTP 编程

华东师范大学 陈良育

# 网页访问



- 网页是特殊的网络服务(HTTP, Hypertext Transfer Protocol)
  - 在浏览器输入URL地址
  - 浏览器将连接到远程服务器上(IP+80Port)
  - 请求下载一个HTML文件下来, 放到本地临时文件夹中
  - 在浏览器显示出来



```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,user-scalable=0,
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<meta http-equiv="pragma" content="no-cache" />
<meta content="yes" name="apple-mobile-web-app-capable" />
<meta content="telephone=no" name="format-detection" />
<meta name="baidu-site-verification" content="sefVIMkGoW" />
<meta name="google-site-verification" content="7qvJN_s810dDdDn12kj
<title>华东师范大学</title>

<link type="text/css" href="/css/system/system.css" rel="stylesh
<link type="text/css" href="/upload/site/1/style/87/87.css" rel="
<link type="text/css" href="/upload/site/00/02/2/style/92/92.css"
<LINK href="/css/tpl2/system.css" type="text/css" rel="sty
<LINK href="/css/tpl2/default/default.css" type="text/css"
```

Internet Explorer 存储网页、图像和媒体的副本以便以后快速查看。

检查存储的页面的较新版本:

- ☐ 每次访问网页时(E)
- ☐ 每次启动 Internet Explorer 时(S)
- ☒ 自动(A)
- ☐ 从不(N)

使用的磁盘空间(8-1024MB)(D)  
(推荐: 50-250MB)

330

当前位置:

C:\Users\Tom\AppData\Local\Microsoft\Windows\NetCache\



# HTTP

- HTTP

- 超文本传输协议(HyperText Transfer Protocol)
- 用于从WWW (World Wide Web) 服务器传输超文本到本地浏览器的传输协议
- 1989年蒂姆·伯纳斯·李 (Tim Berners Lee) 提出了一种能让远隔两地的研究者们共享知识的设想
- 借助多文档之间相互关联形成的超文本 (HyperText) , 连成可相互参阅的 WWW
- 1990年问世, 1997年发布版本1.1, 2015年发布版本2.0
- 资源文件采用HTML编写, 以URL形式对外提供



# HTML

- HTML

- 超文本标记语言(HyperText Markup Language)
- 标准语法 <http://www.w3school.com.cn/html/index.asp>
- 表单 form

```
<html>
  <title>登陆</title>
  <body>
    <form name='form1'>
      id: <input type='text' name='id'><br>
      name: <input type='text' name='name'><br>
      <input type='submit' value='ok'>
    </form>
  </body>
</html>
```

id:

name:





# HTTP访问方式

- 访问方式

- GET: 从服务器获取资源到客户端
- POST: 从客户端向服务器发送数据
- PUT: 上传文件
- DELETE: 删除文件
- HEAD: 报文头部
- OPTIONS: 询问支持的方法
- TRACE: 追踪路径
- CONNECT: 用隧道协议连接代理





# Java HTTP 编程

- Java HTTP 编程 (java.net 包)
  - 支持模拟成浏览器的方式去访问网页
  - URL, Uniform Resource Locator, 代表一个资源
    - <http://www.ecnu.edu.cn/index.html?a=1&b=2&c=3>
  - URLConnection
    - 获取资源的连接器
    - 根据URL的openConnection()方法获得URLConnection
    - connect方法, 建立和资源的联系通道
    - getInputStream方法, 获取资源的内容

# 总结



- 了解HTTP的基础概念
- 掌握基于URLConnection的HTTP编程

# 代码(1)URLConnectionGetTest.java



```
public class URLConnectionGetTest
{
    public static void main(String[] args)
    {
        try
        {
            String urlName = "http://www.baidu.com";

            URL url = new URL(urlName);
            URLConnection connection = url.openConnection();
            connection.connect();

            // 打印http的头部信息

            Map<String, List<String>> headers = connection.getHeaderFields();
            for (Map.Entry<String, List<String>> entry : headers.entrySet())
            {
                String key = entry.getKey();
                for (String value : entry.getValue())
                    System.out.println(key + ": " + value);
            }
        }
    }
}
```





