

Лекция 2. **Представления** и **маршрутизация** – ключевые элементы Django

Часть 2



Передача параметров в функции представления через URL-адрес с использованием системы маршрутизации:

Дополним файл views.py функциями `products` и `users` :

```
# Create your views here.
```

```
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>")
```

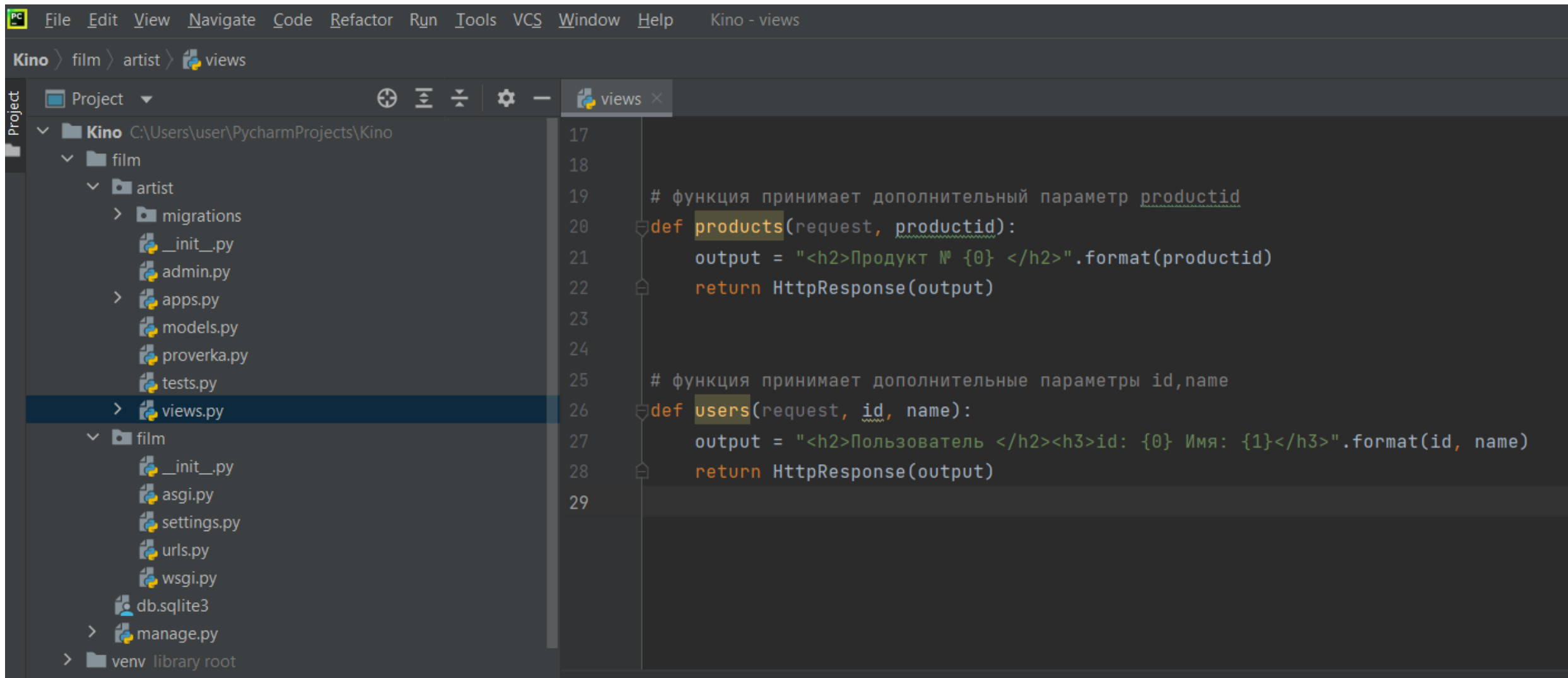
```
def products(request, productid):  
    output = "<h2>Продукт № {0}</h2>".format(productid)  
    return HttpResponse(output)
```

```
def users(request, id, name):  
    output = "<h2>Пользователь </h2><h3>id: {0} Имя: {1}</h3>".format(id, name)  
    return HttpResponse(output)
```

функция `products` принимает
дополнительный параметр `productid`

функция `users` принимает дополнительные
параметры `id, name`





Соответственно изменим файл urls.py:

```
from django.urls import re_path
```

```
from artist import views
```

```
urlpatterns = [
```

```
    re_path(r'^products/(?P<productid>\d+)/', views.products),
```

```
    re_path(r'^users/(?P<id>\d+)/(?P<name>\D+)/', views.users),
```

```
]
```

Здесь ?P<> - представление параметра в шаблоне адреса

Общий формат определения параметра:

(?P<имя_параметра>регулярное выражение)



