## Определение маршрутов и функции path и re\_path

В прошлой теме рассматривалось сопоставление адресов URL и функций, которые обрабатывают запросы по этим адресам. Например, у нас есть следующие функции в файле **views.py**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | from django.http import HttpResponse    def index(request):      return HttpResponse("<h2>Главная</h2>")    def about(request):      return HttpResponse("<h2>О сайте</h2>")    def contact(request):      return HttpResponse("<h2>Контакты</h2>") |

Это так называемые функции-представления или view function. И в файле **urls.py** проекта они сопоставляются с адресами URL с помощью функции **path()**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | from django.urls import path  from hello import views    urlpatterns = [      path('', views.index),      path('about', views.about),      path('contact', views.contact),  ] |

За сопоставление путей и функций-представлений отвечает функция **path()**, которая располагается в пакете **django.urls** и которая принимает четыре параметра:

|  |  |
| --- | --- |
| 1 | path(route, view, kwargs=None, name=None) |

**route**: представляет шаблон адреса URL, которому должен соответствовать запрос

**view**: функция-представление, которое обрабатывает запрос

**kwargs**: дополнительные аргументы, которые передаются в функцию-представление

**name**: название маршрута

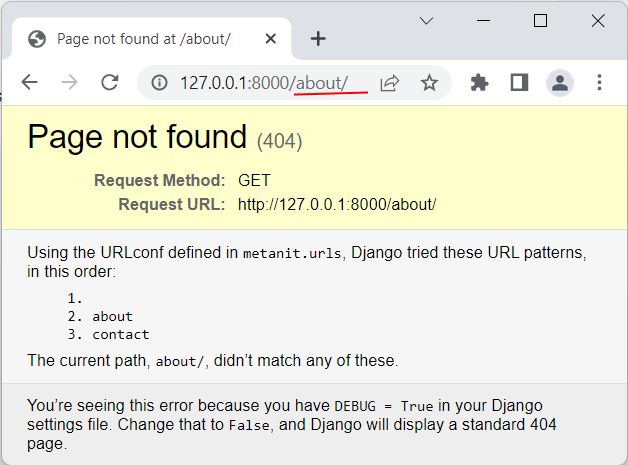
В примере выше применялись только первые два параметра, которые являются обязательными: запрошенный адрес URL и функция, которая обрабатывает запрос по этому адресу. Дополнительно через третий параметр можно указать имя маршрута:

|  |  |
| --- | --- |
| 1 | path('', views.index, name='home'), |

В данном случае маршрут будет называться "home".

### re\_path

Хотя мы можем успешно применять функцию **path()** для определения маршрутов, она довольно ограничена по своему действию. Запрошенный путь должен в точности соответствовать указанному в маршруте адресу URL. Так, в примере выше чтобы функция views.about могла обрабатывать запрос, адрес должен быть в точности "about". Например, стоит нам указать слеш в конце: "about/" и django уже не сможет сопоставить путь с запросом.



В качестве альтернативы для определения маршрутов мы можем использовать функцию **re\_path()**, которая также располагается в пакете **django.urls** и имеет тот же набор параметров:

|  |  |
| --- | --- |
| 1 | re\_path(route, view, kwargs=None, name=None) |

Ее преимущесто состоит в том, что она позволяет задать адреса URL с помощью регулярных выражений.

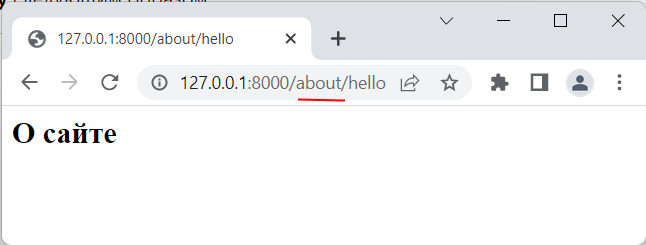
Например, изменим определение файла **urls.py** следующим образом:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | from django.urls import path, re\_path  from hello import views    urlpatterns = [      path('', views.index),      re\_path(r'^about', views.about),      re\_path(r'^contact', views.contact),  ] |

Адрес в первом маршруте по-прежнему образуется с помощью функции path и указывает на корень веб-приложения.

