

Java EE 企业应用系统设计

HTTP 请求处理编程

王晓东

wangxiaodong@ouc.edu.cn

中国海洋大学

November 8, 2018



大纲

HTTP 请求内容

Java EE 请求对象



接下来…

HTTP 请求内容

Java EE 请求对象



Web 工作模式

Web 通常使用**请求—响应**模式。

- ▶ 客户端（浏览器）向服务器发出 HTTP 请求，在 HTTP 请求中包含传递到服务器的数据；
- ▶ Web 服务器接收到请求，对请求进行处理。
- ▶ Web 服务器使用 HTTP 向客户端发送响应。
- ▶ 客户端接收到响应后，进行显示或页面跳转。



HTTP 请求中包含的信息

HTTP 请求中包含的信息包括两部分：**请求头**和**请求体**。

❖ 请求头

```
1 GET /articles/news/today.jsp HTTP/1.1
2 Accept: */*
3 Accept-Language: en-us
4 Connection: Keep-Alive
5 Host: localhost
6 Referer: http://localhost/links.asp
7 User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)
8 Accept-Encoding: gzip, deflate
9 ...
```



HTTP 请求中包含的信息

❖ HTTP 请求头标记和说明

User-Agent 浏览器的机器环境

Accept 浏览器支持哪些 MIME 数据类型

Accept-Charset 浏览器支持的字符编码

Accept-Encoding 浏览器支持哪种数据压缩格式

Accept-Language 浏览器指定的语言环境

Host 浏览器访问的主机名

Referer 浏览器是从哪个页面来的

Cookie 浏览器保存的 cookie 对象

Java EE Web 组件 Servlet 和 JSP 中可以使用请求对象的方法读取这些请求内容，进而进行相应的处理。



HTTP 请求中包含的信息

❖ 请求体

每次 HTTP 请求时，在请求头之后会有一个**空行**，接下来是请求中包含的提交数据，即**请求体**。

❶ GET 请求

无请求体，请求数据直接在请求的 URL 地址中，作为 URL 的一部分发送给 Web 服务器。

```
1 http://localhost:8080/webapp/login.do?id=9001&pass=9001
```

- ▶ 请求体为空，提交数据直接在 URL 上，作为请求头部分传输到 Web 服务器，通过 URL 的 QueryString 部分能得到提交的参数数据。
- ▶ 此种方式对提交数据的大小有限制，不同浏览器会有所不同，如 IE 为 2083 字节。GET 请求时数据会出现在 URL 中，保密性差，实际编程中要尽量避免。



HTTP 请求中包含的信息

❖ 请求体

每次 HTTP 请求时，在请求头之后会有一个**空行**，接下来是请求中包含的提交数据，即**请求体**。

❶ GET 请求

无请求体，请求数据直接在请求的 URL 地址中，作为 URL 的一部分发送给 Web 服务器。

```
1 http://localhost:8080/webapp/login.do?id=9001&pass=9001
```

- ▶ 请求体为空，提交数据直接在 URL 上，作为请求头部分传输到 Web 服务器，通过 URL 的 QueryString 部分能得到提交的参数数据。
- ▶ 此种方式对提交数据的大小有限制，不同浏览器会有所不同，如 IE 为 2083 字节。GET 请求时数据会出现在 URL 中，保密性差，实际编程中要尽量避免。



HTTP 请求中包含的信息

② POST 请求

- ▶ 请求体数据单独打包为数据块，通过 Socket 直接传递到 Web 服务器端，数据不会在地址栏出现。
- ▶ 可以提交大的数据，包括二进制文件，实现文件上传功能。
- ▶ 原则上 POST 请求对提交的数据没有大小限制。



接下来…

HTTP 请求内容

Java EE 请求对象



请求对象类型与生命周期

❖ 请求对象接口类型

- ▶ Java EE 规范中的通用请求对象要实现接口
`javax.servlet.ServletException`
- ▶ HTTP 请求对象要实现接口
`javax.servlet.http.HttpServletRequest`

❖ 请求对象生命周期

在 Java Web 组件开发中，不需要 Servlet 或 JSP 自己创建请求对象，它们由 Web 容器自动创建，并传递给 Servlet 和 JSP 的服务方法 `doGet` 和 `doPost`，在服务处理方法中直接使用请求对象即可。



请求对象类型与生命周期

❖ 请求对象创建

每次 Web 服务器接收到 HTTP 请求时，会自动创建实现 `HttpServletRequest` 接口的对象。在创建该对象之后，Web 服务器将请求头和请求体信息写入请求对象，Servlet 和 JSP 可以通过请求对象的方法取得这些请求信息，继而可以取得用户提交的数据。

❖ 请求对象销毁

当 Web 服务器处理 HTTP 请求，向客户端发送 HTTP 响应结束后，会自动销毁请求对象，保存在请求对象中的数据随即丢失。当下次请求时新的请求对象又会被创建。



