## Параметры представлений

Функции-представления могут принимать параметры, через которые могут передаваться различные данные. Подобные параметры передаются в адресе URL. Например, в запросе

|  |  |
| --- | --- |
| 1 | http://127.0.0.1:8000/index/Tom/38/ |

последние два сегмента Tom/38/ могут представлять параметры URL, которые могут быть связанны с параметрами функции-представления через систему маршрутизации. Подобные параметры еще можно назвать **параметрами маршрута**

### Определение параметров через функцию path

Определим в файле **views.py** следующие функции:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | from django.http import HttpResponse    def index(request):      return HttpResponse("<h2>Главная</h2>")    def user(request, name):      return HttpResponse(f"<h2>Имя: {name}</h2>") |

В данном случае функция user имеет два параметра. Для второго параметра - name мы будем получать данные из строки запроса. То есть это будет параметр маршрута. И как любой другой параметр мы сможем использовать его внутри функции.

Далее в файле **urls.py** определим следующий код:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | from django.urls import path  from hello import views    urlpatterns = [      path("", views.index),      path("user/<str:name>", views.user),  ] |

Для второго маршрута здесь определен параметр name (который соответствует параметру name функции views.user). Параметры заключаются в угловые скобки в формате <спецификатор:название\_параметра>. Например, здесь параметр name имеет спецификатор **str**.

По умолчанию Django предоставляет следующие спецификаторы:

**str**: соответствует любой строке за исключенем символа "/". Если спецификатор не указан, то используется по умолчанию

**int**: соответствует любому положительному числу

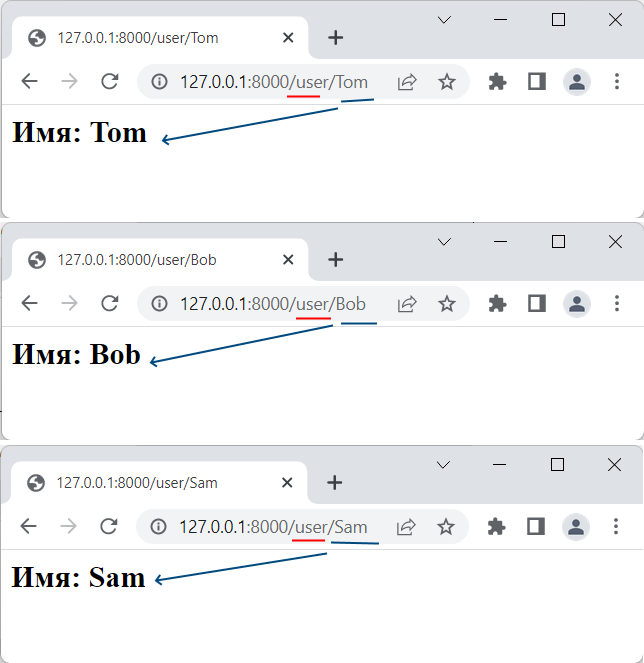
**slug**: соответствует последовательности буквенных символов ASCII, цифр, дефиса и символа подчеркивания, например, building-your-1st-django-site

**uuid**: сооветствует идентификатору UUID, например, 075194d3-6885-417e-a8a8-6c931e272f00

**path**: соответствует любой строке, которая также может включать символ "/" в отличие от спецификатора str

Также отмечу, что количество и название параметров в шаблонах адресов URL соответствуют количеству и названиям параметров соответствующих функций, которые обрабатывают запросы по данным адресам.

Запустим приложение, обратимся к функции views.user, например, с помошью запроса *http://127.0.0.1:8000/user/Tom*, и через строку запроса - через ее третий сегмент мы сможем передать значение для параметра name



Подобным образом можно определить и большее количество параметров. Например, добавим второй параметр в функцию user в **views.py**:

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

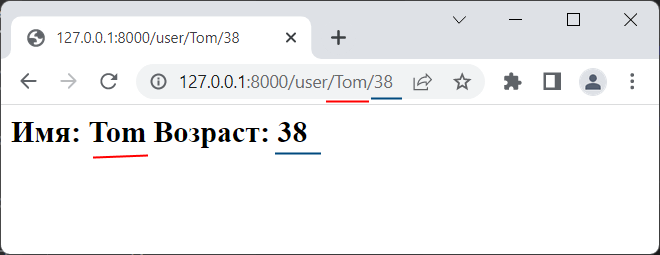
А в файле **urls.py** добавим в маршрут параметр age:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | from django.urls import path  from hello import views    urlpatterns = [      path("", views.index),      path("user/<name>/<int:age>", views.user),  ] |

Предполагается, что параметр age будет представлять число, поэтому для него используется спецификатор **int**. В этом случае мы можем обратиться к функции user, например, с помощью запроса

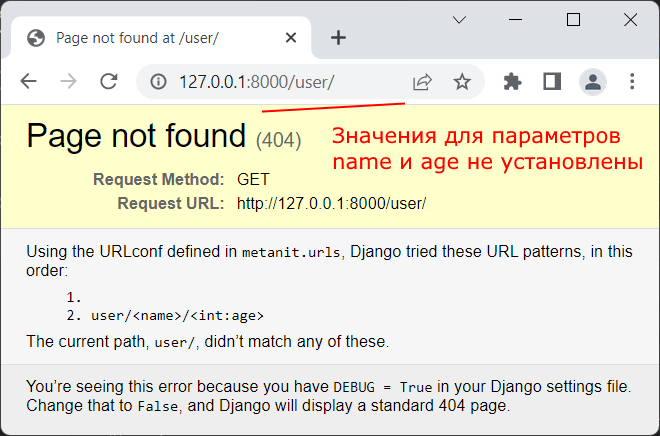
|  |  |
| --- | --- |
| 1 | http://127.0.0.1:8000/user/Tom/38 |

В этом случае сегмент "/Tom/" будет сопоставлен с параметром name, а "/38" - с параметром age.



