

## Отчет по лабораторной работе №5 «Шаблонизация»

Выполнил:

Катков Александр, ИУ5-52

---

Преподаватель:

Гапанюк Ю.Е.

---

2017 г.

## 1) Задание лабораторной работы.

В этой ЛР вы создадите Django-проект, покажете пользователю статичную страницу, познакомитесь с конструкциями шаблонизаторов: переменные, теги, наследование шаблонов.

- Создать проект
  - Реализовать view, в которых генерируются html-страницы
  - В шаблонах должны быть использованы рассмотренные конструкции: переменные, вложенные значения, циклы, условия
  - Все шаблоны должны расширять базовый шаблон
  - Для элементов списка использовать тег include
  - По нажатию на элемент списка должна открываться страница информации об элементе
- Для верстки необходимо использовать Bootstrap<sup>21</sup>. Выведите название построенной

## 2) Листинг

Urls.py

```
"""
The 'urlpatterns' list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/1.11/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  url(r'^$', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  url(r'^$', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.conf.urls import url, include
    2. Add a URL to urlpatterns:  url(r'^blog/', include('blog.urls'))
"""
import ...

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^function_view/', views.function_view),
    url(r'^class_based_view/', views.ExampleClassBased.as_view()),
    url(r'^example_view/', views.ExampleView.as_view()),
    url(r'^orders/', views.OrdersView.as_view()),
    url(r'^order/(?P<id>\d+)', views.OrderView.as_view(), name='order_url'),
    url(r'^zadanie/', views.ZadanieView.as_view()),
    url(r'^zadanie/(?P<id>\d+)', views.ZadanieView.as_view(), name='zadanie_url'),
```

## Views.py

```
from django.shortcuts import render
from django.http import HttpResponse
from django.views.generic import View
# Create your views here.

def function_view(request):
    return HttpResponse('response from function view')

class ExampleClassBased(View):
    def get(self, request):
        return HttpResponse('response from class based view')

class ExampleView(View):
    def get(self, request):
        return render(request, 'example.html', {'list': ['1', '2', '3', 4, 5]})

class OrdersView(View):
    def get(self, request):
        data = {
            'orders': [
                {'title': 'Первый заказ', 'id': 1},
                {'title': 'Второй заказ', 'id': 2},
                {'title': 'Третий заказ', 'id': 3}
            ]
        }
        return render(request, 'orders.html', data)

class OrderView(View):
    def get(self, request, id):
        data = {
            'order': {
                'id': id
            }
        }
        return render(request, 'order.html', data)

class OrderView(View):
    def get(self, request, id):
        data = {
            'order': {
                'id': id
            }
        }
        return render(request, 'order.html', data)

class ZadanieView(View):
    def get(self, request):
        data = {
            'orders': [
                {'title': 'Первый заказ', 'id': 1},
                {'title': 'Второй заказ', 'id': 2},
                {'title': 'Третий заказ', 'id': 3}
            ]
        }
        return render(request, 'zadanie1.html', data)

class Zadanie1View(View):
    def get(self, request, id1):
        data = {
            'order': {
                'id': id1
            }
        }
        return render(request, 'el_zadanie.html', data)
```

## Base.html

```
<!DOCTYPE html>

<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>{% block title %}{% endblock %}</title>
  {% block head %}{% endblock %}
</head>
<body>
<div>Общая часть</div>
<div>
  {% block body %}Переопределяется в наследниках{% endblock %}
</div>
</body>
</html>
```

## Orders.html

```
{% extends 'base.html' %}
{% load static %}
{% block head %}
  <link rel="stylesheet" type="text/css" href="{% static 'css/aaa.css' %}">
{% endblock %}
{% block title %}Заказы{% endblock %}
{% block body %}
  <ul>
    {% for order in orders %}
      {% include 'order_element.html' with element=order %}
    {% empty %}
      <li>пустой список</li>
    {% endfor %}
  </ul>
{% endblock %}
```

## Zadanie.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- The above 3 meta tags "must" come first in the head; any other head content must come "after" these tags -->
  <title>{% block title %}{% endblock %}</title>

  <!-- Bootstrap -->
  <link href="css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" type="text/css" href="{% static 'css/bootstrap.min.css' %}">
  <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
  <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
  <!--[if lt IE 9]>
    <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
    <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
  <![endif]-->
</head>
<body>
  {% block body %}Переопределяется в наследниках{% endblock %}

