Факультатив ФКН

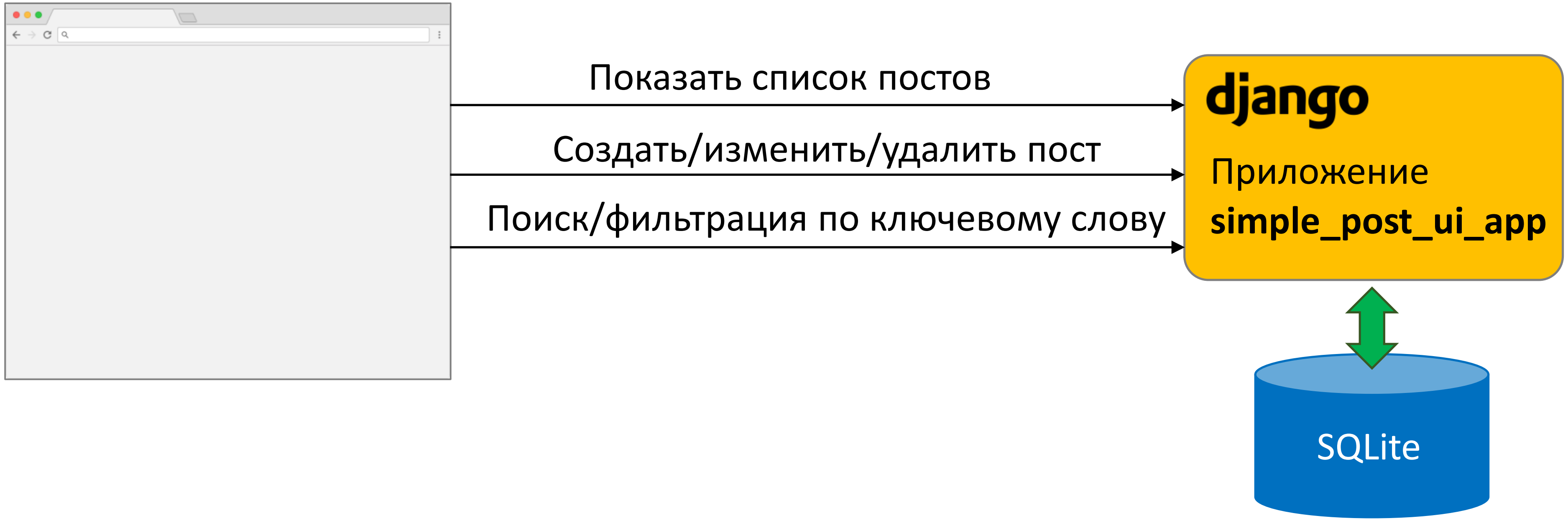
РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ НА БАЗЕ ПЛАТФОРМЫ TIZEN

Преподаватель
Папулин Сергей Юрьевич (***papulin_hse@mail.ru***)

Москва, 2018

Django приложение SimplePost

Основные функции



Показать список постов

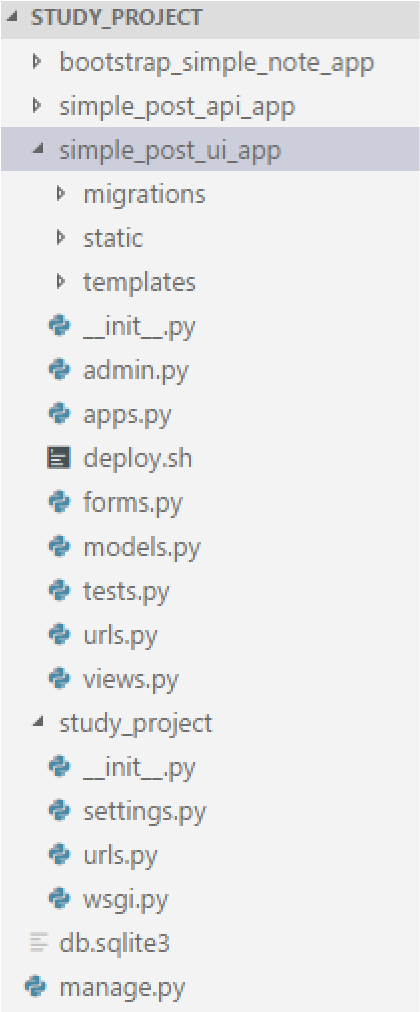
Создать/изменить/удалить пост

Поиск/фильтрация по ключевому слову

django

Приложение **simple_post_ui_app**

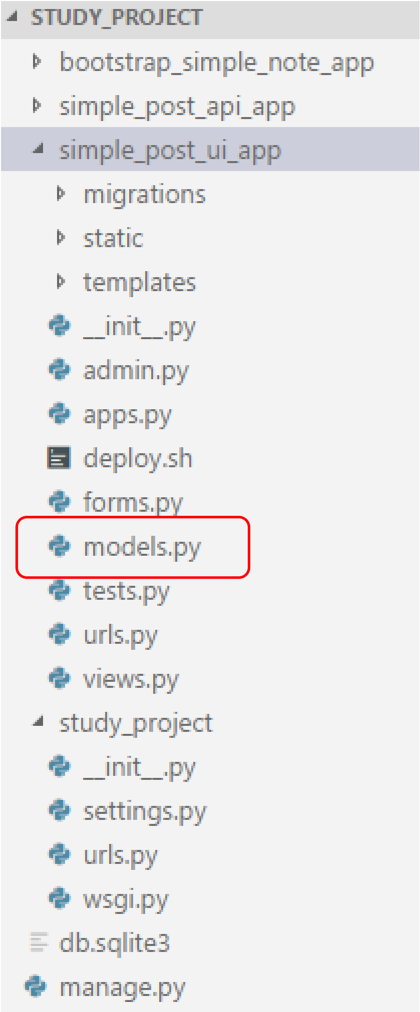
SQLite



```
STUDY_PROJECT
  bootstrap_simple_note_app
  simple_post_api_app
  simple_post_ui_app
    migrations
    static
    templates
    __init__.py
    admin.py
    apps.py
    deploy.sh
    forms.py
    models.py
    tests.py
    urls.py
    views.py
  study_project
    __init__.py
    settings.py
    urls.py
    wsgi.py
  db.sqlite3
  manage.py
```

Модель данных

```python
class SimplePost(models.Model):

    # The id field will be auto-created

    header = models.CharField(max_length=200, blank=False, null=False)
    content = models.TextField(blank=False)
    created = models.DateTimeField(auto_now_add=True)
    updated = models.DateTimeField(auto_now=True)

    def get_absolute_url(self):
        from django.urls import reverse
        return reverse("testapp_simple-post-list", kwargs={"pk": self.id})
```



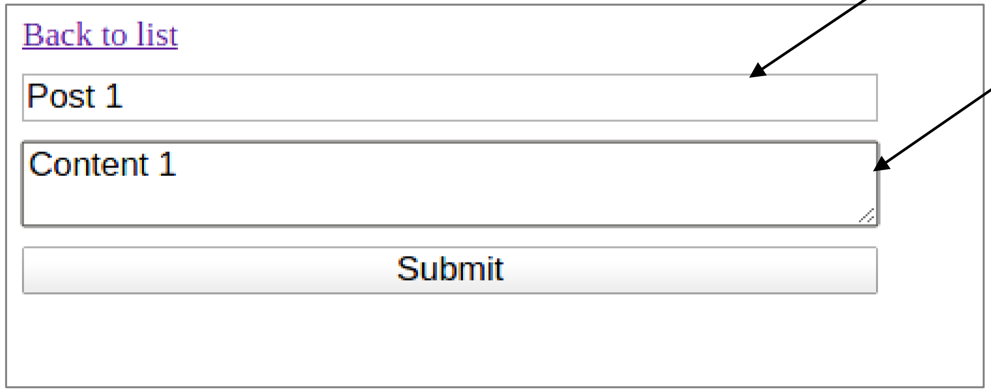
```
STUDY_PROJECT
  bootstrap_simple_note_app
  simple_post_api_app
  simple_post_ui_app
    migrations
    static
    templates
    __init__.py
    admin.py
    apps.py
    deploy.sh
    forms.py
    models.py
    tests.py
    urls.py
    views.py
  study_project
    __init__.py
    settings.py
    urls.py
    wsgi.py
  db.sqlite3
  manage.py
```

