

[第3步] 编写views

参考资料及教程

<https://docs.djangoproject.com/en/1.9/intro/tutorial03/>

目标

1. 在urls的目录下，找到views.py，参考 tutorial03 的编写两个views:
 - a. 一个叫 encode，负责在POST里读出需要encode的url，然后encode为一个short url
 - i. 可以直接return HttpResponseRedirect(short_url)作为返回结果

```
def encode(request)
    # get long url from request.POST
    ...
    # convert to short url
    ...
    # create an object in database
    ...
    return HttpResponseRedirect(short_url)
```

- b. 一个叫做 decode，负责把短网址从url里拿short url出来，然后还原为对应的long url
 - i. 还原之后，可以重定向到对应的longurl
 - ii. <https://docs.djangoproject.com/en/1.9/ref/request-response/#django.http.HttpResponseRedirect>

```
def decode(request, short_url):
    # get long url from db
    ...
    # if does not exist, return error info
    return HttpResponse("error")
    # redirect to long url
    return HttpResponseRedirect(long_url)
```

- c. 如何进行数据库查询操作？
 - i. <https://docs.djangoproject.com/en/1.9/topics/db/queries/>
 - ii. 主要使用的方法是 Url.objects.get 和 Url.objects.filter

d. 如何创建一条新的数据在数据库里？

```
# 方法2
url = Url.objects.create(
    short=short,
    long=long
)# 方法1
url = Url()
url.short = xxx
url.long = xxx
url.save()

# 方法2
url = Url.objects.create(
    short=short,
    long=long
)
```

2. 在urls.py（router文件）里编写对应的规则，使得：
 - a. /<shorturl> 指向到 decode 的views
 - b. /data/shorten/ 指向到 encode 的views
3. 完成之后，将代码上传到你的github上，并把地址粘贴在下面：
 - a. (粘贴你的github地址)

完成的同学请签到

QA

Q:如何使用github?

A:网上资料很多，比如官方教程：<https://guides.github.com/activities/hello-world/>

其他问题请留言