File Edit View Navigate Code Refactor Run Tools VCS Window Help Kino - urls

Kino > film > film > urls

Project

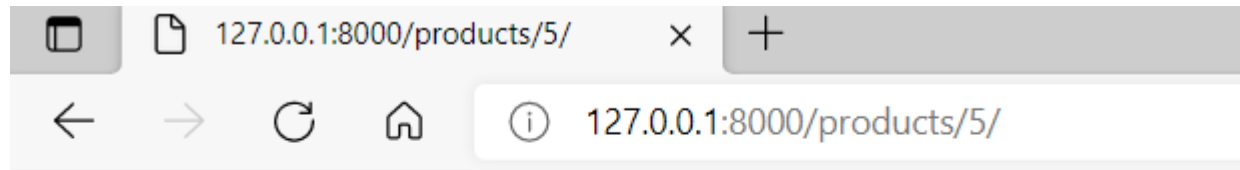
- Project
- Kino C:\Users\user\PycharmProjects\Kino
 - film
 - artist
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - proverka.py
 - tests.py
 - views.py
 - film
 - __init__.py
 - asgi.py
 - settings.py
 - urls.py
 - wsgi.py
 - db.sqlite3
 - manage.py
 - venv library root
 - main.py

```
14      2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15      """
16      from django.contrib import admin
17      from django.urls import path, re_path
18      from artist import views
19
20      urlpatterns = [
21          path('', views.index, name='home'),
22          path('admin/', admin.site.urls),
23          re_path(r'^about', views.about, name='about'),
24          re_path(r'^contact', views.contact, name='contact'),
25          re_path(r'^products/(?P<productid>\d+)/', views.products),
26          re_path(r'^users/(?P<id>\d+)/(?P<name>\D+)/', views.users),
27      ]
28
29
```



Запустим локальный сервер разработки: `python manage.py runserver`, и укажем адрес:

<http://127.0.0.1:8000/products/5> - (запрос данных о продукте №5, будет возвращена страница с информацией о продукте №5)



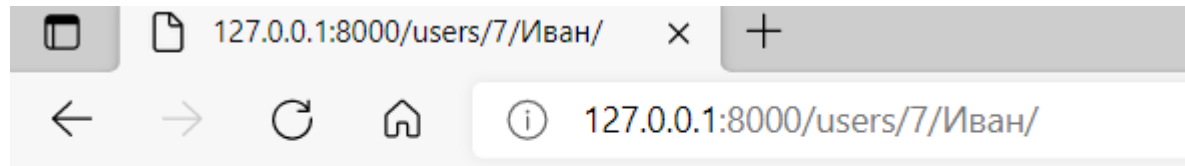
Продукт № 5



Продукт № 9

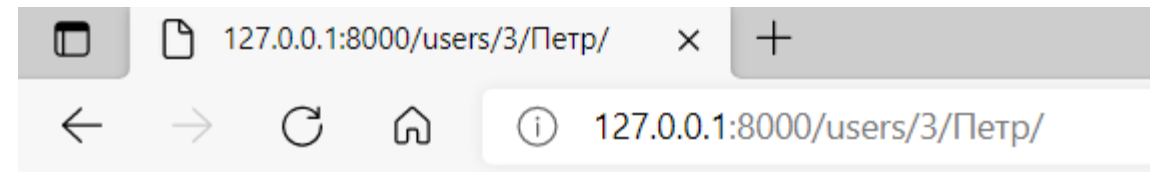


<http://127.0.0.1:8000/users/7/Иван/> (запрос о пользователе с идентификационным номером 7 по имени Иван, будет возвращена страница с информацией о пользователе с идентификационным номером 7 по имени Иван)



