RESTful是一种架构设计风格，提供了设计原则和约束条件，而不是架构，而满足这些约束条件和原则的应用程序或设计就是 Restful架构或服务，RESTful的核心就是后端将资源发布为URI，前端通过URI访问资源，并通过HTTP动词表示要对资源进行的操作。

1. URL设计

1.1动词+宾语

动词通常就是五种 HTTP方法，对应CRUD操作。

* GET：读取（Read）
* POST：新建（Create）
* PUT：更新（Update）
* PATCH：更新（Update），通常是部分更新
* DELETE：删除（Delete）  
  1.2 动词覆盖

有些客户端只能get，post，服务器可以接受post模拟其他方式

POST /api/Person/4 HTTP/1.1

X-HTTP-Method-Override: PUT

1.3 宾语为名词  
 宾语就是API的URL，是HTTP动词作用的对象。它应该是名词，不能是动词。

1.4 复数URL

没有统一的规定，但是常见的操作是读取一个集合，就使用复数

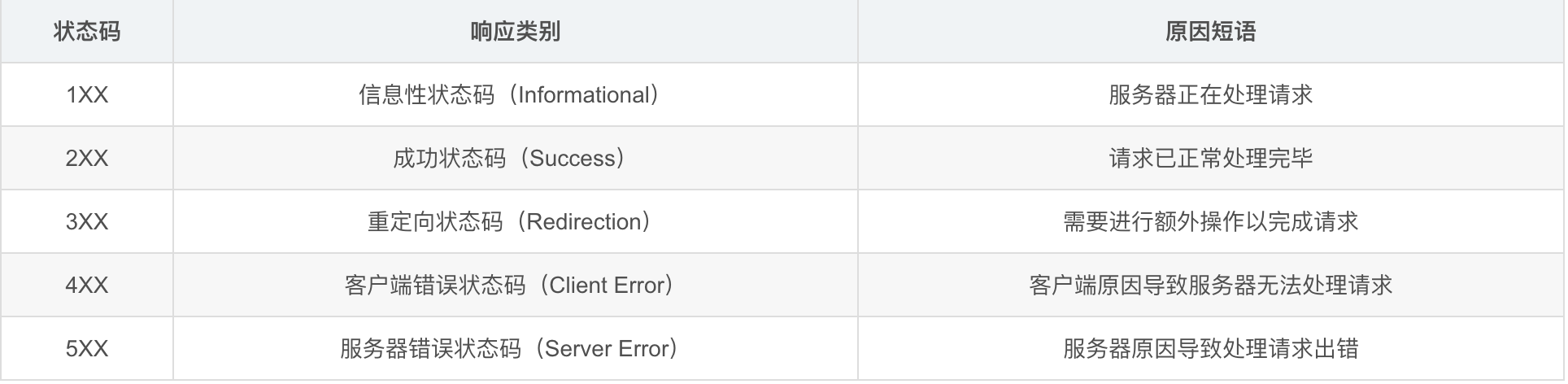
1.5 避免多级URL

