**url\_for使用详解**

一般我们通过一个 URL 就可以执行到某一个函数。如果反过来，我们知道一个函数，怎么去获得

这个 URL 呢？ url\_for 函数就可以帮我们实现这个功能。

**url\_for()**  ：函数接收两个及以上的参数，他接收函数名作为第一个参数，接收对应URL规则的命名参数，如果还出现其他的参数，则会添加到 URL 的后面作为查询参数。

如：

@app.route(**'/post/list/<page>/'**)

**def** my\_list(page):

**return 'my list'**

@app.route(**'/'**)

**def** hello\_world():

# 构建出来的url：/post/my\_list/2?num=8

**return** url\_for(**'my\_list'**,page=2,num=8)

# return "/post/list/2？num=8"

**为什么选择`url\_for` 而不选择直接在代码中拼 URL 的原因有两点：**

1. 将来如果修改了 URL ，但没有修改该 URL 对应的函数名，就不用到处去替换 URL 了。

2. url\_for() 函数会转义一些特殊字符和 unicode 字符串，这些事情 url\_for 会自动的帮我们

搞定。

如：

@app.route(**'/login/'**)

**def** login():

**return 'login'**

@app.route(**'/'**)

**def** hello\_world():

**return** url\_for(**'login'**, next=**'/'**)

# /login/?next=/

# 会自动的将/编码，不需要手动去处理。

# url=/login/?next=%2F