  <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
  <!-- Include all compiled plugins (below), or include individual files as needed -->
  <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

## Zadanie1.html

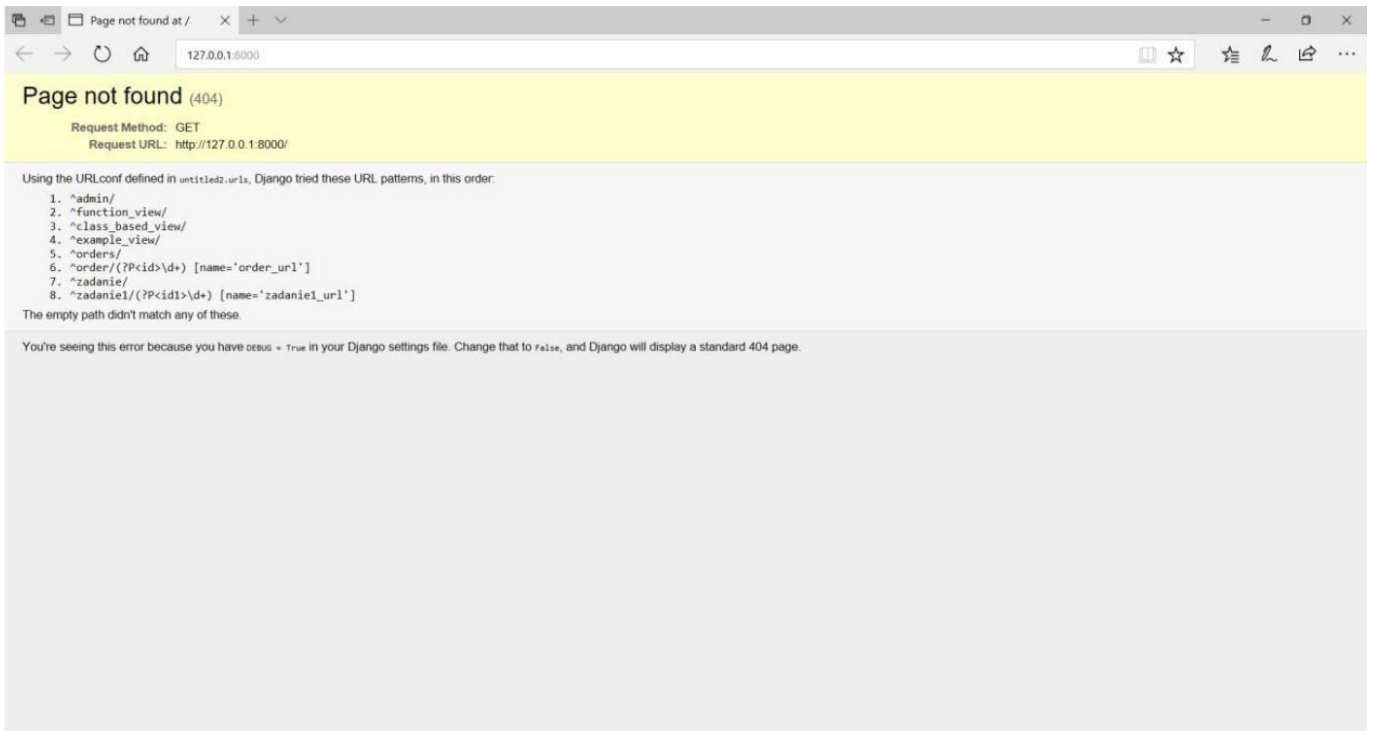
```
{% extends 'zadanie.html' %}

{% block title %}{% endblock %}
{% block body %}
    {% for z in orders %}
        {% include 'zadanie2.html' with element=z %}
    {% endfor %}
{% endblock %}
```

## Zadanie2.html

```
<div class="alert alert-success" role="alert">
    <a href="{% url 'zadanie1 url' element.id %}" class="proba">{{ element.title }}</a>
</div>
```

### 3) Результат:



- Первый заказ
- Второй заказ
- Третий заказ

Заказ № 1

127.0.0.1:8000/order/1

Общая часть  
Страница заказа №1

127.0.0.1

127.0.0.1:8000/zadanie

Первый заказ

Второй заказ

Третий заказ

127.0.0.1

127.0.0.1:8000/zadanie1/1

1