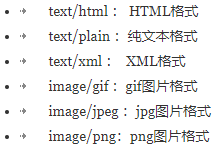
# 关于http请求头（Accpet）和实体头（Content-Type）

## 常见的媒体类型：

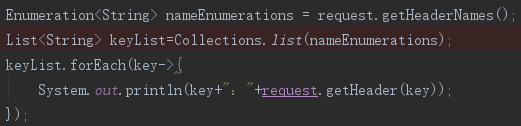


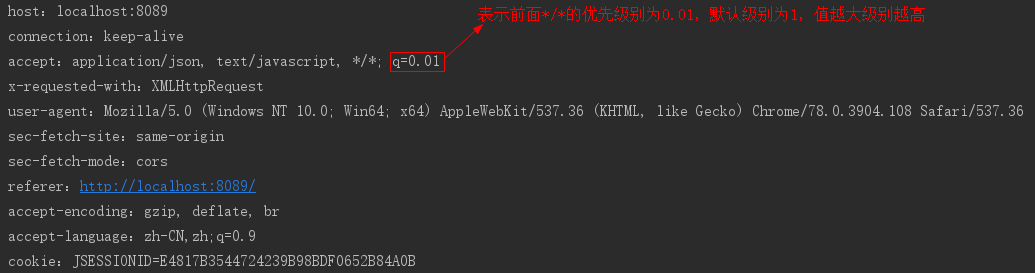
以application开头的媒体类型格式：

* application/xhtml+xml ：XHTML格式
* application/xml     ： XML数据格式
* application/atom+xml  ：Atom XML聚合格式
* application/json    ： JSON数据格式
* application/pdf       ：pdf格式
* application/msword  ： Word文档格式
* application/octet-stream ： 二进制流数据（如常见的文件下载）
* application/x-www-form-urlencoded ： <form encType=””>中默认的encType，form表单数据被编码为key/value格式发送到服务器（表单默认的提交数据的格式）

## Accpet

Accpet：表示的是请求头，描述客户端允许接收的数据  
下面是一个javaweb中获取headers的例子：



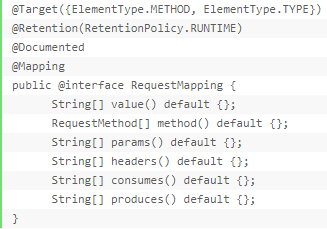


## Content-Type

Content-Type：表示的是实体头，描述客户端发送的body数据类型（即实体部分）

## 在SpringMvc中的运用

### @RequestMapping注解：

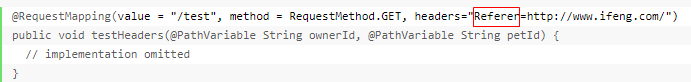


value:  指定请求的实际地址， 比如 /action/info之类。  
method：  指定请求的method类型， GET、POST、PUT、DELETE等  
consumes： 指定处理请求的提交内容类型（Content-Type），例如application/json, text/html;  
produces:    指定返回的内容类型，仅当request请求头中的(Accept)类型中包含该指定类型才返回  
params： 指定request中必须包含某些参数值是，才让该方法处理  
headers： 指定request中必须包含某些指定的header值，才能让该方法处理请求

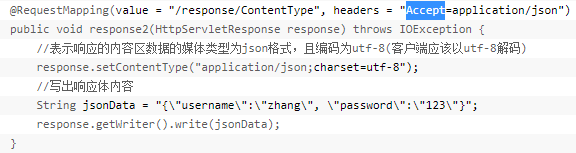
其中，consumes， produces使用content-typ信息进行过滤信息；headers中可以使用content-type进行过滤和判断。

#### 例子：

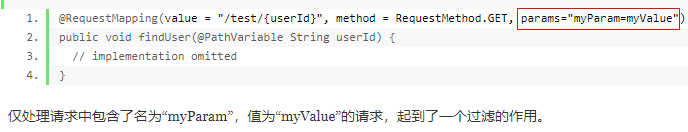
1. 使用referer：



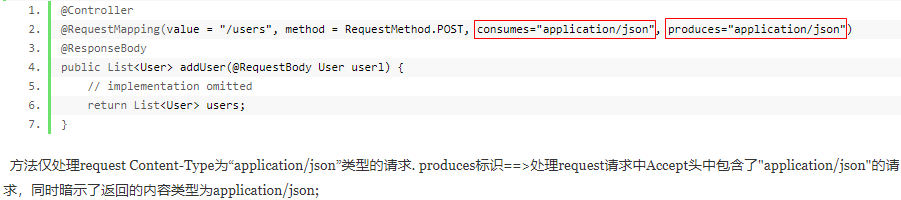
1. Accpet：



1. params（下面展示一个使用params的过滤例子）：



1. consumes/produces：



### 在Springmvc中是如何确定将服务端返回的数据类型转为哪种格式

1. **如何确定使用返回哪种dataType**

<https://blog.csdn.net/Hmily_hui/article/details/82994868>

<http://www.mytju.com/classcode/news_readNews.asp?newsID=959>

1. **消息转换器**

<https://blog.csdn.net/shicuicui1234/article/details/78963817>

1. **@RequestBody注解**

<https://cloud.tencent.com/developer/article/1414464>

### Requests和Response部分的参数介绍：

<https://www.cnblogs.com/111testing/p/6037579.html>

## 请求头中的权重系数（q）：

q 是权重系数，范围 0 =< q <= 1，q 值越大，请求越倾向于获得其“;”之前的类型表示的内容，若没有指定 q 值，则默认为1，若被赋值为0，则用于提醒服务器哪些是浏览器不接受的内容类型。