Создать пост

View. Создать пост

Браузер

GET
http://127.0.0.1:8000/simple-post-ui/create/



Back to list

Post 1

Content 1

header

content

Создать пост

POST
http://127.0.0.1:8000/simple-post-ui/create/

Перенаправление на список всех постов

Показать список постов

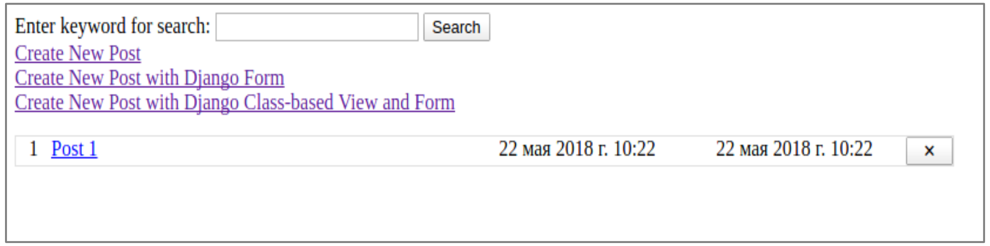
http://127.0.0.1:8000/simple-post-ui/

HTML страница

django

Приложение
simple_post_ui_app

http://127.0.0.1:8000/simple-post-ui/



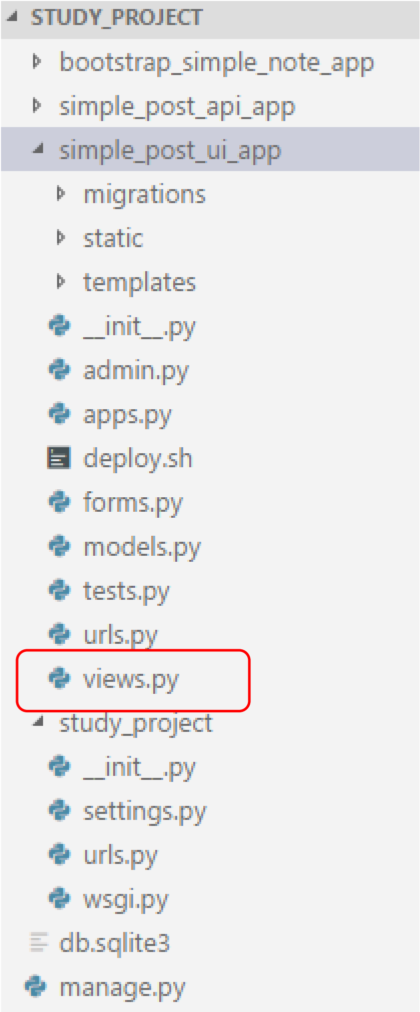
Enter keyword for search:

Create New Post
Create New Post with Django Form
Create New Post with Django Class-based View and Form

1	Post 1	22 мая 2018 г. 10:22	22 мая 2018 г. 10:22	×

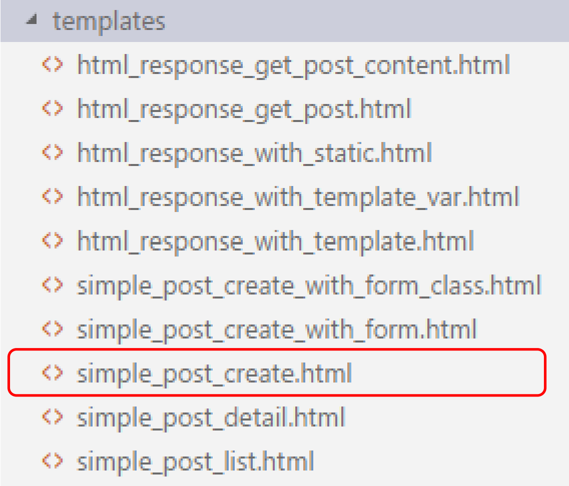
View. Создать пост

```
from simple_post_ui_app.models import SimplePost


# create
def create_simple_post(request):

    if request.method == "POST":

        header = request.POST.get("header", None)
        content = request.POST.get("content", None)

    if header is not None and content is not None:
        #obj = SimplePost.objects.create(header=header, content=content) # in one step
        obj = SimplePost(header=header, content=content)
        obj.save()
        return redirect("testapp_simple-post-list")

    if request.method == "GET":
        return render(request, "simple_post_ui_app/templates/simple_post_create.html")

    return HttpResponseNotFound("<h1>Unable to perform</h1>")
```

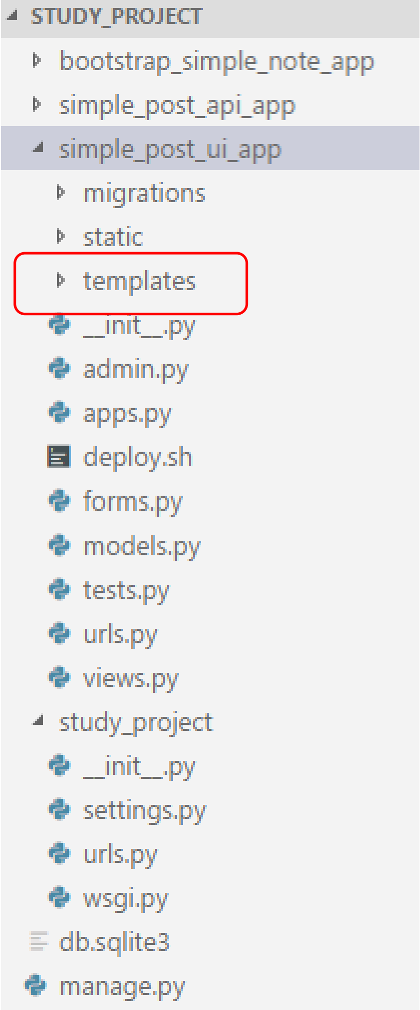


- STUDY_PROJECT
  - bootstrap_simple_note_app
  - simple_post_api_app
  - simple_post_ui_app
    - migrations
    - static
    - templates
    - __init__.py
    - admin.py
    - apps.py
    - deploy.sh
    - forms.py
    - models.py
    - tests.py
    - urls.py
    - views.py
  - study_project
    - __init__.py
    - settings.py
    - urls.py
    - wsgi.py
  - db.sqlite3
  - manage.py

Template. Создать пост



- templates
  - html_response_get_post_content.html
  - html_response_get_post.html
  - html_response_with_static.html
  - html_response_with_template_var.html
  - html_response_with_template.html
  - simple_post_create_with_form_class.html
  - simple_post_create_with_form.html
  - simple_post_create.html
  - simple_post_detail.html
  - simple_post_list.html

