**捕获的参数如何和include()协同工作**

一个被包含的URLconf接收任何来自parent URLconfs的被捕获的参数，比如:

[?](http://www.jb51.net/article/69704.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | # root urls.py    from django.conf.urls.defaults import \*    urlpatterns = patterns('',   (r'^(?P<username>\w+)/blog/', include('foo.urls.blog')),  )    # foo/urls/blog.py    from django.conf.urls.defaults import \*    urlpatterns = patterns('',   (r'^$', 'foo.views.blog\_index'),   (r'^archive/$', 'foo.views.blog\_archive'),  ) |

在这个例子中，被捕获的 username 变量将传递给被包含的 URLconf，进而传递给那个URLconf中的 每一个 视图函数。

注意，这个被捕获的参数 总是 传递到被包含的URLconf中的 每一 行，不管那些行对应的视图是否需要这些参数。 因此，这个技术只有在你确实需要那个被传递的参数的时候才显得有用。  
**额外的URLconf如何和include()协同工作**

相似的，你可以传递额外的URLconf选项到 include() , 就像你可以通过字典传递额外的URLconf选项到普通的视图。 当你这样做的时候，被包含URLconf的 每一 行都会收到那些额外的参数。

比如，下面的两个URLconf在功能上是相等的。

第一个：

[?](http://www.jb51.net/article/69704.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | # urls.py    from django.conf.urls.defaults import \*    urlpatterns = patterns('',   (r'^blog/', include('inner'), {'blogid': 3}),  )    # inner.py    from django.conf.urls.defaults import \*    urlpatterns = patterns('',   (r'^archive/$', 'mysite.views.archive'),   (r'^about/$', 'mysite.views.about'),   (r'^rss/$', 'mysite.views.rss'),  ) |

第二个

[?](http://www.jb51.net/article/69704.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | # urls.py    from django.conf.urls.defaults import \*    urlpatterns = patterns('',   (r'^blog/', include('inner')),  )    # inner.py    from django.conf.urls.defaults import \*    urlpatterns = patterns('',   (r'^archive/$', 'mysite.views.archive', {'blogid': 3}),   (r'^about/$', 'mysite.views.about', {'blogid': 3}),   (r'^rss/$', 'mysite.views.rss', {'blogid': 3}),  ) |

这个例子和前面关于被捕获的参数一样（在上一节就解释过这一点），额外的选项将 总是 被传递到被包含的URLconf中的 每一 行，不管那一行对应的视图是否确实作为有效参数接收这些选项，因此，这个技术只有在你确实需要那个被传递的额外参数的时候才显得有用。 因为这个原因，这种技术仅当你确信在涉及到的接受到额外你给出的选项的每个URLconf时有用的才奏效。