链滴

REST API URI 的七大设计原则

作者：w8854123

原文链接：https://ld246.com/article/1498612412089

来源网站：链滴

原文：http://blog.restcase.com/7-rules-for-rest-api-uri-design/

作者：Guy Levin

翻译：Aladdin

在了解REST API URI设计的规则之前，让我们快速浏览一些我们将要讨论的术语。

URIs

REST API使用统一资源标识符（URI）来寻址资源。在当今互联网上，充斥着各种各样的URI设计规，既有像//api.example.com/louvre/leonardo-da-vinci/mona-lisa这样能够清楚的传达API资源模的文章，也有很难理解的文章，例如：//api.example.com/68dd0-a9d3-11e0-9f1c-0800200c9a66；Tim Berners-Lee在他的“Axioms of Web Architecture”一文中将URI的不透明度总结成一句话：

唯一可以使用标识符的是引用对象。在不取消引用时，就不应该查看URI字符串的内容以获取其他信。

——蒂姆·伯纳斯 - 李

客户端必须遵循Web的链接范例，将URI视为不透明标识符。

REST API设计人员应该在考虑将REST API资源模型传达给潜在的客户端开发者的前提下，创造URI。这篇文章中，我将尝试为REST API URI 引入一套设计规则。

先跳过规则，URI的通用语法也适用与本文中的URI。RFC 3986定义了通用URI语法，如下所示：

URI = scheme “://” authority “/” path [“?” query][“#” fragment]

规则1：URI结尾不应包含（/）

这是作为URI路径中处理中最重要的规则之一，正斜杠（/）不会增加语义值，且可能导致混淆。REST API不允许一个尾部的斜杠，不应该将它们包含在提供给客户端的链接的结尾处。

许多Web组件和框架将平等对待以下两个URI：

http://api.canvas.com/shapes/

http://api.canvas.com/shapes

但是，实际上URI中的每个字符都会计入资源的唯一身份的识别中。

两个不同的URI映射到两个不同的资源。如果URI不同，那么资源也是如此，反之亦然。因此，REST AI必须生成和传递精确的URI，不能容忍任何的客户端尝试不精确的资源定位。

有些API碰到这种情况，可能设计为让客户端重定向到相应没有尾斜杠的URI（也有可能会返回301 - 来资源重定向）。

规则2：正斜杠分隔符（/）必须用来指示层级关系

URI的路径中的正斜杠（/）字符用于指示资源之间的层次关系。

例如：

http://api.canvas.com/shapes/polygons/quadrilaterals/squares;

规则3：应使用连字符（ - ）来提高URI的可读性

为了使您的URI容易让人们理解，请使用连字符（ - ）字符来提高长路径中名称的可读性。在路径中应该使用连字符代空格连接两个单词 。

例如：

http://api.example.com/blogs/guy-levin/posts/this-is-my-first-post

规则4：不得在URI中使用下划线（_）

一些文本查看器为了区分强调URI，常常会在URI下加上下划线。这样下划线（_）字符可能被文本查器中默认的下划线部分地遮蔽或完全隐藏。

为避免这种混淆，请使用连字符（ - ）而不是下划线

规则5：URI路径中首选小写字母

方便时，URI路径中首选小写字母，因为大写字母有时会导致一些问题。RFC 3986将URI定义为区分小写，但scheme 和 host components除外。

例如：

http://api.example.com/my-folder/my-doc

HTTP://API.EXAMPLE.COM/my-folder/my-doc

这个URI很好。URI格式规范（RFC 3986）认为该URI与URI＃1相同。

http://api.example.com/My-Folder/my-doc

而这个URI与URI 1和2不同，这可能会导致不必要的混淆。

规则6：文件扩展名不应包含在URI中

在Web上，（.）字符通常用于分隔URI的文件名和扩展名。

REST API不应在URI中包含人造文件扩展名，来指示邮件实体的格式。相反，他们应该依赖通过Content-Type中的header传递media type，来确定如何处理正文的内容。

http://api.college.com/students/3248234/courses/2005/fall.json

http://api.college.com/students/3248234/courses/2005/fall

如上所示：不应使用文件扩展名来表示格式。

应鼓励REST API客户端使用HTTP提供的格式选择机制Accept request header。

为了是链接和调试更简单，REST API应该支持通过查询参数来支持媒体类型的选择。

规则7：端点名称是单数还是复数?

keep-it-simple的原则在这里同样适用。虽然一些”语法学家”会告诉你使用复数来描述资源的单个例是错误的，但实际上为了保持URI格式的一致性建议使用复数形式。

本着API提供商更容易实施和API使用者更容易操作的原则，可以不必纠结一些奇怪的复数（person/people，goose/geese）。

但是应该怎么处理层级关系呢？如果一个关系只能存在于另一个资源中，RESTful原则就会提供有用指导。我们来看一下这个例子。学生有一些课程。这些课程在逻辑上映射到学生终端，如下所示：

http://api.college.com/students/3248234/courses - 检索id为3248234的学生学习的所有课程的单。

http://api.college.com/students/3248234/courses/physics -检索该学生的物理课程

结论

当你在设计REST API服务时，您必须注意这些由URI定义的资源。

正在构建的服务中的每个资源将至少有一个URI标识它。这个URI最好是有意义的，且能充分描述资。URI应遵循可预测的层次结构，用来提高其可理解性，可用性：可预测的意义在于它们是一致的，的层次结构在数据关系上是有意义的。

RESTful API是为使用者编写的。URI的名称和结构应该能够向使用者传达更清晰的含义。通过遵循上规则，您将创建一个更清晰的的REST API与更友好的客户端。这些并不是REST的规则或约束，仅仅是PI的增强和补充。

我也建议你来看看http://blog.restcase.com/5-basic-rest-api-design-guidelines/这篇文章。

最后，望大家牢记：你在为你的客户端设计API URI，而不仅仅是为你的数据。