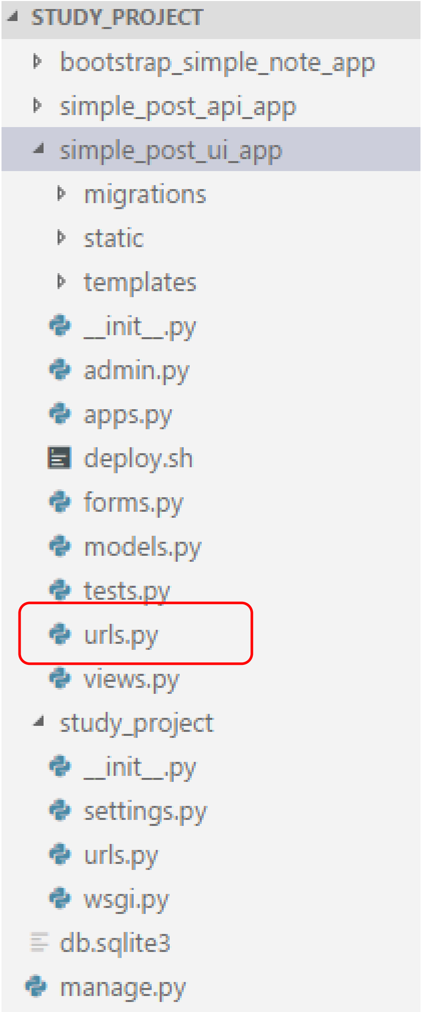


- STUDY_PROJECT
  - bootstrap_simple_note_app
  - simple_post_api_app
  - simple_post_ui_app
    - migrations
    - static
    - templates
    - __init__.py
    - admin.py
    - apps.py
    - deploy.sh
    - forms.py
    - models.py
    - tests.py
    - urls.py
    - views.py
  - study_project
    - __init__.py
    - settings.py
    - urls.py
    - wsgi.py
  - db.sqlite3
  - manage.py

```
<!DOCTYPE html>
<html lang="en">
<head>
    {% load static %}
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet" href="{% static 'simple_post_ui_app/css/style.css' %}">
</head>
<body>
    <a href="{% url 'testapp_simple-post-list' %}">Back to list</a>
    <form class="create" method="post" action="{% url 'testapp_create-simple-post' %}">
        {% csrf_token %}
        <input type="text" placeholder="Header" name="header">
        <textarea placeholder="Content" name="content"></textarea>
        <button type="submit">Submit</button>
    </form>
</body>
</html>
```

Url. Создать пост

```
# create
url(    r'^create/$’,
        views.create_simple_post,
        name="testapp_create-simple-post"),
```

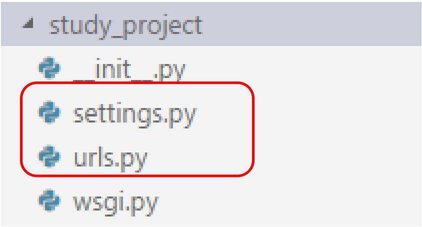


STUDY_PROJECT
bootstrap_simple_note_app
simple_post_api_app
simple_post_ui_app
migrations
static
templates
__init__.py
admin.py
apps.py
deploy.sh
forms.py
models.py
tests.py
urls.py
views.py
study_project
__init__.py
settings.py
urls.py
wsgi.py
db.sqlite3
manage.py

Settings и urls

settings.py

```
INSTALLED_APPS = [
    …
    "simple_post_ui_app",
    …
]
```



```
◢ study_project
    __init__.py
    settings.py
    urls.py
    wsgi.py
```

urls.py

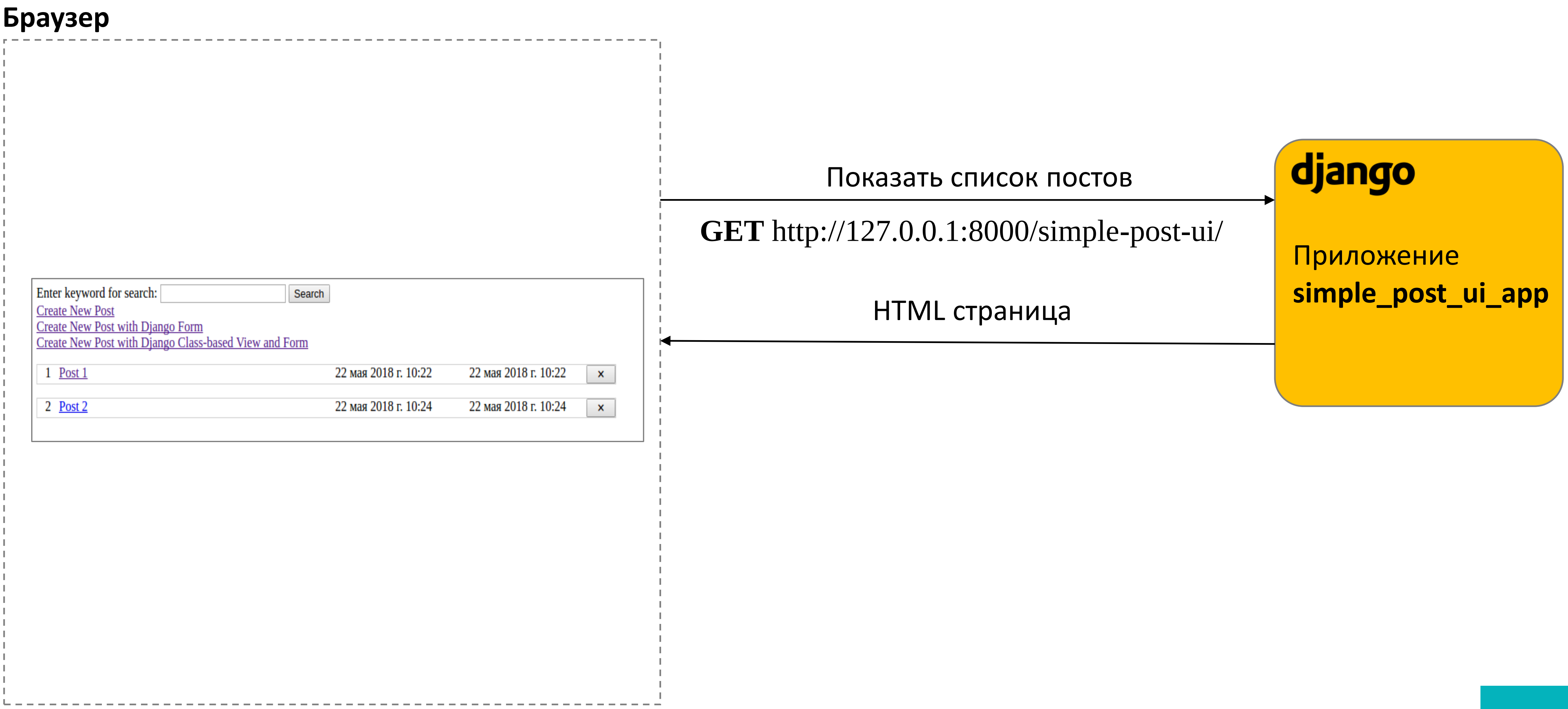
```
urlpatterns = [
        url(r'^simple-post-ui/', include('simple_post_ui_app.urls')),
]
```

terminal

```
python manage.py makemigrations simple_post_ui_app

python manage.py migrate simple_post_ui_app

python manage.py runserver
```