Пользователь

id: 7 Имя: Иван

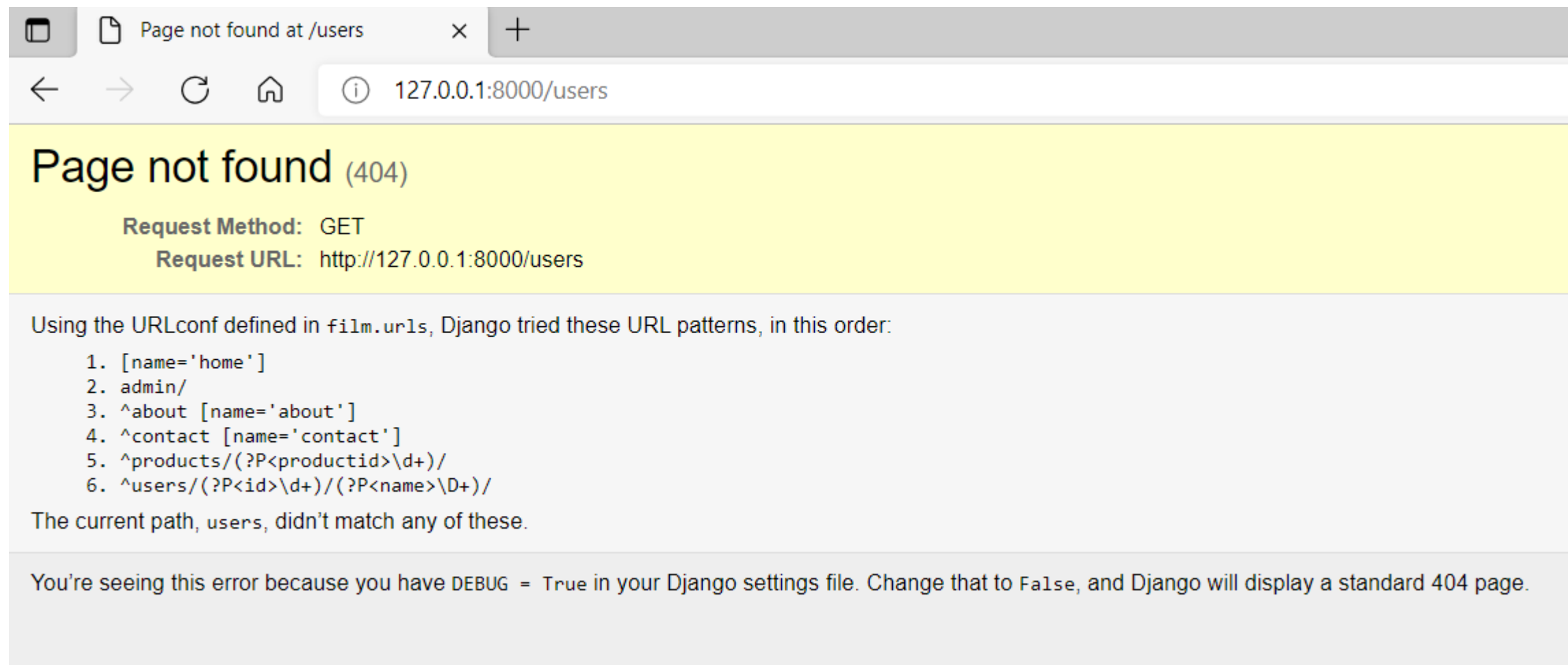



Пользователь





id: 3 Имя: Петр



Однако, если адрес будет введен неточно и/или данные по запросу не будут найдены, то пользователю будет выдано сообщение об ошибке:



 Page not found at /products × +

    ⓘ 127.0.0.1:8000/products

Page not found (404)

Request Method: GET
Request URL: http://127.0.0.1:8000/products

Using the URLconf defined in `film.urls`, Django tried these URL patterns, in this order:

1. `[name='home']`
2. `admin/`
3. `^about [name='about']`
4. `^contact [name='contact']`
5. `^products/(?P<productid>\d+)/`
6. `^users/(?P<id>\d+)/(?P<name>\D+)/`

The current path, `products`, didn't match any of these.

You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.



Определение параметров по умолчанию во избежание ошибки, возникающей при неправильном наборе адреса

Если адрес будет введен неточно и/или данные по запросу не будут найдены, то пользователю будет выдано сообщение об ошибке.

Чтобы это предотвратить, можно использовать параметры по умолчанию. Тогда вместо сообщения об ошибке будут возвращены данные по умолчанию. Так, если в функцию не будет передано значение для параметра `productid`, то будет присвоено значение по умолчанию **1** и вместо сообщения об ошибке будут выданы данные по умолчанию. Для этого:

```
def products(request, productid=1):  
  
    output = "<h2>Продукт № {0} </h2>".format(productid)  
  
    return HttpResponse (output)
```

И в файле `urls.py` надо будет прописать еще маршрут:

```
re_path(r'^products/$',views.products).
```



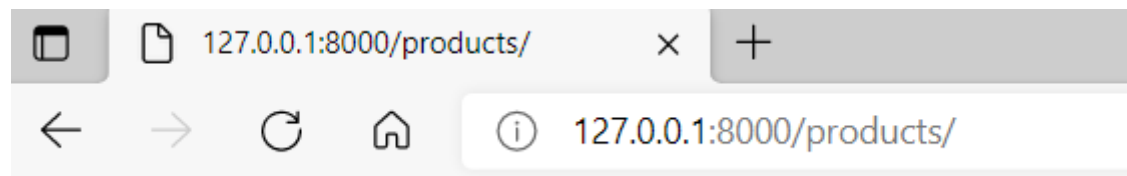
```
19 # функция принимает дополнительный параметр productid
20 def products(request, productid=1):
21     output = "<h2>Продукт № {0} </h2>".format(productid)
22     return HttpResponse(output)
23
```

Значение параметра `productid`
по умолчанию

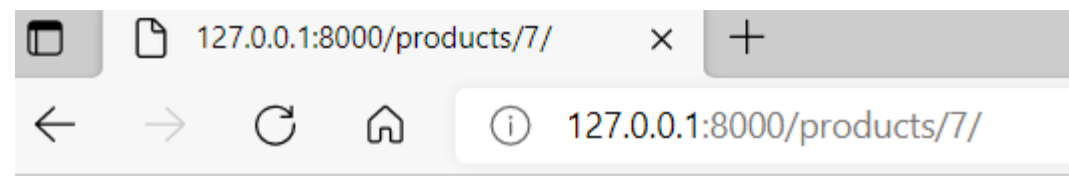
```
14 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 from django.contrib import admin
17 from django.urls import path, re_path
18 from artist import views
19
20 urlpatterns = [
21     path('', views.index, name='home'),
22     path('admin/', admin.site.urls),
23     re_path(r'^about', views.about, name='about'),
24     re_path(r'^contact', views.contact, name='contact'),
25     re_path(r'^products/(?P<productid>\d+)/', views.products),
26     re_path(r'^users/(?P<id>\d+)/(?P<name>\D+)/', views.users),
27     re_path(r'^products/$', views.products),
28
29
30 ]
31
```

Дополнительный маршрут





Продукт № 1



Продукт № 7



Page not found (404)

Request Method: GET

Request URL: http://127.0.0.1:8000/productsjjj

Using the URLconf defined in `film.urls`, Django tried these URL patterns, in this order:

1. `[name='home']`
2. `admin/`
3. `^about [name='about']`
4. `^contact [name='contact']`
5. `^products/(?P<productid>\d+)/`
6. `^users/(?P<id>\d+)/(?P<name>\D+)/`
7. `^products/$`

The current path, `productsjjj`, didn't match any of these.

