链滴

django path

作者：alaikis

原文链接：https://ld246.com/article/1519216866851

来源网站：链滴

2017年12月2号，Django2.0发布！！！所以之前1.8版本的已经有所改变。

其中url变成path。

9月23日Django发布了2.0a1版本，这是一个 feature freeze 版本，如果没有什么意外的话，2.0正版不会再增加新的功能了。按照以往的规律，预计正式版将在12月发布。

2.0无疑是一个里程碑版本，因为这是第一个只支持Python3.X的版本，和1.x是不兼容的。

What’ s new in Django2.0 文档中一共列出了三个新的特性：

- 更简单的URL路由语法 (Simplified URL routing syntax)
- admin应用的针对移动设备的优化改进(Mobile-friendly contrib.admin)
- 支持SQL开窗表达式(Window expressions)

第一个特性，主要用于动态路由定义上。在Django2.0代码实现中，主要的变化是新增了 django.urls path 函数，它允许使用一种更加简洁、可读的路由语法。比如之前的版本的代码：

[python] view plain copy

1.

2. url(r'^articles/(?P[0-9]{4})//span>, views.year_archive),

新语法支持类型转化，在上述的例子中， year_archive函数接收到的year参数就变成整数而不是字符。在新版本中也可以写为：

如果你有接触过 Flask 框架，就会发现和 Variable-Rules 的语法形式和功能都是相类似的。

问题引入

下面是Django1.X的一段代码：

[python] view plain copy

1. from django.conf.urls import url
2. def year_archive(request, year):
3.

```
year = int(year) # convert str to int
```

4.

```
# Get articles from database
```

5. def detail_view(request, article_id):
6.

```
pass
```

7. def edit_view(request, article_id):

```
8.
 pass
9. def delete_view(request, article_id):
10.
 pass
11. urlpatterns = [
12.
 url('articles/(?P[0-9]{4})/', year_archive),
13.
 url('article/(?P[a-zA-Z0-9]+)/detail/', detail_view),
14.
 url('articles/(?P[a-zA-Z0-9]+)/edit/', edit_view),
15.
 url('articles/(?P[a-zA-Z0-9]+)/delete/', delete_view),
16. ]
```

考虑下这样的两个问题：

第一个问题，函数 year_archive 中year参数是字符串类型的，因此需要先转化为整数类型的变量值当然year=int(year) 不会有诸如如TypeError或者ValueError的异常。那么有没有一种方法，在url中使得这一转化步骤可以由Django自动完成？

第二个问题，三个路由中article_id都是同样的正则表达式，但是你需要写三遍，当之后article_id规改变后，需要同时修改三处代码，那么有没有一种方法，只需修改一处即可？

在Django2.0中，可以使用 path 解决以上的两个问题。

基本示例

这是一个简单的例子：

[python] view plain copy

```
1. from django.urls import path
2. from . import views
3. urlpatterns = [
4.
```

```
 path('articles/2003/', views.special_case_2003),
5.

 path('articles//', views.year_archive),
6.

 path('articles///', views.month_archive),
7.

 path('articles////', views.article_detail),
8. ]
```

基本规则：

- 使用尖括号(<>)从url中捕获值。
- 捕获值中可以包含一个转化器类型（converter type），比如使用 `` 捕获一个整数变量。若果没有化器，将匹配任何字符串，当然也包括了 / 字符。
- 无需添加前导斜杠。

以下是根据 2.0官方文档 而整理的示例分析表：

请求URL	匹配项	视图函数调用式
/articles/2005/03/	第3个	views.month_archive(request, year=2005, month=3)
/articles/2003/	第1个	views.special_case_2003(request)
/articles/2003	无	-
/articles/2003/03/building-a-django-site/	第4个	views.article_detail(request, year=2003, month=3, slug=”building-a-django-site”)

path转化器

文档原文是Path converters，暂且翻译为转化器。

Django默认支持以下5个转化器：

- str,匹配除了路径分隔符（ /）之外的非空字符串，这是默认的形式
- int,匹配正整数，包含0。
- slug,匹配字母、数字以及横杠、下划线组成的字符串。
- uuid,匹配格式化的uuid，如 075194d3-6885-417e-a8a8-6c931e272f00。
- path,匹配任何非空字符串，包含了路径分隔符

