



链滴

# RESTful API

作者: [helly](#)

原文链接: <https://ld246.com/article/1499071177511>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p><strong>一、有关 RESTful API 的一些术语</strong><br>  
<strong>资源</strong>：一个对象的单独实例。<br>  
<strong>集合</strong>：一群同种对象。<br>  
<strong>端点</strong>：这个 API 在服务器上的 URL 用于表达一个资源或者一个集合。<br>  
<strong>幂等</strong>：无边际效应，多次操作得到相同的结果。<br>  
<strong>URL 段</strong>：在 URL 里面已斜杠分隔的内容。</p>  
<p><strong>二、版本化 RESTful API</strong><br>  
将 API 部署到二级域名下并带上版本号：<br>  
<a href="https://ld246.com/forward?goto=https%3A%2F%2Fapi.example.com%2Fv1" target  
"\_blank" rel="nofollow ugc">https://api.example.com/v1</a></p>  
<p><strong>三、RESTful API 用到的 HTTP 请求方法</strong><br>  
<strong>GET (选择)</strong>：从服务器上获取一个具体的资源或者一个资源列表。<br>  
<strong>POST (创建)</strong>：在服务器上创建一个新的资源。<br>  
<strong>PUT (更新)</strong>：以整体的方式更新服务器上的一个资源。<br>  
<strong>PATCH (更新)</strong>：只更新服务器上一个资源的一个属性。<br>  
<strong>DELETE (删除)</strong>：删除服务器上的一个资源。</p>  
<p><strong>四、端点</strong><br>  
<strong>GET /zoos</strong>：List all Zoos.<br>  
<strong>POST /zoos</strong>：Create a new Zoo.<br>  
<strong>GET /zoos/ZID</strong>：Retrieve an entire Zoo object.<br>  
<strong>PUT /zoos/ZID</strong>：Update a Zoo (entire object).<br>  
<strong>PATCH /zoos/ZID</strong>：Update a Zoo (partial object).<br>  
<strong>DELETE /zoos/ZID</strong>：Delete a Zoo.</p>  
<p><strong>五、过滤器、排序、分页</strong><br>  
<strong>?limit=10</strong>：分页时限制返回的资源数量。<br>  
<strong>?offset=10</strong>：分页时限制资源查找起点。<br>  
<strong>?topic=soil</strong>：使用条件匹配来过滤记录，条件字段是数据库表的字段。<br>  
\*\*?sortBy=name&order=asc\*\*：对结果按特定属性进行排序。</p>  
<p><strong>限制 API 返回值的域</strong><br>  
GET /ticketsfields=id,subject,customer\_name,updated\_at&state=open&sort=-update\_at</p>  
<p>不符合 CRUD 的操作</p>  
<ol>  
<li>重构你的行为 action。当你的行为不需要参数的时候，你可以把 active 对应到 activated 这个源，（更新使用 patch）.</li>  
<li>以子资源对待。例如:github 上，对一个 gists 加星操作：PUT /gists/:id/star 并且取消星操作：DELETE /gists/:id/star.</li>  
<li>有时候 action 实在难以和某个资源对应上例如 search。那就这么办吧。</li>  
</ol>  
<p><strong>六、状态码</strong><br>  
<strong>200 OK – [GET]</strong>：服务器成功找到客户端请求的数据。<br>  
<strong>201 CREATED – [POST/PUT/PATCH]</strong>：服务器创建资源成功。<br>  
<strong>204 NO CONTENT – [DELETE]</strong>：服务器删除资源成功。<br>  
<strong>400 INVALID REQUEST – [POST/PUT/PATCH]</strong>：客户端提供的数据不正确，服务器没有执行操作。<br>  
<strong>404 NOT FOUND – [GET/POST/PUT/PATCH/DELETE]</strong>：客户端引用了一个存在的资源或集合，服务器没有执行操作。<br>  
<strong>500 INTERNAL SERVER ERROR – [GET/POST/PUT/PATCH/DELETE]</strong>：服务发生内部错误，客户端无法得知结果，即便请求已经处理成功。</p>  
<p><strong>七、返回文档</strong><br>  
<strong>GET /collection</strong>：返回一系列资源对象。<br>  
<strong>GET /collection/resource</strong>：返回单独的资源对象。<br>  
<strong>POST /collection</strong>：返回新创建的资源对象。<br>  
<strong>PUT /collection/resource</strong>：返回完整的资源对象。<br>

**PATCH /collection/resource** : 返回完整的资源对象。 <br>

**DELETE /collection/resource** : 返回一个空文档。 </p>

<p>八、内容类型<br>

Host: api.example.org<br>

Accept: application/json<br>

Content-Type: application/json<br>

Content-Length: 24</p>

<p><strong>九、HTTP 封包</strong><br>

{<br>

status : {<br>

code : code,<br>

msg : msg<br>

}, <br>

data : {<br>

...<br>

}<br>

}</p>

<p>文档</p>

<p>命名方式<br>

是蛇形命令（下划线和小写）还是驼峰命名？如果使用 json 那么最好的应该是遵守 JAVASCRIPT 的命名方法-也就是说骆驼命名法。如果你正在使用多种语言写一个库，那么最好按照那些语言所推荐的，java, c#使用骆驼，python, ruby 使用 snake。 </p>

<p><strong>参考资料</strong>: <br>

<a href="https://ld246.com/forward?goto=https%3A%2F%2Fgithub.com%2Faisuhua%2Frestful-api-design-references" target="\_blank" rel="nofollow ugc">https://github.com/aisuhua/restful-api-design-references</a> </p>