You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.



И аналогично сделаем для users:

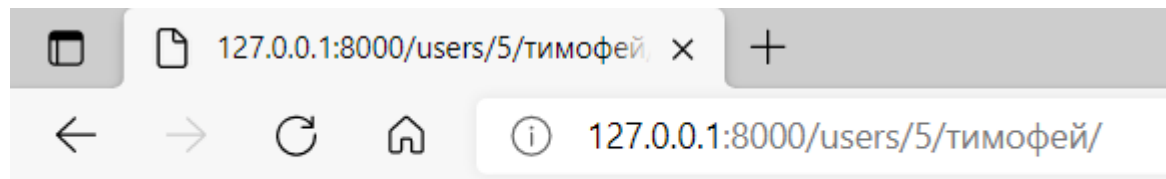
```
4
5 # функция принимает дополнительные параметры id,name
6 def users(request, id=1, name="Некто"):
7     output = "<h2>Пользователь </h2><h3>id: {0} Имя: {1}</h3>".format(id, name)
8     return HttpResponse(output)
9
```

Значения
параметров id, name
по умолчанию

```
0 urlpatterns = [
1     path('', views.index, name='home'),
2     path('admin/', admin.site.urls),
3     re_path(r'^about$', views.about, name='about'),
4     re_path(r'^contact$', views.contact, name='contact'),
5     re_path(r'^products/(?P<productid>\d+)/', views.products),
6     re_path(r'^users/(?P<id>\d+)/(?P<name>\D+)/', views.users),
7     re_path(r'^products/$', views.products),
8     re_path(r'^users/$', views.users),
9
0
```

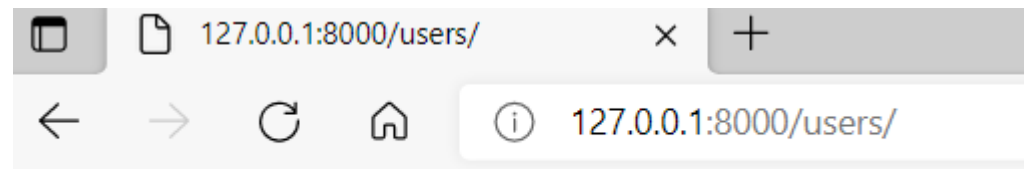
Дополнительный маршрут





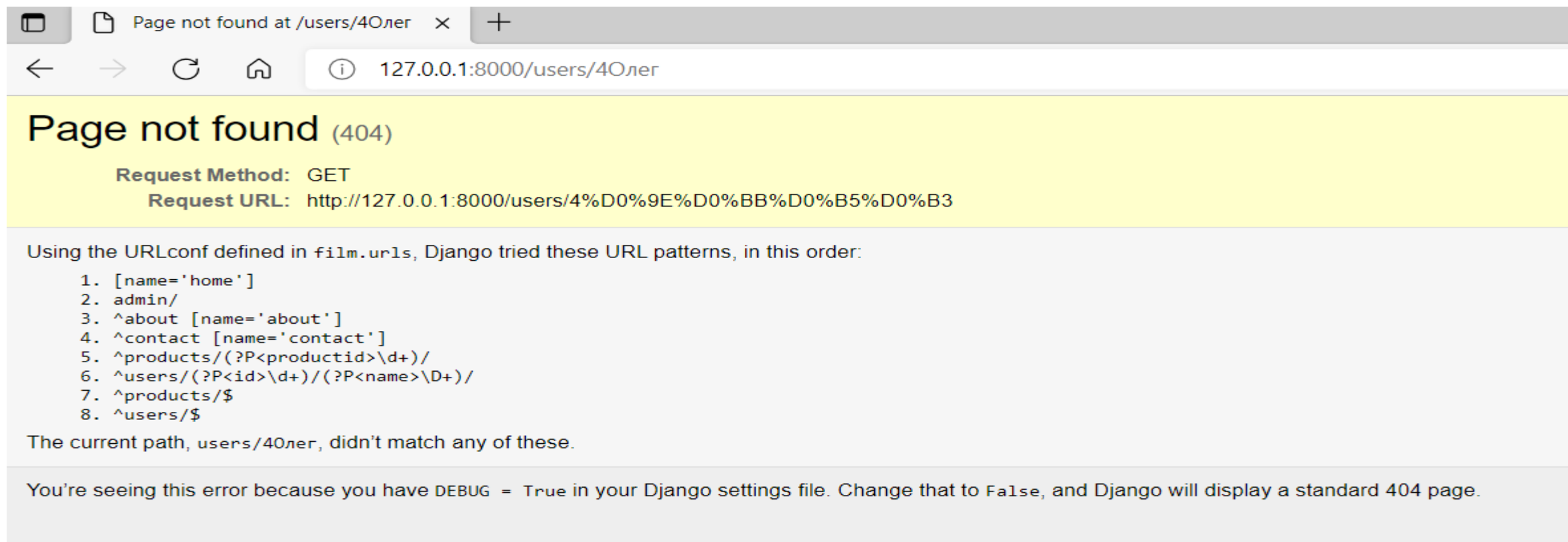
Пользователь

id: 5 Имя: тимофей



Пользователь

id: 1 Имя: Некто



Определение параметров через функцию path()