注册自定义转化器

对于一些复杂或者复用的需要，可以定义自己的转化器。转化器是一个类或接口，它的要求有三点：

● regex 类属性，字符串类型

● to_python(self, value) 方法，value是由类属性 regex 所匹配到的字符串，返回具体的Python变值，以供Django传递到对应的视图函数中。

● to_url(self, value) 方法，和 to_python 相反，value是一个具体的Python变量值，返回其字符串通常用于url反向引用。

例子：

[python] view plain copy

```
1. class FourDigitYearConverter:
2.
    regex = '[0-9]{4}'
3.
    def to_python(self, value):
4.
        return int(value)
5.
    def to_url(self, value):
6.
        return '%04d' % value
```

使用register_converter 将其注册到URL配置中：

[python] view plain copy

```
1. from django.urls import register_converter, path
2. from . import converters, views
3. register_converter(converters.FourDigitYearConverter, 'yyyy')
4. urlpatterns = [
5.
    path('articles/2003/', views.special_case_2003),
6.
```

原文链接：django path

```
 path('articles//', views.year_archive),
7.

 ...
8. ]
```

使用正则表达式

如果上述的paths和converters还是无法满足需求，也可以使用正则表达式，这时应当使用 django.url .re_path函数。

在Python正则表达式中，命名式分组语法为 (?Ppattern) ，其中name为名称， pattern为待匹配的式。

之前的示例代码也可以写为：

[python] view plain copy

```
1. from django.urls import path, re_path
2. from . import views
3. urlpatterns = [
4.

 path('articles/2003/', views.special_case_2003),
5.

 re_path('articles/(?P[0-9]{4})/', views.year_archive),
6.

 re_path('articles/(?P[0-9]{4})/(?P[0-9]{2})/', views.month_archive),
7.

 re_path('articles/(?P[0-9]{4})/(?P[0-9]{2})/(?P[^/]+)/', views.article_detail),
8. ]
```

这段代码和之前的代码实现了基本的功能，但是还是有一些区别：

- 这里的代码匹配更加严格，比如year=10000在这里就无法匹配。
- 传递给视图函数的变量都是字符串类型，这点和 url 是一致的。

无命名分组

一般来说，不建议使用这种方式，因为有可能引入歧义，甚至错误。

Import变动

django.urls.path 可以看成是 django.conf.urls.url 的增强形式。

为了方便，其引用路径也有所变化。

1.X	2.0	备注
-	django.urls.path	新增，url的强版
django.conf.urls.include	django.urls.include	路径变更
django.conf.urls.url	django.urls.re_path	名同功能，url不会立即废弃

总结

新的path语法可以解决一下以下几个场景：

- 类型自动转化
- 公用正则表达式

将问题引入一节的代码使用新的path函数可以改写如下：

[python] view plain copy

```
1. from django.urls import path, register_converter
2. class ArticleIdConverter:
3.
    regex = '[a-zA-Z0-9]+'
4.
    def to_python(self, value):
5.
        return value
6.
    def to_url(self, value):
7.
        return value
8. register_converter(ArticleIdConverter, 'article_id')
```

原文链接：django path

```
9. def year_archive(request, year):
10.
 year = int(year) # convert str to int
11.
 # Get articles from database
12. def detail_view(request, article_id):
13.
 pass
14. def edit_view(request, article_id):
15.
 pass
16. def delete_view(request, article_id):
17.
 pass
18. urlpatterns = [
19.
 url('articles/(?P[0-9]{4})/', year_archive),
20.
 url('article//detail/', detail_view),
21.
 url('articles//edit/', edit_view),
22.
 url('articles//delete/', delete_view),
23. ]
```

从流程来看，包含了四个步骤：匹配 => 捕获 => 转化 => 视图调用，和之前相比多了转化这一步。