

喵先生 的进阶之路

简单 快乐 喵

浅析Patch和Put的区别

📅 2018-06-09

在写后台更新接口的时候，定义路由使用 `@PutMapping`，这个注解，但是，后来又改成的 `@PatchMapping`，所以就查了下两者的区别。

问题描述

原来在进行更新操作的时候，都是使用 `put` 方法来进行请求的，但是这次在写更新的时候，只需要更新一个字段，所以使用了 `patch` 方法，那么，`patch` 和 `put` 这两者有什么区别呢？

局部与整体

最基本的，`patch` 是对 `put` 的一个补充。举个例子：

一个计量单位实体 `MeasurementUnit`，他有 `name`，`weight`，`measurementUnitCategory` 等多个字段，在这里，我们要只修改 `weight` 这个字段，这时应该如何选择呢？

通常，我们为了省事，就会直接将修改了 `weight` 的完整的 `MeasurementUnit` 对象直接传给后台。但是这种做法实际上并不明智，这会浪费大量的网络带宽。

但是 `patch` 呢，他会只将 `weight` 传到制定资源去，表示这是一个局部更新，后端只更新收到的字段。

幂等性

另一个方面的区别：`put` 是幂等的，而 `patch` 是非幂等的。

这里有一个比较有意思的概念，叫做 幂等性。什么是 幂等性 呢，就是在请求某一资源的时候，多次操作与一次操作对被操作的对象达到的效果是一样的。简单举个例子：

用上面的更新来举例子，将权重 `weight` 更新为1，这种操作应该是执行多次都是达到同一个效果。

另一个操作，将 weight 更新为 +1 ，这种操作显然多次操作就和一次操作不同了。

友情链接：

<https://segmentfault.com/q/1010000011686226>

<https://segmentfault.com/q/1010000005685904>


坚持原创技术分享，您的支持将鼓励我继续创作！

Donate

[# java](#) [# http](#)

[◀ PostMapping处理参数”/“](#)

[继承遗留问题 ▶](#)

© 2017 – 2018  朴世超

Erstellt mit [Hexo](#) | Theme – [NexT.Gemini v5.1.3](#)