Формат параметров функции **path()**: <спецификатор:название_параметра>

Django предоставляет спецификаторы:

str - любой текст, кроме символа (/)

int – любое целое положительное число

slug -последовательность буквенных символов ASCII, цифр, дефиса и символа подчеркивания.

uuid – идентификатор UUID

path – любая строка, включая символ подчеркивания



При указании пути применим спецификаторы int и str:

```
path('products/<int:productid>/', views.products),
```

```
path('users/<int:id>/<str:name>/', views.users),
```

```
from django.contrib import admin
from django.urls import path, re_path
from artist import views

urlpatterns = [
    path('', views.index, name='home'),
    path('admin/', admin.site.urls),
    re_path(r'^about', views.about, name='about'),
    re_path(r'^contact', views.contact, name='contact'),
    path('products/<int:productid>/', views.products),
    path('users/<int:id>/<str:name>/', views.users),
]
```




```
# Create your views here.

def index(request):
    return HttpResponse('Привет, Артист! Это работает представление index')

def about(request):
    return HttpResponse(' Это страница с информацией о сайте')

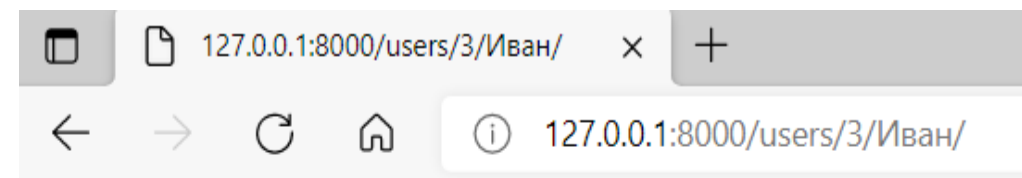
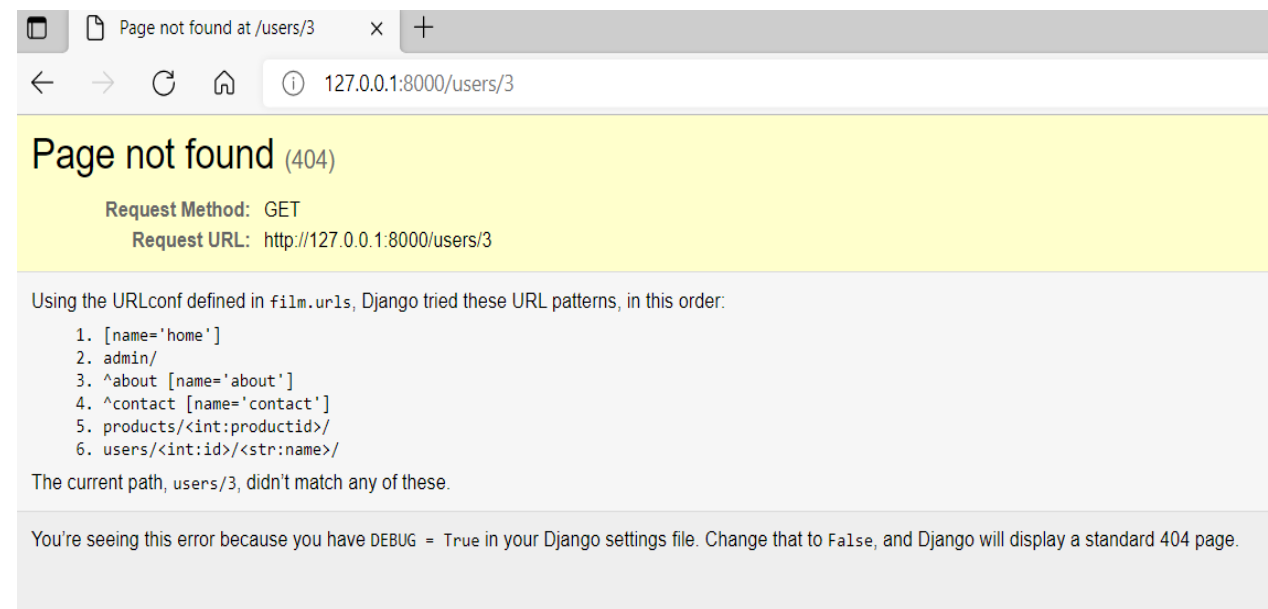
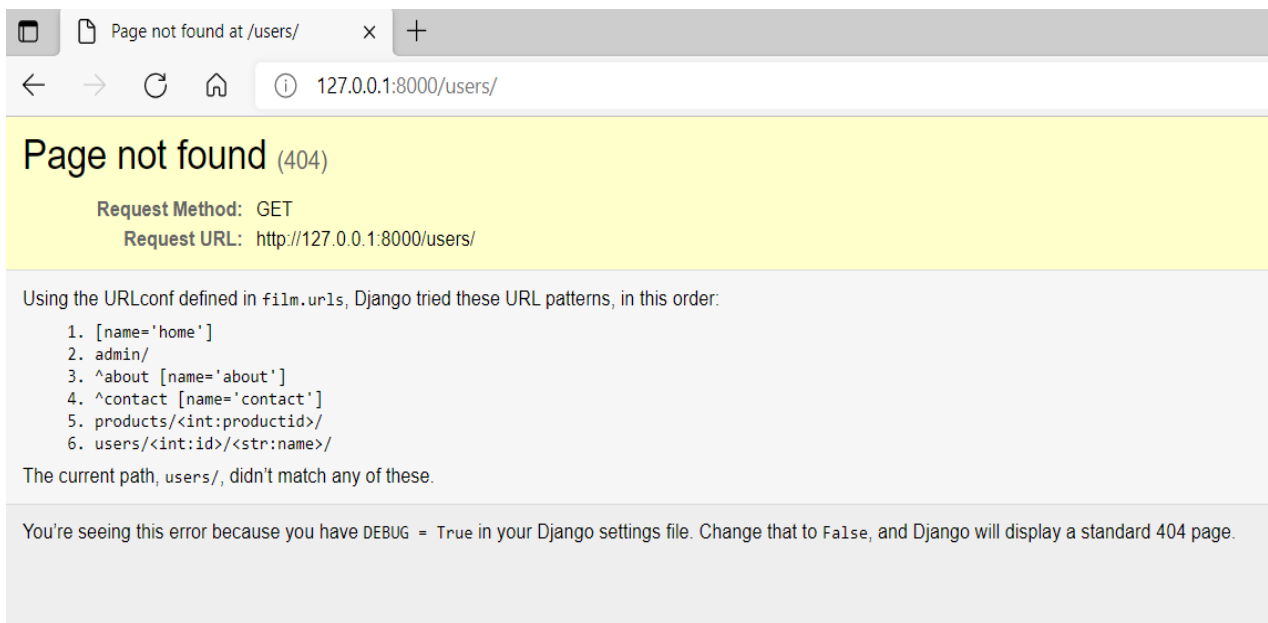
def contact(request):
    return HttpResponse('Здесь размещены контакты')

# функция принимает дополнительный параметр productid
def products(request, productid=1):
    output = "<h2>Продукт № {0} </h2>".format(productid)
    return HttpResponse(output)

# функция принимает дополнительные параметры id,name
def users(request, id=1, name="Некто"):
    output = "<h2>Пользователь </h2><h3>id: {0} Имя: {1}</h3>".format(id, name)
    return HttpResponse(output)
```

