04 HTTP协议

- **1 概念**

  - 概念：Hyper Text Transfer Protocol，超文本传输协议，规定了浏览器和服务器之间数据传输的规则。

- **2 特点**

  1. 基于TCP协议：面向连接，安全
  2. 基于请求-响应模型的：一次请求对应一次响应
  3. HTTP协议是无状态的协议：对于事务处理没有记忆能力。每次请求-响应都是独立的。
     - 缺点：多次请求间不能共享数据。
     - 优点：速度快

- **3 HTTP-请求协议**

请求行：请求数据第一行
（请求方式、资源路径、协议）

请求头：第二行开始，格式key：value

请求体：POST请求，存放请求参数

```
GET /brand/findAll?name=OPPO&status=1 HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/...
```

```
POST /brand HTTP/1.1
Accept: application/json, text/plain, */*
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9
Content-Length: 161
Content-Type: application/json;charset=UTF-8
Cookie: Idea-8296eb32=841b16f0-0cfe-495a-9cc9-d5aaa71501a6; JSESSIONID=0FDE4E430876BD9C5C955F061207386F
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/...

{"status":1,"brandName":"黑马","companyName":"黑马程序员","id":"","description":"黑马程序员"}
```

Host	请求的主机名
User-Agent	浏览器版本，例如Chrome浏览器的标识类似Mozilla/5.0 ... Chrome/79，IE浏览器的标识类似Mozilla/5.0 (Windows NT ...) like Gecko
Accept	表示浏览器能接收的资源类型，如text/*，image/*或者*/*表示所有；
Accept-Language	表示浏览器偏好的语言，服务器可以据此返回不同语言的网页；
Accept-Encoding	表示浏览器可以支持的压缩类型，例如gzip，deflate等。
Content-Type	请求主体的数据类型。
Content-Length	请求主体的大小（单位：字节）。

. Http协议中请求数据分为哪几个部分？

- 请求行（请求数据的第一行）
- 请求头（key：value）
- 请求体（与请求头之间隔了一个空行）

- **4 请求数据获取**
  - Web服务器(Tomcat)对HTTP协议的请求数据进行解析，并进行了封装(HttpServletRequest)，在调用Controller方法的时候传递给了该方法。这样，就使得程序员不必直接对协议进行操作，让Web开发更加便捷。

HelloController

```java
@RequestMapping("/request")
public String request(HttpServletRequest request){
    //  1.获取请求参数name, age
    String name = request.getParameter("name"); // Tom
    //  2.获取请求路径uri 和 url
    String uri = request.getRequestURI(); // /request
    String url = request.getRequestURL().toString(); // http://localhost:8080/request
    //  3.获取请求头 User-Agent
    String userAgent = request.getHeader("User-Agent"); // Mozilla/5.0 (Windows NT 10.0; Win64; x64)
    //  4.获取请求方式
    String method = request.getMethod(); // GET
    //  5.获取请求的查询字符串
    String queryString = request.getQueryString(); // name=Tomcat&age=10
    return "request success";
}
```

1．HTTP请求数据需要程序员自己解析吗？

- 不需要，web服务器负责对HTTP请求数据进行解析，并封装为了请求对象

2．如何获取请求数据？

- HttpServletRequest对象里面封装了所有的请求信息

- **5 HTTP协议-响应数据协议**

```
HTTP/1.1 200 OK
Content-Type: application/json
Transfer-Encoding: chunked
Date: Tue, 10 May 2022 07:51:07 GMT
Keep-Alive: timeout=60
Connection: keep-alive

[{id: 1, brandName: "阿里巴巴",companyName: "腾讯计算机系统有限公司", description: "玩玩玩"}]
```