Показать все посты

View. Показать все посты



Браузер

Enter keyword for search: Search

Create New Post

Create New Post with Django Form

Create New Post with Django Class-based View and Form

1	Post 1	22 мая 2018 г. 10:22	22 мая 2018 г. 10:22	×
2	Post 2	22 мая 2018 г. 10:24	22 мая 2018 г. 10:24	×

Показать список постов

GET http://127.0.0.1:8000/simple-post-ui/

HTML страница

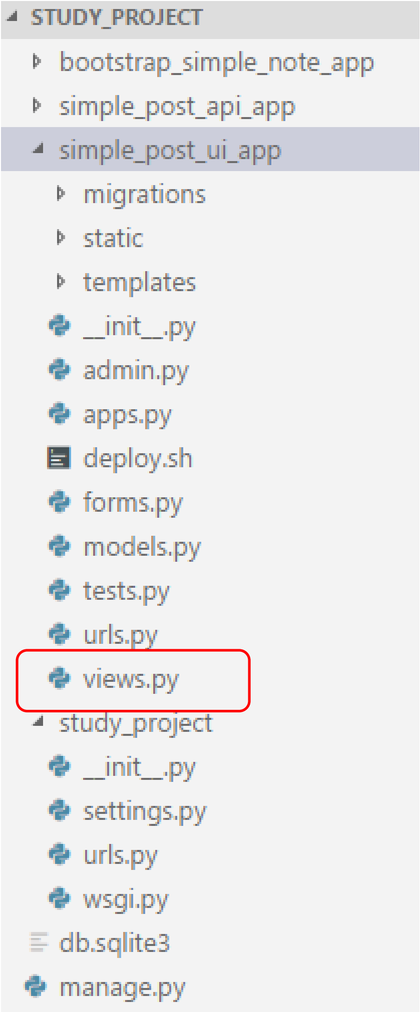
django

Приложение

simple_post_ui_app

View. Показать все посты

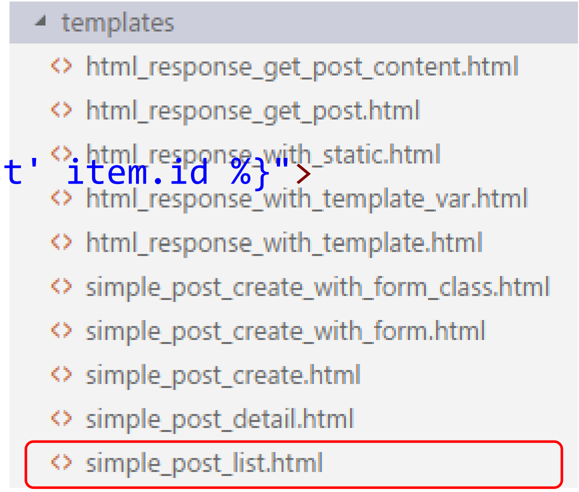
```
# select
def simple_post_list_view(request, pk=None):
    if request.method == "GET" and pk is None:
        objs = SimplePost.objects.all()
        context_var = {"simple_post_objects": objs}
        return render(request,
        "simple_post_ui_app/templates/simple_post_list.html", context_var)

    if request.method == "GET" and pk is not None:
        obj = SimplePost.objects.get(id=pk)
        context_var = {"simple_post_object": obj}
        return render(request,
        "simple_post_ui_app/templates/simple_post_detail.html", context_var)

    return HttpResponseNotFound("<h1>Unable to perform</h1>")
```



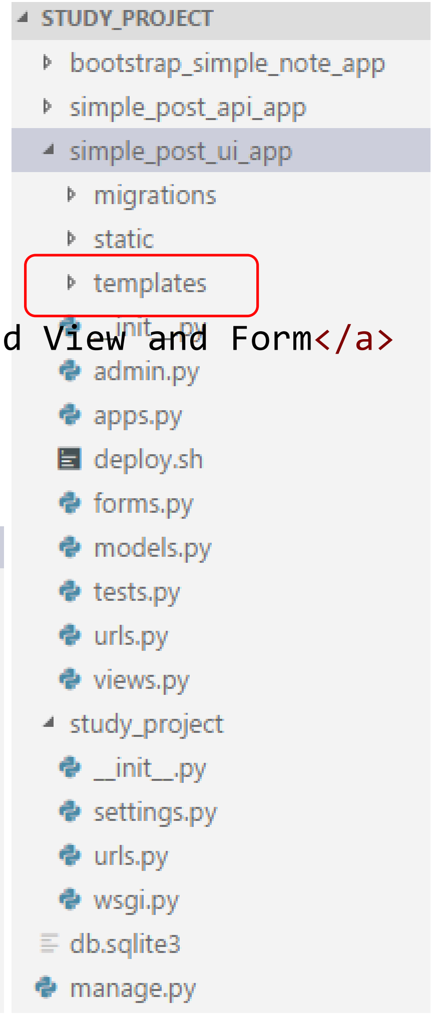
```
STUDY_PROJECT
  bootstrap_simple_note_app
  simple_post_api_app
  simple_post_ui_app
    migrations
    static
    templates
    __init__.py
    admin.py
    apps.py
    deploy.sh
    forms.py
    models.py
    tests.py
    urls.py
    views.py
  study_project
    __init__.py
    settings.py
    urls.py
    wsgi.py
  db.sqlite3
  manage.py
```

Template. Показать все посты

```
<form class="search" method="get" action="{% url 'testapp_filter-simple-post' %}">
    <label>Enter keyword for search: </label>
    <input type="text" name="keyword">
    <button type="submit">Search</button>
</form>
<a href="{% url 'testapp_create-simple-post' %}">Create New Post</a><br>
<a href="{% url 'testapp_create-simple-post-form' %}">Create New Post with Django Form</a><br>
<a href="{% url 'testapp_create-simple-post-form-class' %}">Create New Post with Django Class-based View and Form</a>
<div class="simple-post-items">
    {% for item in simple_post_objects %}
    <div class="simple-post-item">
        <span>{{ forloop.counter }}</span>
        <a href="{{ item.get_absolute_url }}">{{ item.header }}</a>
        <span>{{ item.created }}</span>
        <span>{{ item.updated }}</span>
        <form method="post" action="{% url 'testapp_delete-simple-post' item.id %}">
            {% csrf_token %}
            <input type="hidden" name="id" value="{{ item.id }}">
            <button type="submit">&#10005;</button>
        </form>
    </div>
    {% empty %}
    There are no posts
    {% endfor %}
</div>
```



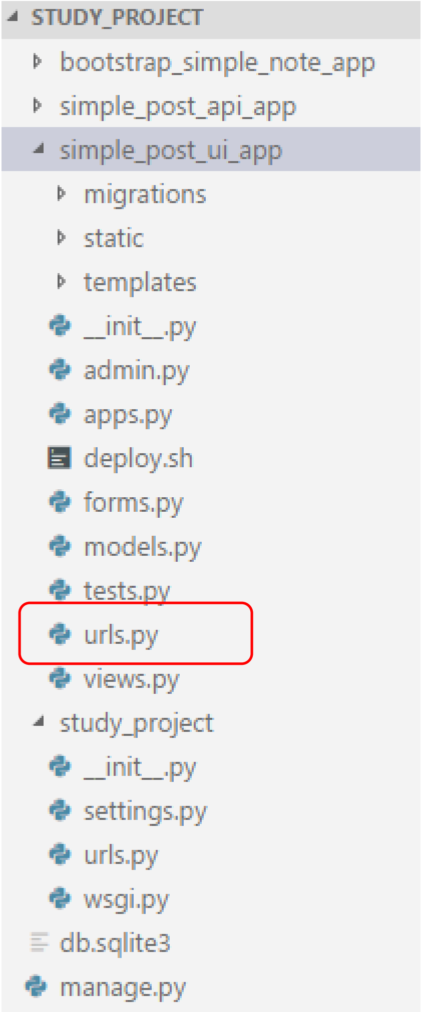
- templates
  - html_response_get_post_content.html
  - html_response_get_post.html
  - html_response_with_static.html
  - html_response_with_template_var.html
  - html_response_with_template.html
  - simple_post_create_with_form_class.html
  - simple_post_create_with_form.html
  - simple_post_create.html
  - simple_post_detail.html
  - simple_post_list.html



- STUDY_PROJECT
  - bootstrap_simple_note_app
  - simple_post_api_app
  - simple_post_ui_app
    - migrations
    - static
    - templates
    - __init__.py
    - admin.py
    - apps.py
    - deploy.sh
    - forms.py
    - models.py
    - tests.py
    - urls.py
    - views.py
  - study_project
    - __init__.py
    - settings.py
    - urls.py
    - wsgi.py
  - db.sqlite3
  - manage.py

Url. Показать все посты

```
# select all
url(    r'^$’,
        views.simple_post_list_view,
        name="testapp_simple-post-list"),
```

