

返回完整的URLs

通常，从Web API返回绝对完整的URI（如 `http://example.com/foobar`），比起返回相对路径的URI（如 `/foobar`），可能要更好一点。

这样做的好处有：

- 更为明确。
- 减少客户端的工作量
- 当在没有本机URI类型的表示（如JSON）中找到字符串时，它的含义没有任何歧义
- 使用超链接标记HTML变得容易

REST框架提供了两个实用函数，让你从Web API返回绝对URIs更加简单，也就是 `reverse` 和 `reverse_lazy` 方法。

DRF不强制你必须使用它们，但是如果用了，自我描述的API将能够自动为你链接其输出，使得浏览API更加容易。

reverse

签名: `reverse(viewname, *args, **kwargs)`

`reverse`就是解析或生成url的意思，其行为和Django原生的 `django.urls.reverse` 基本一样，除了它使用request中的信息确定主机和端口信息，然后返回的一个完全限定的URL。

此时，你应该将request作为关键字参数传递给reverse方法，例如：

```
1  from rest_framework.reverse import reverse
2  from rest_framework.views import APIView
3  from django.utils.timezone import now
4
5  class APIRootView(APIView):
6      def get(self, request):
7          year = now().year
8          data = {
9              ...
10             'year-summary-url': reverse('year-summary', args=[year],
11             request=request) #看这里
12         }
13         return Response(data)
```

reverse_lazy

签名: `reverse_lazy(viewname, *args, **kwargs)`

和Django原生的 `django.urls.reverse_lazy` 方法基本一样, 除了它使用request中的信息确定主机和端口信息, 然后返回的一个完全限定的URL。

和 `reverse` 一样, 也要将request作为关键字参数传递给方法, 例如:

```
1 api_root = reverse_lazy('api-root', request=request)
```