Остальные два маршрута образуются с помощью функции re\_path(). Причем, поскольку определяется регуляное выражение, то перед строкой с шаблоном адреса URL ставится буква **r**. В самом шаблоне адреса можно использовать различные элементы синтаксиса регулярных выражений. В частности, выражение ^about указывает, что адрес должен начинаться с "about". Однако он необязательно в точности должен соответствовать строке "about", как это было в случае с функцией path.

Например, мы можем обратиться по любому адресу, главное чтобы он начинался с "about", и тогда подобный запрос будет обрабатываться функцией views.about.



### Очередность маршрутов

Когда запрос приходит к приложению, то система проверяет соответствие запроса маршрутам по мере их определения: вначале сравнивается первый маршрут, если он не подходит, то сравнивается второй и так далее. Поэтому более общие маршруты должны определяться в последнюю очередь, а более конкретные маршруты должны идти в начале. Например:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | from django.urls import path, re\_path  from hello import views    urlpatterns = [      re\_path(r'^about/contact/', views.contact),      re\_path(r'^about', views.about),      path('', views.index),  ] |

В данном случае адрес "^about/contact" представляет более конкретный маршрут по сравнению c "^about". Поэтому он определяется в первую очередь.

Если бы было наоборот:

|  |  |
| --- | --- |
| 1  2  3  4  5 | urlpatterns = [      path('', views.index),      re\_path(r'^about', views.about),      re\_path(r'^about/contact', views.contact),  ] |

то запрос по адресу "about/contact" обрабатывался бы функцией views.about

### Основные элементы синтаксиса регуляных выражений

Некоторые базовые элементы регуляных выражений, которые можно использовать для определения адресов URL:

^(начало адреса)

$(конец адреса)

+(1 и более символов)

?(0 или 1 символ)

{n}(n символов)

{n, m}(от n до m символов)

.(любой символ)

\d+(одна или несколько цифр)

\D+(одна или несколько НЕ цифр)

\w+(один или несколько буквенных символов)

Рассмотрим несколько возможных сопоставлений шаблонов адресов и запросов:

|  |  |
| --- | --- |
| **Адрес** | **Запрос** |
| r'^$' | http://127.0.0.1/ (корень сайта) |
| r'^about' | http://127.0.0.1/about/ или http://127.0.0.1/about/contact |
| r'^about/contact' | http://127.0.0.1/about/contact |
| r'^products/\d+/' | http://127.0.0.1/products/23/ или http://127.0.0.1/products/6459/abc  Но не соответствует запросу http://127.0.0.1/products/abc/ |
| r'^products/\D+/' | http://127.0.0.1/products/abc/ или http://127.0.0.1/products/abc/123  Не соответствует запросу http://127.0.0.1/products/123/ или http://127.0.0.1/products/123/abc |
| r'^products/phones|tablets/' | http://127.0.0.1/products/phones/1 или http://127.0.0.1/products/tablets/  Не соответствует запросу http://127.0.0.1/products/clothes/ |
| r'^products/\w+' | http://127.0.0.1/products/abc/ или http://127.0.0.1/products/123/  Не соответствует запросу http://127.0.0.1/products/abc-123 |
| r'^products/[-\w]+/' | http://127.0.0.1/products/abc-123 |
| r'^products/[A-Z]{2}/' | http://127.0.0.1/products/RU  Не соответствует запросам http://127.0.0.1/products/Ru или http://127.0.0.1/products/RUS |

### Передача значений в функцию

Выше были рассмотрены все параметры функций path и re\_path, кроме одного - kwargs, который позволяет передать в функцию-представление некоторые значения. Например, в файле **views.py** определим следующие функции:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | from django.http import HttpResponse    def index(request):      return HttpResponse("<h2>Главная</h2>")    def about(request, name, age):      return HttpResponse(f"""              <h2>О пользователе</h2>              <p>Имя: {name}</p>              <p>Возраст: {age}</p>      """) |

Здесь функция about() также принимает два дополнительных параметра: name и age (условно имя и возраст пользователя). В функции их значения отправляются пользователю вместе с остальным содержимым.

Изменим файл **urls.py**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | from django.urls import path  from hello import views    urlpatterns = [      path('', views.index),      path('about', views.about, kwargs={"name":"Tom", "age": 38}),  ] |

С помощью параметра kwargs в функцию about передается словарь с двумя значениями - для двух параметров функции. Соответственно при обращении к этой функции мы увидим в браузере соответствующие данные:

