

上节课我们通过向/api/posts/21这个路径发送了一个get请求, 获取到了响应的内容。响应内容的格式是application/ld+json格式。这是一种特殊的json格式, 我们只看响应内容后面几行的话, 它就是一个普通的json格式。

但是在普通的json格式上面, 又添加了@context, @id和@type这几项, 另外就是author属性。它并没有获取到文章作者的详细信息, 而是返回一个字符串, 我这里找到了知乎上另一篇关于RESTful API的文章。

这篇文章对RESTful的架构进行了更明了的解释, 每个URI都代表一种资源。客户端和服务端之间通过某种形式来传递这种资源, 比如json、xml、image或者txt等等。html也是一种表现形式。第3条客户端通过特定的HTTP动词, 就是HTTP方法, 对服务器端的资源进行操作, 来实现表现层的状态转换。

回到API文档页面, 往上看, 现在每一条URI都代表了一种资源。当我们使用特定的HTTP方法访问URI时, 就会对资源进行操作, 资源的状态就会进行转换, 然后以某种表现形式, 比如说json或者xml等等格式返回到客户端。这就是RESTful架构规范。

我们继续往下看, RESTful的极致就是HATEOAS, 实际上RESTful API最好做到Hypermedia, Hypermedia就是在返回的结果中提供链接, 链接向其他的API方法。使得用户不查找文档, 也知道下一步应该做什么, 我们再看RedHat网站上对于RESTful API的一些介绍。

组件间的统一接口, 使信息以标准形式传输, 我们已经创建了相对统一的接口。所请求的资源可识别, 并与发送给客户端的表述分离开。客户端通过接收的表述操作资源, 因为表述中包含操作所需要的充足信息, 第3条返回给客户端的自描述信息包含充足的信息, 能够指明客户端应该如何处理所接收的信息。第4条超文本超媒体可用, 只在访问资源后, 客户端能够使用超链接查找当前可用的所有其他操作, 也就是说我们的响应内容中应该包含客户端继续操作资源的所有信息。

比如说我们在响应内容中添加一个URI, 然后客户端可以使用这个URI来对当前的资源进行下一步的操作, 其实我们的html代码就是这个标准的一种实现。客户端在获取到html代码之后, 可以通过页面上的超链接对页面的资源进行下一步操作。

回到文档页面, 我们查看响应的结果。在响应内容前三行, 第一条是@context, @context就是对当前数据的一种描述。我们已经获取到了文章的数据, 如果我们不看前三行, 我们知道id就代表的是文章的id, title就代表的是文章的标题, author代表文章的作者。但是同样的属性也可以代表其他的内容。title属性也可能代表的是其他的意思, 并不一定是文章的标题。这里@context属性指向了一个链接, 这个链接就对文章的各个属性进行了描述, 我们复制链接地址访问一下。

打开新的标签, 输入127.0.0.1:8000/, 粘贴。在响应的内容中, 对响应的文章数据进行了描述。这里描述相应的数据中, title的值它指向的是文章类的title属性。然后我们看第一项@vocab, 我们访问这个地址。

我们看到当前的地址是docs.jsonld, 我们当前的内容格式就是.jsonld格式, .jsonld格式就是Hypermedia的一种实现。回到页面我们往下看, 这里有对Post类的一些描述。这里描述的类中的属性, 比如说title属性, 它就是string类型的字符串。继续往下看, summary也是string类型的字符串, 有了这些对Post类各个属性的描述。我们就可以知道API响应的内容, 各个属性所代表的意义。

我们再来看第2条, @id, 这个@id和我们的id属性很像, 但是它们所代表的意义却完全不同。@id又叫做国际化资源标识符, 简称叫做IRI。IRI地址, 它指向了全网唯一的一个资源, 这个id属性它是数据库中当前文章的主键id。当我们访问@id这个路径时, 它就指向了全网唯一的一个资源。资源链接到的数据就是我们的当前id为21的文章。

我们复制这个地址, 打开新的标签页, 输入127.0.0.1:8000, 粘贴。后面我们加上.jsonld。我们就获取到了全网唯一的一条数据, 再次回到文档页面。

@type它描述的是当前数据的类型是Post类型，在作者这里。它指向的也是一个IRI，我们可以通过访问这个地址来获取当前的作者信息，jsonId就是Hypermedia的一个表现形式。这样我们通过API Platform创建的API都实现了REST规范。

我们刷新文档页面，文档中还生成了User类的API接口，但是这些接口中有一些API我们是不需要的。在下节课我们对API的操作进行设置，可以删除某些API。