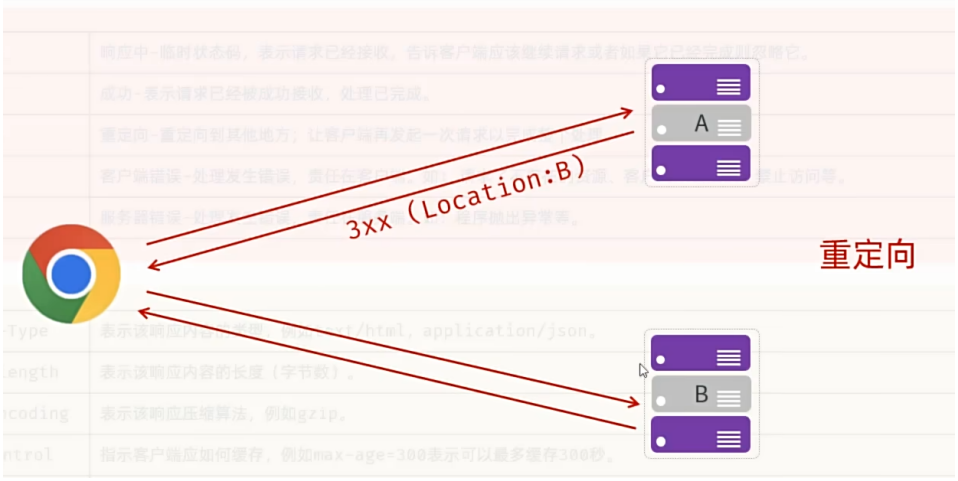
响应行：响应数据第一行(协议、状态码、描述)

响应头：第二行开始，格式key: value

响应体：最后一部分，存放响应数据

1xx	响应中-临时状态码，表示请求已经接收，告诉客户端应该继续请求或者如果它已经完成则忽略它。
2xx	成功-表示请求已经被成功接收，处理已完成。
3xx	重定向-重定向到其他地方；让客户端再发起一次请求以完成整个处理。
4xx	客户端错误-处理发生错误，责任在客户端。如：请求了不存在的资源、客户端未被授权、禁止访问等。
5xx	服务器错误-处理发生错误，责任在服务端。如：程序抛出异常等。

Content-Type	表示该响应内容的类型，例如text/html, application/json。
Content-Length	表示该响应内容的长度（字节数）。
Content-Encoding	表示该响应压缩算法，例如gzip。
Cache-Control	指示客户端应如何缓存，例如max-age=300表示可以最多缓存300秒。
Set-Cookie	告诉浏览器为当前页面所在的域设置cookie。



状态码	描述
200	客户端请求成功。
404	请求资源不存在，URL输入有误，或者网站资源被删除了。
500	服务器发生不可预期的错误。

- **6 响应数据设置**
  - Web服务器对HTTP协议的响应数据进行了封装(HttpServletResponse)，并在调用Controller方法的时候传递给了该方法。这样，就使得程序员不必直接对协议进行操作，让Web开发更加便捷。

方式一：基于HttpServletResponse封装

```
@RequestMapping("/response")
public void response(HttpServletResponse response) throws IOException {
    // 1.设置响应状态码
    response.setStatus(401);
    // 2.设置响应头
    response.setHeader("itheima","itheima");
    // 3.设置响应体
    response.getWriter().write("<h1>Hello Response</h1>");
}
```

方式二：基于ResponseEntity封装

```
@RequestMapping("/response")
public ResponseEntity<String> response() {
    return ResponseEntity.status(401)          // 1.设置响应状态码
            .header("group", "itcast")         // 2.设置响应头
            .body("<h1>Hello Response</h1>"); // 3.设置响应体
}
```

⚠ 注意： 响应状态码 和 响应头如果没有特殊要求的话，通常不手动设定。服务器会根据请求处理的逻辑，自动设置响应状态码和响应头。

1. HTTP响应数据需要程序员自己手动设置吗？
    - 不需要
    - Web服务器对HTTP响应数据进行了封装(HttpServletResponse)
2. 响应状态码、响应头需要我们手动指定吗？
    - 通常情况下，我们无需手动制定，服务器会根据请求逻辑自动设置