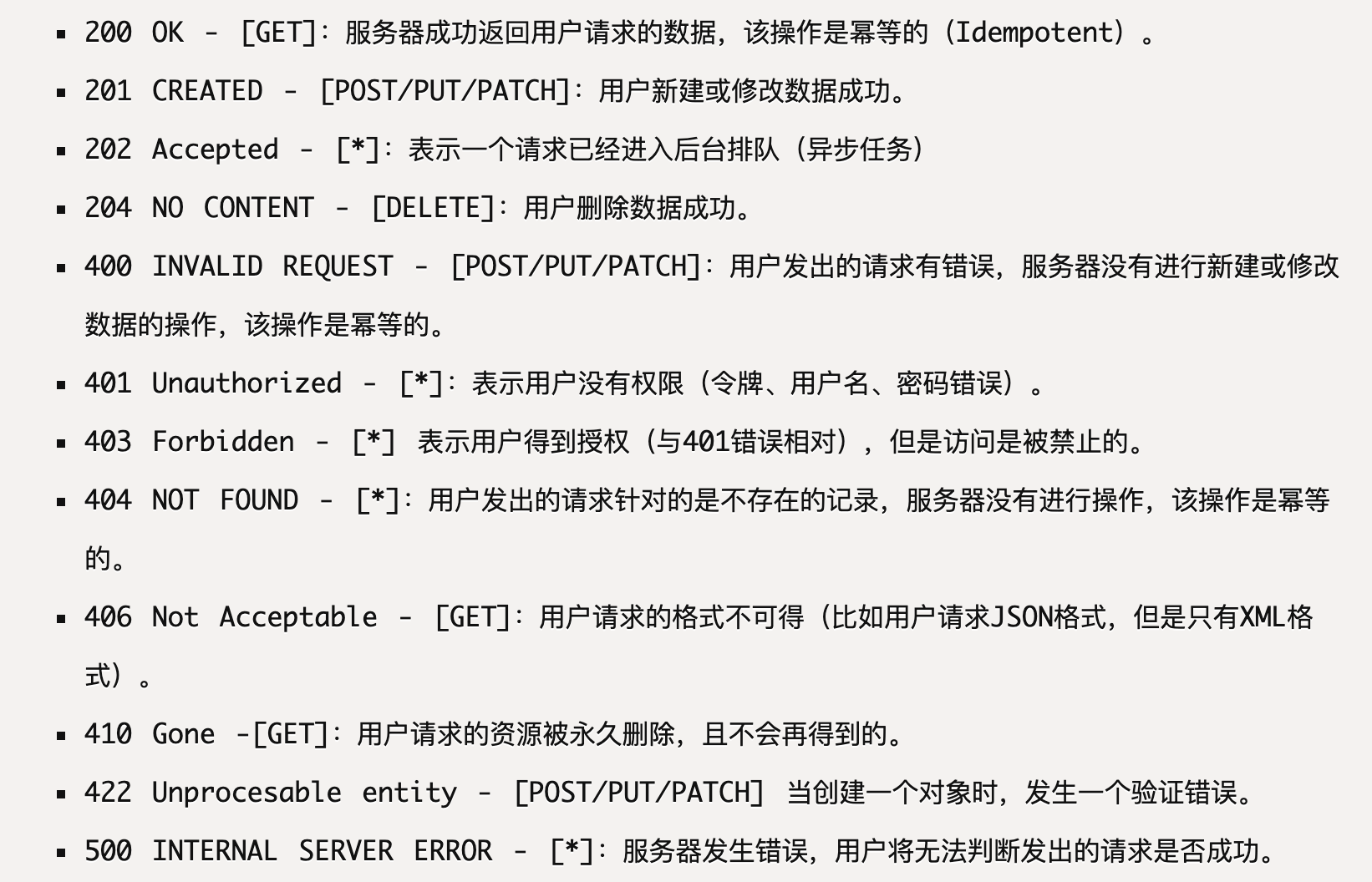
资源需要多级分类，因此很容易写出多级的 URL。可以选择查询字符串的写法。

GET /articles?published=true

1. 状态码

2.1 状态码必须精确





1. 服务器回应

3.1 不以文本形式返回

API 返回的数据格式，不应该是纯文本，而应该是一个 JSON 对象，因为这样才能返回标准的结构化数据。所以，服务器回应的 HTTP 头的Content-Type属性要设为application/json。

3.2 发生错误时，不要返回200

不正确做法：

HTTP/1.1 200 OK

Content-Type: application/json

{

"status": "failure",

"data": {

"error": "Expected at least two items in list."

}

}

3.3 提供链接

API的使用者未必知道，URL是怎么设计的。一个解决方法就是，在回应中，给出相关链接，便于下一步操作。这样的话，用户只要记住一个URL，就可以发现其他的URL。这种方法叫做 HATEOAS。