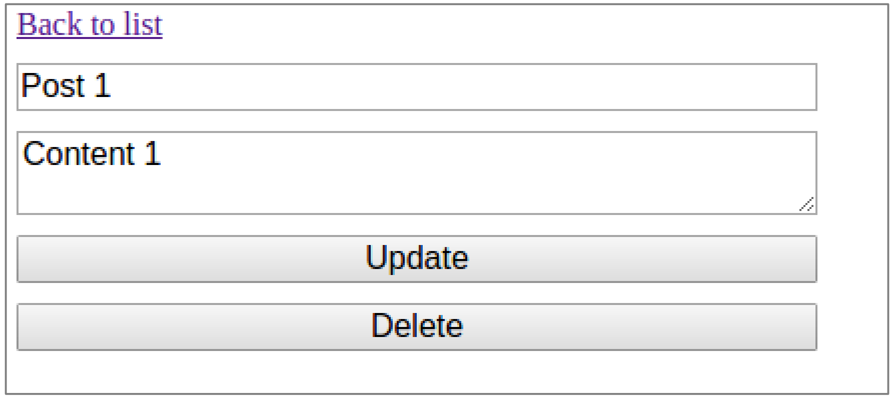


```
STUDY_PROJECT
  bootstrap_simple_note_app
  simple_post_api_app
  simple_post_ui_app
    migrations
    static
    templates
    __init__.py
    admin.py
    apps.py
    deploy.sh
    forms.py
    models.py
    tests.py
    urls.py
    views.py
  study_project
    __init__.py
    settings.py
    urls.py
    wsgi.py
  db.sqlite3
  manage.py
```

Изменить пост

View. Изменить пост

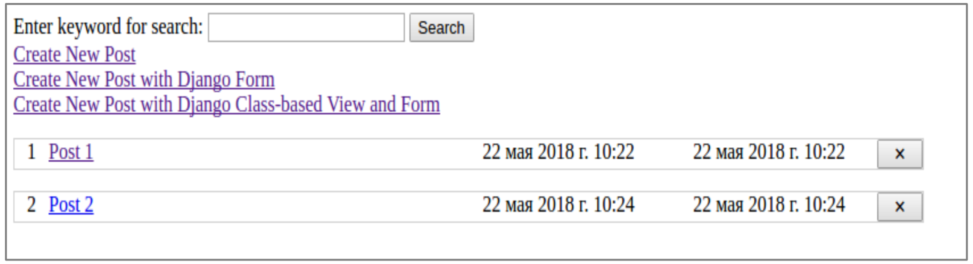
Браузер



Back to list

Post 1

Content 1



Enter keyword for search:

Create New Post

Create New Post with Django Form

Create New Post with Django Class-based View and Form

1	Post 1	22 мая 2018 г. 10:22	22 мая 2018 г. 10:22	×
2	Post 2	22 мая 2018 г. 10:24	22 мая 2018 г. 10:24	×

Показать пост

Primary Key

GET http://127.0.0.1:8000/simple-post-ui/**pk**/

HTML страница

POST

http://127.0.0.1:8000/simple-post-ui/**pk**/update/

Перенаправление на список всех постов

Показать список постов

http://127.0.0.1:8000/simple-post-ui/

HTML страница

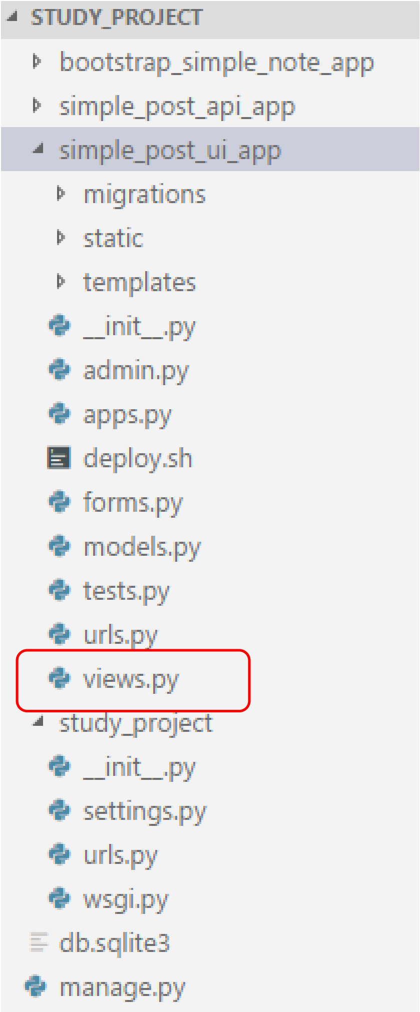
django

Приложение

simple_post_ui_app

View. Изменить пост

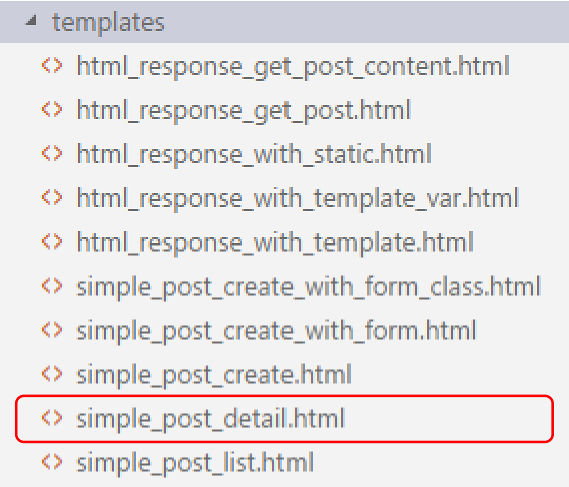
```python
# update
def update_simple_post(request, pk=None):

    if request.method == "POST":

        id = request.POST.get("id", None)
        header = request.POST.get("header", None)
        content = request.POST.get("content", None)

        if header is not None and content is not None and id is not None:
            try:
                obj = SimplePost.objects.get(pk=id)
                obj.header = header
                obj.content = content
                obj.save(update_fields=["header", "content", "updated"])
            except SimplePost.DoesNotExist:
                raise Http404("There is no such id")

        return redirect("testapp_simple-post-list")

    return HttpResponseNotFound("<h1>Unable to perform</h1>")
```



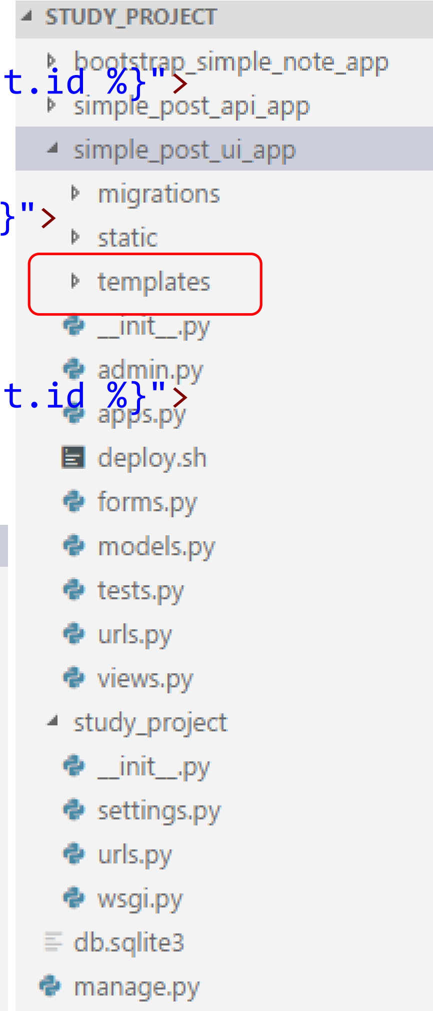
```
◢ STUDY_PROJECT
  ▷ bootstrap_simple_note_app
  ▷ simple_post_api_app
  ◢ simple_post_ui_app
    ▷ migrations
    ▷ static
    ▷ templates
      __init__.py
      admin.py
      apps.py
      deploy.sh
      forms.py
      models.py
      tests.py
      urls.py
      views.py
  ◢ study_project
      __init__.py
      settings.py
      urls.py
      wsgi.py
    db.sqlite3
    manage.py
```