#### Значения для параметров по умолчанию

В примере выше использовались два параметра, но что, если мы не передадим для одного или обоих параметров значения?



В этом случае мы получим ошибку. Параметры маршрута являются частью шаблона URL. Соответственно если в строке запроса не передаются значения для параметров, такой запрос не соответствует маршруту. Поэтому Django не сможет найти нужный маршрут для обработки запроса, и мы получим ошибку 404 (ресурс не найден).

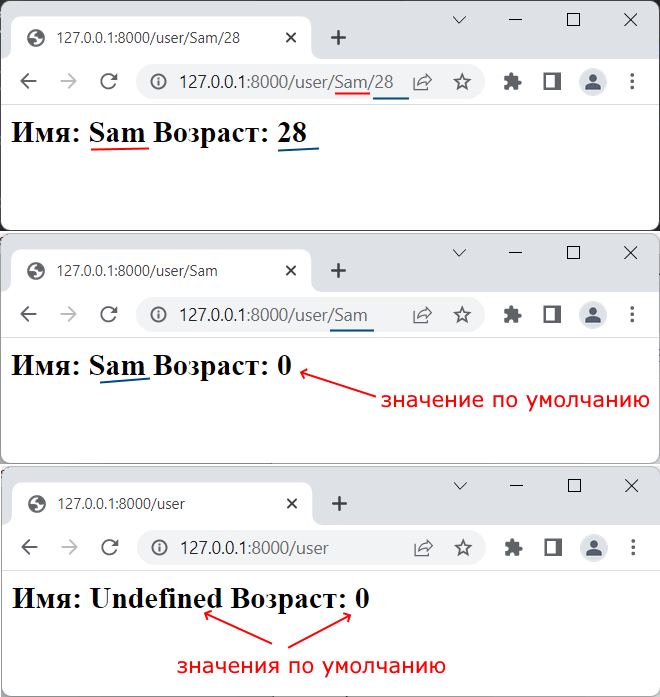
Однако такое поведение не всегда может быть желательным. И мы можем задать для параметров маршрута значения по умолчанию на случай, если через строку запроса не передаются значения. Так, для функции user в **views.py** определим значения для параметров по умолчанию:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | from django.http import HttpResponse    def index(request):      return HttpResponse("<h2>Главная</h2>")    def user(request, name="Undefined", age =0):      return HttpResponse(f"<h2>Имя: {name}  Возраст: {age}</h2>") |

В данном случае, если для параметра name не передается значение, то он получает в качестве значения строку "Undefined". Для параметра age значение по умолчанию 0.

В этом случае для функции user в файле **urls.py** надо определить дополнительные маршруты, которые не учитывают необязательные параметры:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | from django.urls import path  from hello import views    urlpatterns = [      path("", views.index),      path("user", views.user),      path("user/<name>", views.user),      path("user/<name>/<int:age>", views.user),  ] |



### Определение параметров через функцию re\_path

Подобным образом мы можем использовать функцию **re\_path** для определения параметров. Определим в приложении в файле **views.py** следующие функции:

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

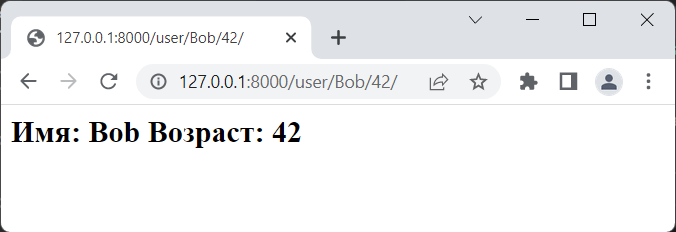
Теперь изменим файл **urls.py**, чтобы он мог сопоставить данные функции с запросами:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | from django.urls import path, re\_path  from hello import views    urlpatterns = [      path("", views.index),      re\_path(r"^user/(?P<name>\D+)/(?P<age>\d+)", views.user),  ] |

Для представления параметра в шаблоне адреса используется выражение ?P<>. Общее определение параметру соответствует формату (?P<имя\_параметра>регулярное\_выражение). Между угловыми скобками помещается название параметра. После закрывающей угловой скобки идет регулярное выражение, которому дожно соответствовать значение параметра.

Во втором шаблоне адреса определяются два параметра: name и age. При этом параметр age должен представлять число, а параметр name должен состоять только из буквенных символов.

Теперь мы можем через адресную строку передать данные в приложение:



Также мы можем указать для определенных параметров значения по умолчанию:

|  |  |
| --- | --- |
| 1  2 | def user(request, name="Undefined", age =0):      return HttpResponse(f"<h2>Имя: {name}  Возраст: {age}</h2>") |

В этом случае надо дополнительно определить еще маршруты в файле **urls.py** для тех запросов, в которых не передаются значения для маршрутов:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | from django.urls import path, re\_path  from hello import views    urlpatterns = [      path("", views.index),      re\_path(r"^user/(?P<name>\D+)/(?P<age>\d+)", views.user),      re\_path(r"^user/(?P<name>\D+)", views.user),      re\_path(r"^user", views.user),  ] |

Обратите внимание на порядок размещения маршрутов: в отличие от случая с функцией path теперь сначала размещаются более конкретные маршруты с большим количеством параметров.