## 代码(2) URLConnectionGetTest.java

// 输出将要收到的内容属性信息

```
System.out.println("-----");
System.out.println("getContentType: " + connection.getContentType());
System.out.println("getContentLength: " + connection.getContentLength());
System.out.println("getContentEncoding: " + connection.getContentEncoding());
System.out.println("getDate: " + connection.getDate());
System.out.println("getExpiration: " + connection.getExpiration());
System.out.println("getLastModified: " + connection.getLastModified());
System.out.println("-----");
```

```
BufferedReader br = new BufferedReader(new InputStreamReader(connection.getInputStream(), "UTF-8"));
```

// 输出收到的内容

```
String line = "";
while((line=br.readLine()) != null)
{
    System.out.println(line);
}
br.close();
```

```
catch (IOException e)
{
    e.printStackTrace();
}
```

```
}
}
```

# 代码(3) URLConnectionPostTest.java



```
public class URLConnectionPostTest
{
    public static void main(String[] args) throws IOException
    {
        String urlString = "https://tools.usps.com/go/ZipLookupAction.action";
        Object userAgent = "HTTPie/0.9.2";
        Object redirects = "1";
        CookieHandler.setDefault(new CookieManager(null, CookiePolicy.ACCEPT_ALL));

        Map<String, String> params = new HashMap<String, String>();
        params.put("tAddress", "1 Market Street");
        params.put("tCity", "San Francisco");
        params.put("sState", "CA");
        String result = doPost(new URL(urlString), params,
            userAgent == null ? null : userAgent.toString(),
            redirects == null ? -1 : Integer.parseInt(redirects.toString()));
        System.out.println(result);
    }
}
```



## 代码(4) HttpURLConnectionPostTest.java

```
public static String doPost(URL url, Map<String, String> nameValuePairs, String userAgent, int redirects)
    throws IOException
{
    HttpURLConnection connection = (HttpURLConnection) url.openConnection();
    if (userAgent != null)
        connection.setRequestProperty("User-Agent", userAgent);

    if (redirects >= 0)
        connection.setInstanceFollowRedirects(false);

    connection.setDoOutput(true);
```





# 代码(5) URLConnectionPostTest.java

```
//输出请求的参数
try (PrintWriter out = new PrintWriter(connection.getOutputStream()))
{
    boolean first = true;
    for (Map.Entry<String, String> pair : nameValuePairs.entrySet())
    {
        //参数必须这样拼接 a=1&b=2&c=3
        if (first)
        {
            first = false;
        }
        else
        {
            out.print('&');
        }
        String name = pair.getKey();
        String value = pair.getValue();
        out.print(name);
        out.print('=');
        out.print(URLEncoder.encode(value, "UTF-8"));
    }
}
```



# 代码(6) HttpURLConnectionPostTest.java



```
String encoding = connection.getContentEncoding();
if (encoding == null)
{
    encoding = "UTF-8";
}

if (redirects > 0)
{
    int responseCode = connection.getResponseCode();
    System.out.println("responseCode: " + responseCode);
    if (responseCode == HttpURLConnection.HTTP_MOVED_PERM
        || responseCode == HttpURLConnection.HTTP_MOVED_TEMP
        || responseCode == HttpURLConnection.HTTP_SEE_OTHER)
    {
        String location = connection.getHeaderField("Location");
        if (location != null)
        {
            URL base = connection.getURL();
            connection.disconnect();
            return doPost(new URL(base, location), nameValuePairs, userAgent, redirects - 1);
        }
    }
}
else if (redirects == 0)
{
    throw new IOException("Too many redirects");
}
```

# 代码(7) URLConnectionPostTest.java



//接下来获取html 内容

```
StringBuilder response = new StringBuilder();
try (Scanner in = new Scanner(connection.getInputStream(), encoding))
{
    while (in.hasNextLine())
    {
        response.append(in.nextLine());
        response.append("\n");
    }
}
catch (IOException e)
{
    InputStream err = connection.getErrorStream();
    if (err == null) throw e;
    try (Scanner in = new Scanner(err))
    {
        response.append(in.nextLine());
        response.append("\n");
    }
}

return response.toString();
}
```



谢谢!