Template. Изменить пост

```
<body>
    <a href="{% url 'testapp_simple-post-list' %}">Back to list</a>
    <form class="detail" method="post" action="{% url 'testapp_update-simple-post' simple_post_object.id %}">
        {% csrf_token %}
        <input type="hidden" name="id" value="{{ simple_post_object.id }}">
        <input type="text" placeholder="Header" name="header" value="{{ simple_post_object.header }}">
        <textarea placeholder="Content" name="content">{{ simple_post_object.content }}</textarea>
        <button type="submit">Update</button>
    </form>
    <form class="detail" method="post" action="{% url 'testapp_delete-simple-post' simple_post_object.id %}">
        {% csrf_token %}
        <input type="hidden" name="id" value="{{ simple_post_object.id }}">
        <button type="submit">Delete</button>
    </form>
</body>
```



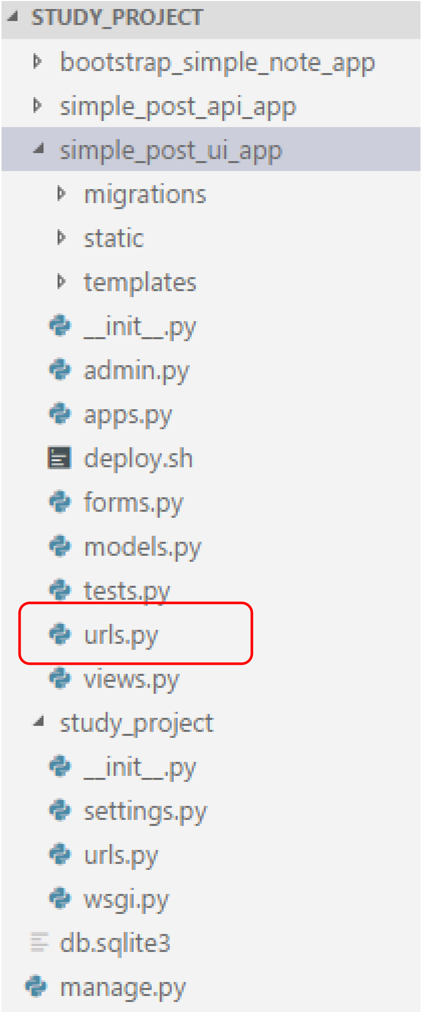
templates
html_response_get_post_content.html
html_response_get_post.html
html_response_with_static.html
html_response_with_template_var.html
html_response_with_template.html
simple_post_create_with_form_class.html
simple_post_create_with_form.html
simple_post_create.html
simple_post_detail.html
simple_post_list.html



STUDY_PROJECT
bootstrap_simple_note_app
simple_post_api_app
simple_post_ui_app
migrations
static
templates
__init__.py
admin.py
apps.py
deploy.sh
forms.py
models.py
tests.py
urls.py
views.py
study_project
__init__.py
settings.py
urls.py
wsgi.py
db.sqlite3
manage.py

Url. Изменить пост

```
# select by id
url(    r'^(?P<pk>\d+)/$’,
        views.simple_post_list_view,
        name="testapp_simple-post-list"),

# update
url(    r'^(?P<pk>\d+)/update$’,
        views.update_simple_post,
        name="testapp_update-simple-post"),
```

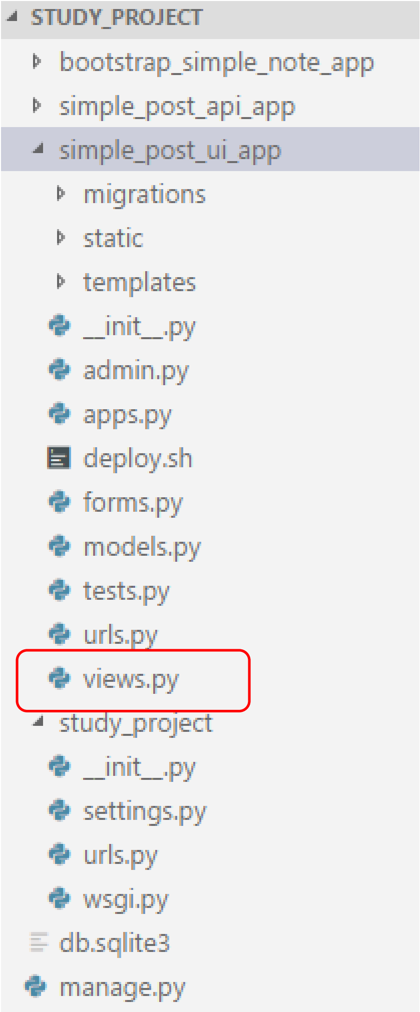


- STUDY_PROJECT
  - bootstrap_simple_note_app
  - simple_post_api_app
  - simple_post_ui_app
    - migrations
    - static
    - templates
    - __init__.py
    - admin.py
    - apps.py
    - deploy.sh
    - forms.py
    - models.py
    - tests.py
    - urls.py
    - views.py
  - study_project
    - __init__.py
    - settings.py
    - urls.py
    - wsgi.py
  - db.sqlite3
  - manage.py

Удалить пост

View. Удалить пост

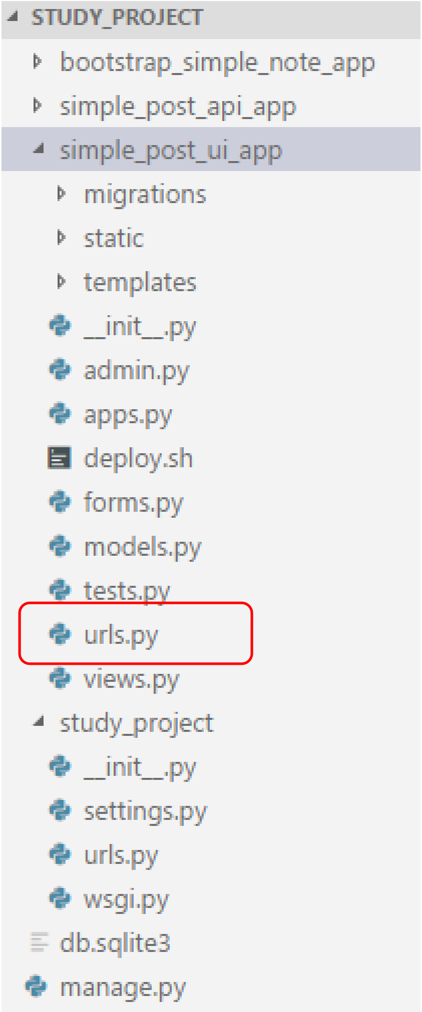
```
# delete
def delete_simple_post(request, pk=None):

    if request.method == "POST":

        id = request.POST.get("id", None)

        if id is not None:
            try:
                obj = SimplePost.objects.get(pk=id)
                obj.delete()
            except SimplePost.DoesNotExist:
                raise Http404("There is no such id")

        return redirect("testapp_simple-post-list")

    return HttpResponseNotFound("<h1>Unable to perform</h1>")
```



```
STUDY_PROJECT
  bootstrap_simple_note_app
  simple_post_api_app
  simple_post_ui_app
    migrations
    static
    templates
    __init__.py
    admin.py
    apps.py
    deploy.sh
    forms.py
    models.py
    tests.py
    urls.py
    views.py
  study_project
    __init__.py
    settings.py
    urls.py
    wsgi.py
  db.sqlite3
  manage.py
```

Url. Удалить пост

```
# delete
url(    r'^(?P<pk>\d+)/delete’,
        views.delete_simple_post,
        name="testapp_delete-simple-post"),
```

