

**Московский государственный технический  
университет им. Н.Э. Баумана.**

**Факультет «Информатика и управление»**

**Кафедра ИУ5. Курс «Разработка интернет приложений»**

**Отчет по лабораторной работе №5**

**«Django. ШАБЛОНИЗАЦИЯ»**

Выполнил:  
студент группы ИУ5-53  
Егоров С.А.

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю.Е.

Москва, 2017 г.

## 1 Задание

Создать Django-проект, показать пользователю статичную страницу, познакомиться с конструкциями шаблонизаторов: переменными, тегами, наследованием шаблонов.

1. Создать проект
2. Реализовать view, в которых генерируются html-страницы
3. В шаблонах должны быть использованы рассмотренные конструкции: переменные, вложенные значения, циклы, условия
4. Все шаблоны должны расширять базовый шаблон
5. Для элементов списка использовать тег include
6. По нажатию на элемент списка должна открываться страница информации об элементе
7. Для верстки необходимо использовать Bootstrap

## 2 Листинг

### urls.py

```
from django.conf.urls import url
from testApp import views

urlpatterns = [
    url(r'^$', views.home),
    url(r'^departments$', views.DepartmentsView.as_view(),
name='departments_url'),
    url(r'^department/(?P<id>\d+)$',
views.DepartmentView.as_view(), name='department_url'),
]
```

### views.py

```
from django.shortcuts import render
import django.views.generic

# Create your views here.
def home(request):
    par = {
        'header': 'Home'
    }
    return render(request, 'home.html', context=par)

class DepartmentsView(django.views.generic.View):
    def get(self, request):
        data = {
            'departments': [
                {'title': 'Отдел маркетинга', 'id': 1},
                {'title': 'Бухгалтерия', 'id': 2}
            ]
        }
```

```

    }
    return render(request, 'departments.html', data)

class DepartmentView(django.views.generic.View):
    def get(self, request, id):
        data = {
            'department':
                {
                    'id': id
                }
        }
        return render(request, 'department.html', data)

```

## base.html

```

{% load static %}
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/html"
xmlns="http://www.w3.org/1999/html">
<head>
    <meta charset="UTF-8">
    <title>{% block title %}{% endblock %}</title>
    <link rel="stylesheet" type="text/css" href="{% static
'css/bootstrap.css' %}">
</head>
<body>
<nav class="navbar navbar-expand-sm bg-primary navbar-dark">
    <ul class="navbar-nav">
        <li class="nav-item">
            <a class="nav-link" href="/">Главная</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="/departments">Отделы</a>
        </li>
    </ul>
</nav>
<div>
    {% block content %}{% endblock %}
</div>
</body>
</html>

```

## home.html

```

{% extends 'base.html' %}

{% block title %}home{% endblock %}

{% block content %}
    <h1>{{ header }}</h1>
{% endblock %}

```

## departments.html

```

{% extends 'base.html' %}

{% block title %}Отделы{% endblock %}

{% block content %}
    <div class="jumbotron">
        <div class="container">

```

```

<div class="page-header">
  <h1>Отделы</h1>
</div>
<ul class="list-group">
  {% for department in departments %}
    {% include 'department_element.html' with
element=department %}
    {% empty %}
      <li>пустой список</li>
    {% endfor %}
  </ul>
</div>
</div>
{% endblock %}

```

### department.html

```

{% extends 'base.html' %}

{% block title %}Отдел № {{ department.id }}{% endblock %}

{% block content %}
  <div>
    <p>Страница отдела № {{ department.id }}</p>
  </div>
{% endblock %}

```

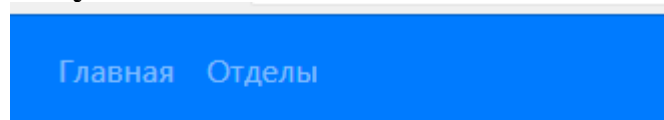
### department\_element.html

```

<li class="list-group-item"><a href="{% url 'department_url' element.id
%}">{{ element.title }}</a> </li>

```

## Результат



# Home