例子中Accept:text/javascript, \*/\* ,q=0.01 q的作用是代表\*/\*的权重最小,由于默认值小于1大于0,?如果设置为0,那么就代表只接受:text/javascript这样一种。

例子 Accept-Language: da, en-gb;q=0.8, en;q=0.7  将意味着：“我更喜欢丹麦语，但会接受英式英语和其他类型的英语。”

## X-Requested-With



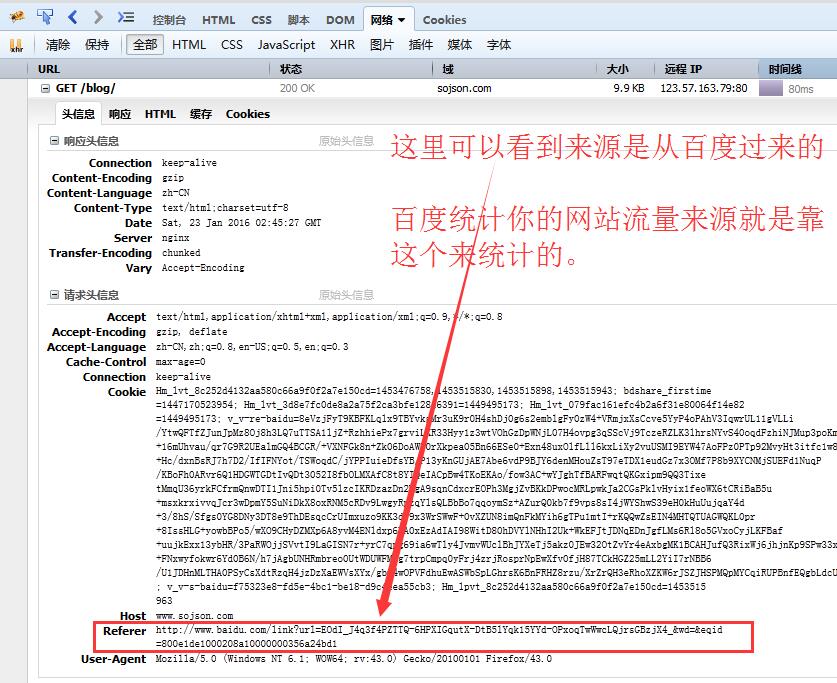
如果 requestedWith 为 null，则为同步请求。如果 requestedWith 为 XMLHttpRequest 则为 Ajax 请求。

<https://blog.csdn.net/heatdeath/article/details/79168614>

## Referer

### 什么是referer

[Referer](https://www.sojson.com/tag_referer.html)是 [HTTP](https://www.sojson.com/tag_http.html)请求header 的一部分，当浏览器（或者模拟浏览器行为）向web 服务器发送请求的时候，头信息里有包含 [Referer](https://www.sojson.com/tag_referer.html)。比如我在www.sojson.com 里有一个www.baidu.com 链接，那么点击这个www.baidu.com ，它的header 信息里就有：Referer=https://www.sojson.com。由此可以看出来吧。它就是表示一个来源。看下图的一个请求的[Referer](https://www.sojson.com/tag_referer.html)信息。



### 这里有一个小问题要说明下。

[Referer](https://www.sojson.com/tag_referer.html)的正确英语拼法是referrer 。由于早期HTTP规范的拼写错误，为了保持向后兼容就将错就错了。其它网络技术的规范企图修正此问题，使用正确拼法，所以目前拼法不统一。还有它第一个字母是大写。

### Referer的作用？

**1、防盗链。**

刚刚前面有提到一个小[Demo](https://www.sojson.com/tag_demo.html)。

我在www.sojson.com里有一个www.baidu.com链接，那么点击这个www.baidu.com，它的header信息里就有：

Referer=https://www.sojson.com

那么可以利用这个来防止盗链了，比如我只允许我自己的网站访问我自己的图片服务器，那我的域名是www.sojson.com，那么图片服务器每次取到Referer来判断一下是不是我自己的域名www.sojson.com，如果是就继续访问，不是就拦截。

这是不是就达到防盗链的效果了？

**2、防止恶意请求。**

比如我的SOJSON网站上，静态请求是\*.html结尾的，动态请求是\*.shtml，那么由此可以这么用，所有的\*.shtml请求，必须[Referer](https://www.sojson.com/tag_referer.html)为我自己的网站。

Referer=https://www.sojson.com

### 空Referer是怎么回事？什么情况下会出现Referer?

首先，我们对空[Referer](https://www.sojson.com/tag_referer.html)的定义为，[Referer](https://www.sojson.com/tag_referer.html)头部的内容为空，或者，一个[HTTP](https://www.sojson.com/tag_http.html)请求中根本不包含[Referer](https://www.sojson.com/tag_referer.html)头部。

那么什么时候[HTTP](https://www.sojson.com/tag_http.html)请求会不包含[Referer](https://www.sojson.com/tag_referer.html)字段呢？根据Referer的定义，它的作用是指示一个请求是从哪里链接过来，那么当一个请求并不是由链接触发产生的，那么自然也就不需要指定这个请求的链接来源。

比如，直接在浏览器的地址栏中输入一个资源的URL地址，那么这种请求是不会包含[Referer](https://www.sojson.com/tag_referer.html)字段的，因为这是一个“凭空产生”的[HTTP](https://www.sojson.com/tag_http.html)请求，并不是从一个地方链接过去的。

参考：<https://www.sojson.com/blog/58.html>