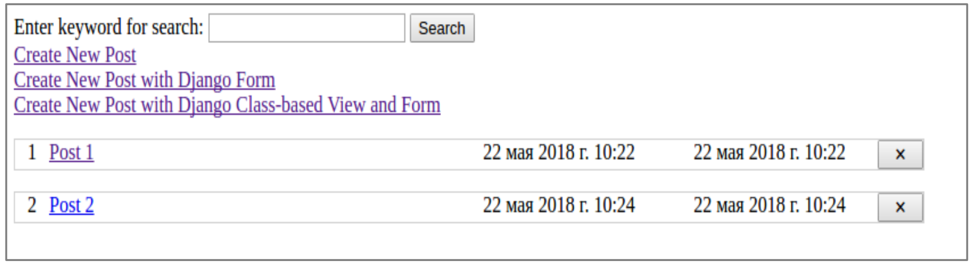


```
◢ STUDY_PROJECT
  ▷ bootstrap_simple_note_app
  ▷ simple_post_api_app
  ◢ simple_post_ui_app
    ▷ migrations
    ▷ static
    ▷ templates
      __init__.py
      admin.py
      apps.py
      deploy.sh
      forms.py
      models.py
      tests.py
      urls.py
      views.py
  ◢ study_project
      __init__.py
      settings.py
      urls.py
      wsgi.py
    db.sqlite3
    manage.py
```

Поиск по ключевому слову

View. Поиск по ключевому слову в заголовке

Браузер



Enter keyword for search: Search

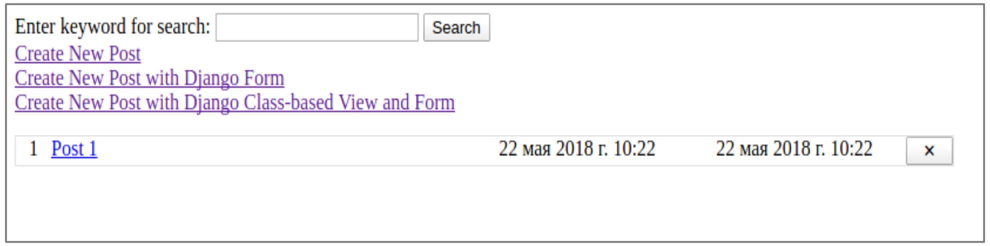
Create New Post

Create New Post with Django Form

Create New Post with Django Class-based View and Form

1 Post 1 22 мая 2018 г. 10:22 22 мая 2018 г. 10:22 ×

2 Post 2 22 мая 2018 г. 10:24 22 мая 2018 г. 10:24 ×



Enter keyword for search: Search

Create New Post

Create New Post with Django Form

Create New Post with Django Class-based View and Form

1 Post 1 22 мая 2018 г. 10:22 22 мая 2018 г. 10:22 ×

Показать список постов

GET

/simple-post-ui/filter/?keyword=1

HTML страница

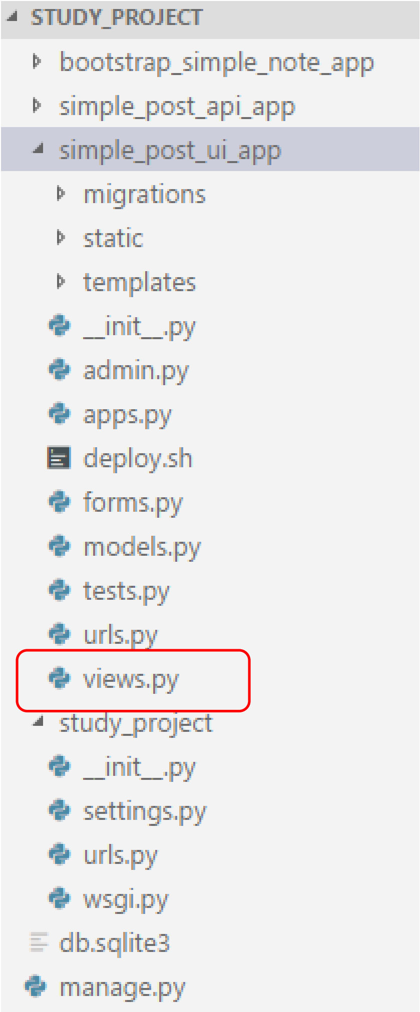
django

Приложение

simple_post_ui_app

View. Поиск по ключевому слову в заголовке

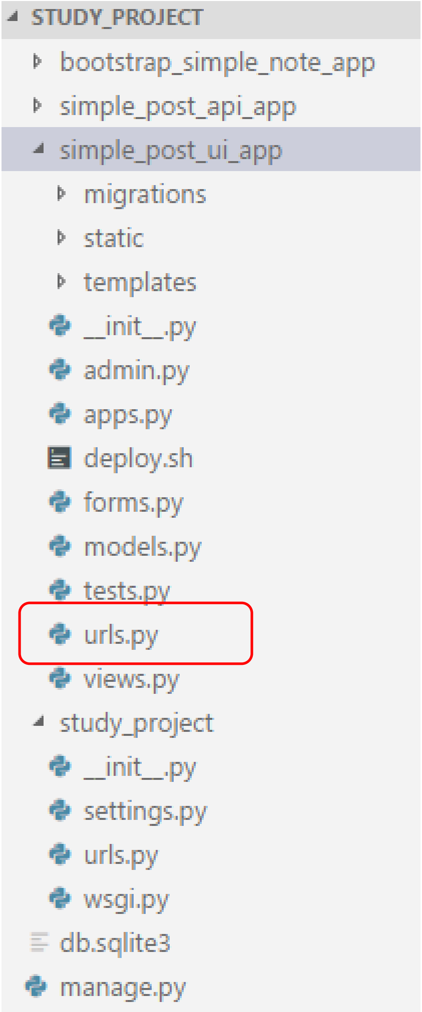
```python
# filter
def filter_simple_posts(request):

    key_word = request.GET.get("keyword", "")

    objs = SimplePost.objects.filter(header__icontains=key_word)
    context_var = {"simple_post_objects": objs}

    return render(request,
    "simple_post_ui_app/templates/simple_post_list.html", context_var)
```



```
STUDY_PROJECT
  bootstrap_simple_note_app
  simple_post_api_app
  simple_post_ui_app
    migrations
    static
    templates
    __init__.py
    admin.py
    apps.py
    deploy.sh
    forms.py
    models.py
    tests.py
    urls.py
    views.py
  study_project
    __init__.py
    settings.py
    urls.py
    wsgi.py
  db.sqlite3
  manage.py
```

Url. Поиск по ключевому слову в заголовке

```
# filter
url(    r'^filter/$’,
        views.filter_simple_posts,
        name="testapp_filter-simple-post"),
```

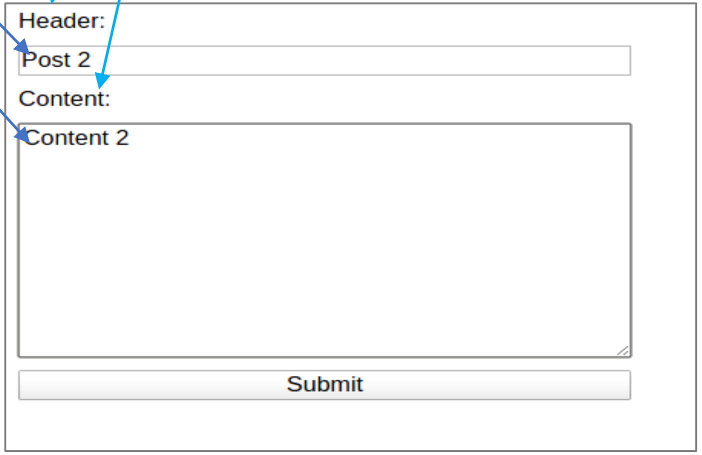


```
STUDY_PROJECT
  bootstrap_simple_note_app
  simple_post_api_app
  simple_post_ui_app
    migrations
    static
    templates
    __init__.py
    admin.py
    apps.py
    deploy.sh
    forms.py
    models.py
    tests.py
    urls.py
    views.py
  study_project
    __init__.py
    settings.py
    urls.py
    wsgi.py
  db.sqlite3
  manage.py
```