Функции products и users в файле views.py:

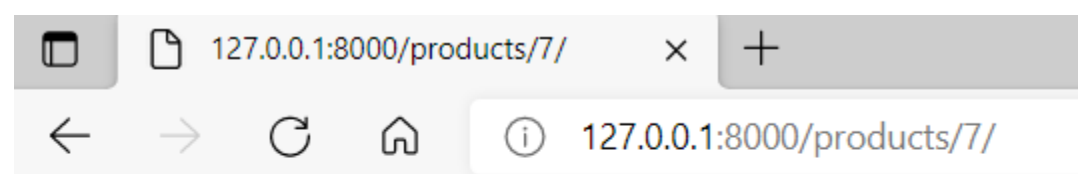
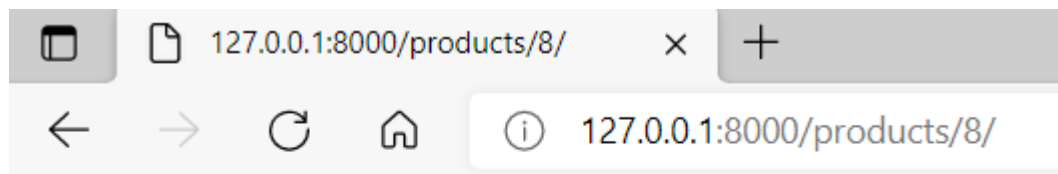
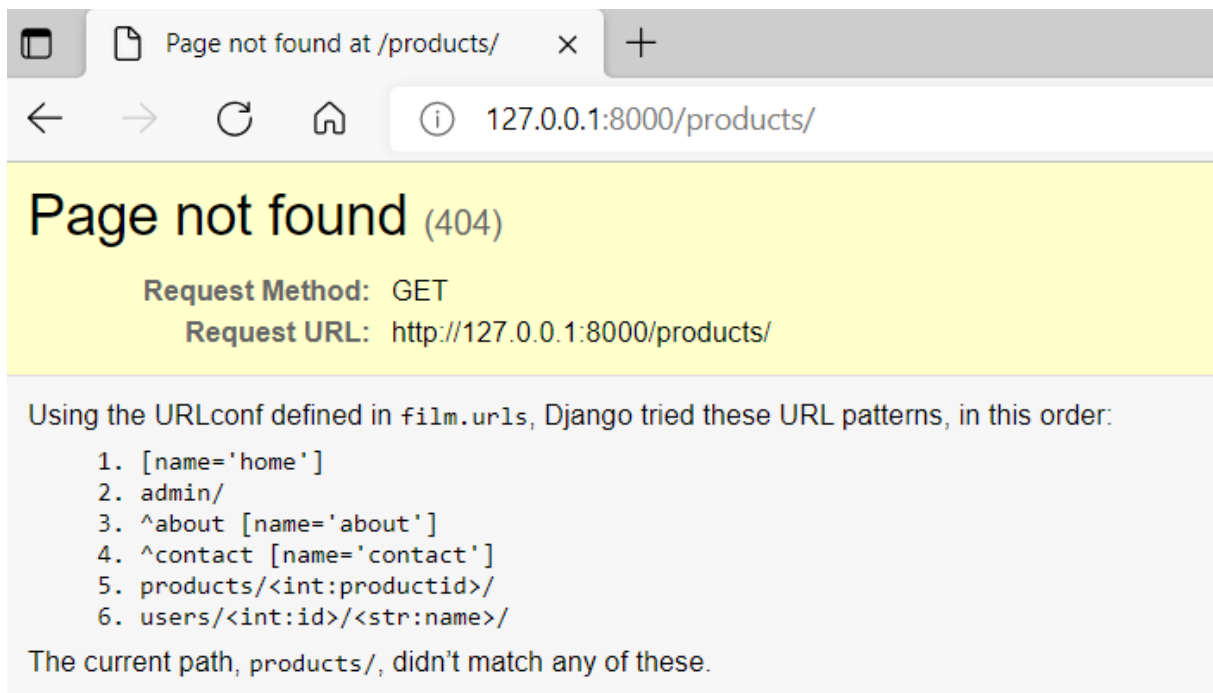