请求对象功能与方法

请求对象方法一般分类为：

- ▶ 取得请求头信息。
- ▶ 取得请求体中包含的提交参数数据，包含表单元素或地址栏 URL 的参数。
- ▶ 取得客户端的有关信息，如请求协议、IP 地址和端口等。
- ▶ 取得服务器端的相关信息，如服务器的 IP 等。
- ▶ 取得请求对象的属性信息，用于在一个请求的转发对象之间传递数据。



取得请求头方法

- ▶ String getHeader(String name)

```
1 String browser = request.getHeader("User-Agent");
```

- ▶ int getIntHeader(String name)

```
1 int size = request.getIntHeader("Content-Length");
```

- ▶ long getDateHeader(String name)

```
1 long datetime = request.getDateHeader("If-Modified-Since");
```

- ▶ Enumeration getHeaderNames()

```
1 for (Enumeration enume = request.getHeaderNames(); enume.hasMoreElements(); ) {  
2     String headerName = (String) enume.nextElement();  
3     System.out.println("Name: " + headerName);  
4 }
```



取得请求中包含的提交参数数据

在 Web 开发中，用户通过表单输入将客户端数据提交到服务器端，这些数据被 Web 服务器自动保存到请求对象中。Web 组件 Servlet 和 JSP 可以通过请求对象获得用户提交的数据。

❖ `String getParameter(String name)` 取得指定名称的参数数据。

参数 name 为 FORM 表单元素的 name 或 URL 参数名称：

产品名称:<input type="text" name="productName" />
productSearch.do?productName=Acer

如下代码可以取得以上的参数名为 productName 的数据：

```
1 String productName = request.getParameter("productName");
```



取得请求中包含的提交参数数据

❖ `String[] getParameterValues(String name)` 取得指定名称的

参数数据数组，如复选框和复选列表。

如下代码取得复选框的选择数据：

爱好：

```
<input type="checkbox" name="behave" value="旅游">旅游  
<input type="checkbox" name="behave" value="读书">读书  
<input type="checkbox" name="behave" value="体育">体育
```

如下代码取得选定的爱好：

```
1 String[] behaves = request.getParameterValues("behave");  
2 for (int i = 0; i < behaves.length; i++) {  
3     out.println(behaves[i]);  
4 }
```



取得请求中包含的提交参数数据

❖ Enumeration `getParameterNames()` 取得所有请求的参数名称。

```
1 for (Enumeration enums = request.getParameterNames(); enums.hasMoreElements();) {  
2     String paramName = (String) enums.nextElement();  
3     System.out.println(paramName);  
4 }
```



取得请求中包含的提交参数数据

❖ `Map getParameterMap()` 取得所有请求的参数名和值，包装在一个 `Map` 对象中，可以使用这个对象同时取得所有的参数名和参数值。

```
1 Map params = request.getParameterMap();
2 Set names = params.keySet();
3 for (Object o: names) {
4     String paramName = (String) o;
5     out.print(paramName + "␣=␣" + params.get(paramName) + "<br/>");
6 }
```



取得请求中包含的提交参数数据

在 Web 开发中，用户通过表单输入将客户端数据提交到服务器端，这些数据被 Web 服务器自动保存到请求对象中。Web 组件 Servlet 和 JSP 可以通过请求对象获得用户提交的数据。

❖ `Map getParameterMap()` 取得所有请求的参数名和值，包装在一个 Map 对象中。

```
1 Map params = request.getParameterMap();
2 Set names = params.keySet();
3 for (Object o: names) {
4     String paramName = (String) o;
5     out.println(paramName + "␣=␣" + params.get(paramName) + "<br/>");
6 }
```



取得请求中包含的提交参数数据

❖ `ServletInputStream` `getInputStream()` throws `IOException`

取得客户端的输入流。

当使用 `getParameter()` 方法后，就无法使用 `getInputStream()` 方法，反之亦然。

当用户提交的数据中包含文件上传时，提交的数据就可以以二进制编码方式提交到服务器。当表单既有文本字段还有文件上传时，就需要对此二进制流进行解析，从而分离出文本和上传文件。可以使用第三方框架实现上传文件的大额处理，如 Apache 的 Common upload 组件。

```
<form action="" method="" enctype="multipart/form-data" />
姓名: <input type="text" name="name" /><br />
照片: <input type="file" name="photo" /><br />
<input type="submit" value="提交" />
</form>
```



取得其他客户端信息

❖ `String getRemoteHost()`

取得请求客户的主机名。

❖ `String getRemoteAddr()`

取得请求客户端的 IP 地址。

❖ `int getRemotePort()`

取得请求客户的端口号。

❖ `String getProtocol()`

取得请求的协议。



取得其他客户端信息

❖ `String getContentType()`

取得请求体的内容类型，以 MIME 表达。

❖ `int getContentLength()`

取得请求体为二进制流时请求体的长度，一般当处理文件上传时使用。

❖ `String getProtocol()`

取得请求的协议，一般为 HTTP，返回 HTTP1.1。



取得服务器端信息

❖ `String getServerName()`

取得服务器的 HOST，一般为 IP 地址。

❖ `int getServerPort()`

取得服务器接收端口。



应用实例

❖ EE_4.3 取得 HTML 表单提交的数据





THE END

wangxiaodong@ouc.edu.cn