Пользователь

id: 3 Имя: Иван





Определение параметров по умолчанию в функции path()

В файле `views.py` заданы значения для параметров по умолчанию:

```
from django.http import HttpResponse
```

```
def products(request, productid=1):
```

```
    output = "<h2>Product № {0}</h2>".format(productid)
```

```
    return HttpResponse(output)
```

```
def users(request, id=1, name="Dirk"):
```

```
    output = "<h2>User</h2><h3>id: {0} name: {1}</h3>".format(id, name)
```

```
    return HttpResponse(output)
```



```
from django.urls import path
```

```
from artist import views
```

```
urlpatterns = [
```

```
    path('products/', views.products),          # маршрут по умолчанию
```

```
    path('products/<int:productid>/', views.products),
```

```
    path('users/', views.users),                # маршрут по умолчанию
```

```
    path('users/<int:id>/<str:name>/', views.users),
```

```
]
```

для каждой функции в файле **urls.py** определим по два маршрута:

один маршрут для случая, когда заданы значения параметров,

второй маршрут для случая, когда значения параметров принимаются по умолчанию



Файл `urls.py` после изменений:

```
from django.contrib import admin
```

```
from django.urls import path
```

```
from django.urls import re_path
```

```
from artist import views
```

```
urlpatterns = [
```

```
    path('', views.index),
```

```
    re_path(r'^about', views.about),
```

```
    re_path(r'^contact', views.contact),
```

```
    path('products/', views.products), # маршрут по умолчанию
```

```
    path('products/<int:productid>/', views.products),
```

```
    path('users/', views.users), # маршрут по умолчанию
```

```
    path('users/<int:id>/<str:name>/', views.users),
```

```
]
```



```
from django.contrib import admin
from django.urls import path, re_path
from artist import views

urlpatterns = [
    path('', views.index, name='home'),
    path('admin/', admin.site.urls),
    re_path(r'^about', views.about, name='about'),
    re_path(r'^contact', views.contact, name='contact'),
    path('products/', views.products),
    path('products/<int:productid>', views.products),
    path('users/', views.users),
    path('users/<int:id>/<str:name>', views.users),
]
```

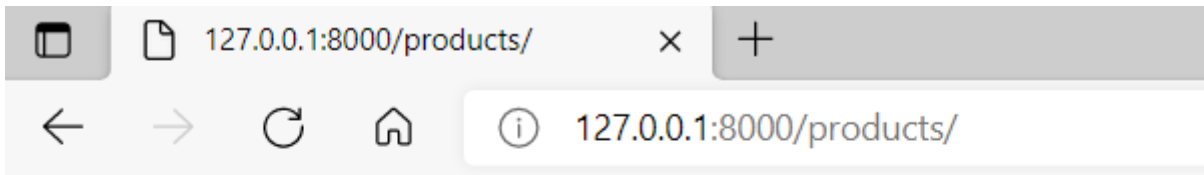


Перейдем на сервер или перезапустим его:

```
python manage.py runserver
```

И попробуем ввести разные варианты адресов для [users](#) и [products](#)

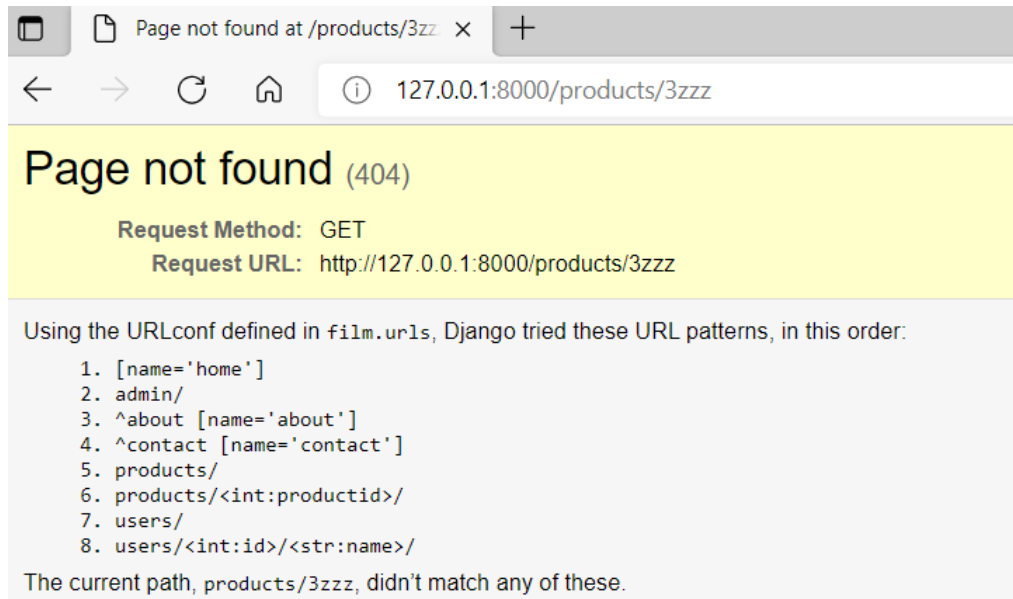


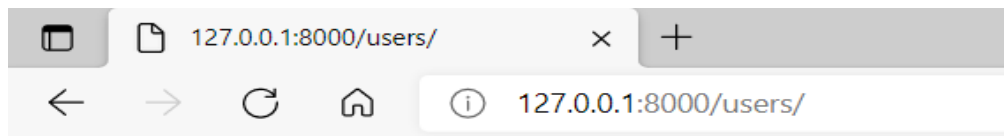


Продукт № 1



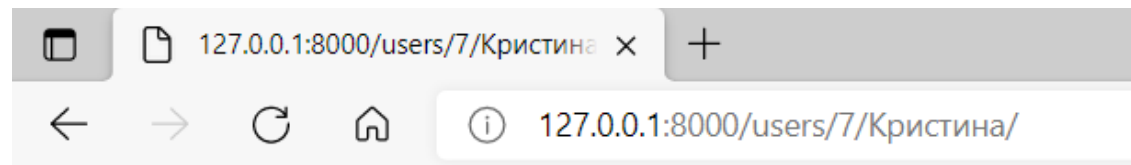
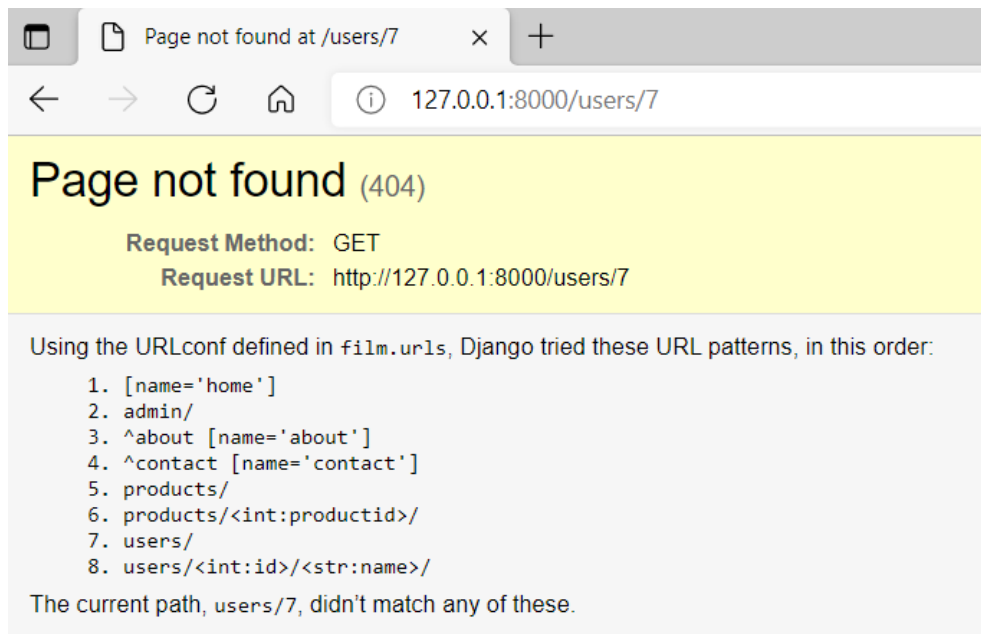
Продукт № 7





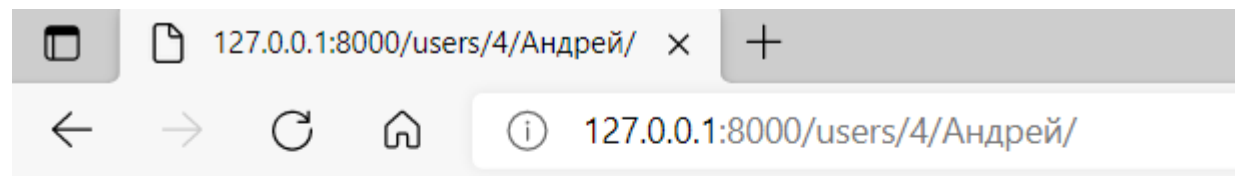
Пользователь

id: 1 Имя: Некто



Пользователь

id: 7 Имя: Кристина



Пользователь

id: 4 Имя: Андрей



Конец части 2. Продолжение следует..