Использование Django Form

Form. Создать пост

```
class SimplePostForm(forms.Form):

    header = forms.CharField(label="Header", max_length=200, required=True)
    content = forms.CharField(label="Content", widget=forms.Textarea)


class SimplePostModelForm(forms.ModelForm):

    class Meta:
        model = SimplePost
        fields = ["id", "header", "content"]
```

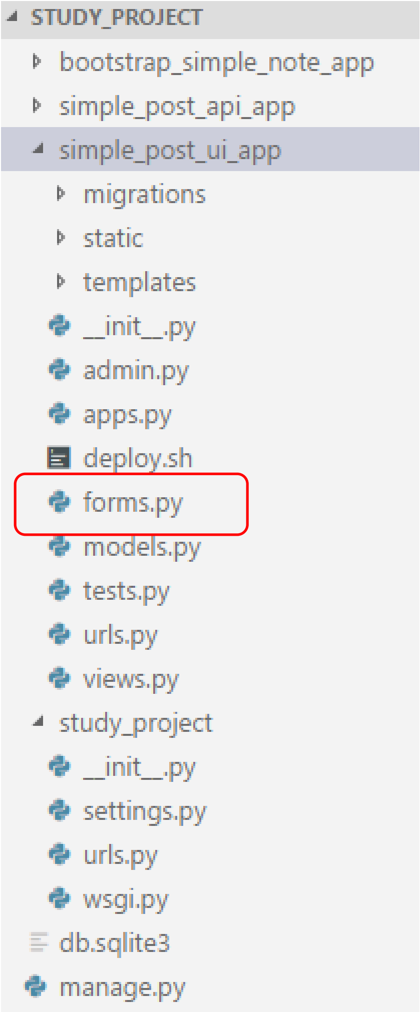


Header:

Post 2

Content:

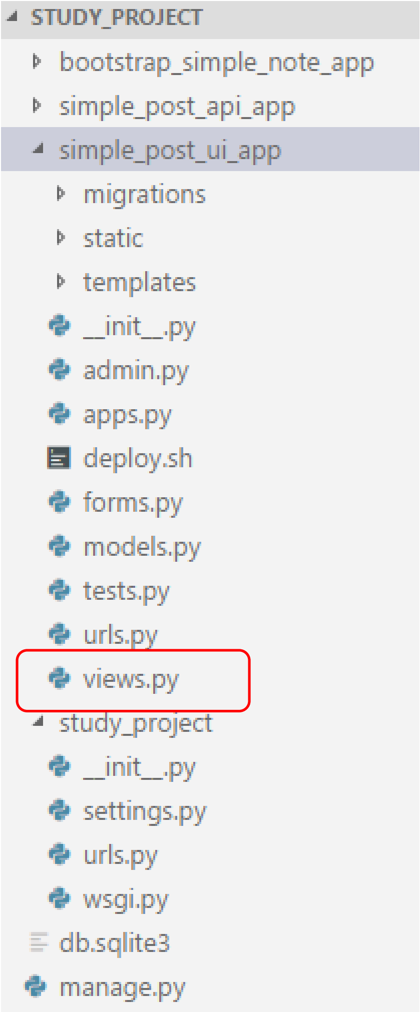
Content 2



```
◢ STUDY_PROJECT
  ▷ bootstrap_simple_note_app
  ▷ simple_post_api_app
  ◢ simple_post_ui_app
    ▷ migrations
    ▷ static
    ▷ templates
      __init__.py
      admin.py
      apps.py
      deploy.sh
      forms.py
      models.py
      tests.py
      urls.py
      views.py
  ◢ study_project
      __init__.py
      settings.py
      urls.py
      wsgi.py
    db.sqlite3
    manage.py
```

View. Создать пост

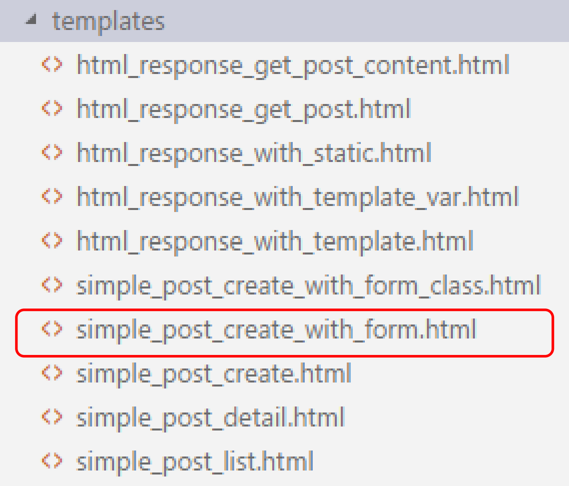
```python
def simple_post_form(request):

    if request.method == "POST":
        form = SimplePostForm(request.POST)
        if form.is_valid():
            SimplePost.objects.create(header=form.cleaned_data["header"],
            content=form.cleaned_data["content"])
            return redirect("testapp_simple-post-list")

    if request.method == "GET":
        form = SimplePostForm()
        return render(request,
        "simple_post_ui_app/templates/simple_post_create_with_form.html",
        {'form': form})
```



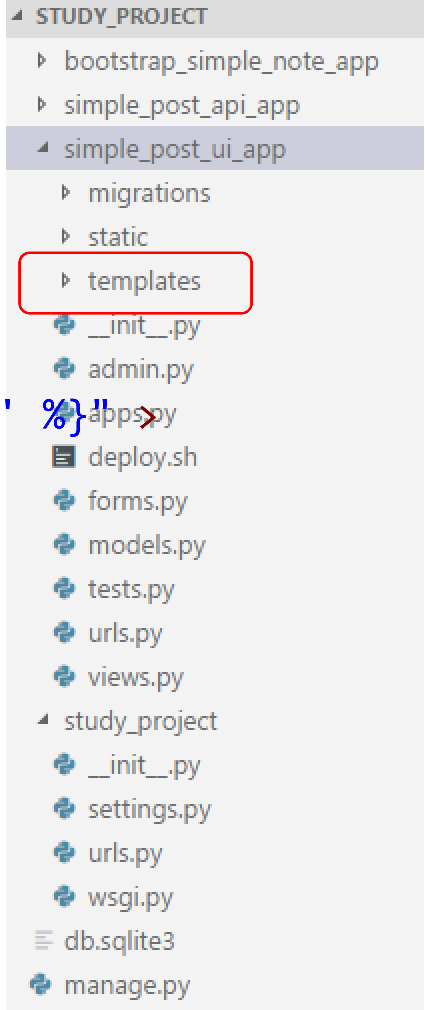
- STUDY_PROJECT
  - bootstrap_simple_note_app
  - simple_post_api_app
  - simple_post_ui_app
    - migrations
    - static
    - templates
    - __init__.py
    - admin.py
    - apps.py
    - deploy.sh
    - forms.py
    - models.py
    - tests.py
    - urls.py
    - views.py
  - study_project
    - __init__.py
    - settings.py
    - urls.py
    - wsgi.py
  - db.sqlite3
  - manage.py

Template. Создать пост

```
<!DOCTYPE html>
<html lang="en">
<head>
    {% load static %}
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet" href="{% static 'simple_post_ui_app/css/style.css' %}">
</head>
<body>
    <form class="create" method="post" action="{% url 'testapp_create-simple-post-form' %}">
    {% csrf_token %}
    {{ form }}
    <input type="submit" value="Submit" />
    </form>
</body>
</html>
```



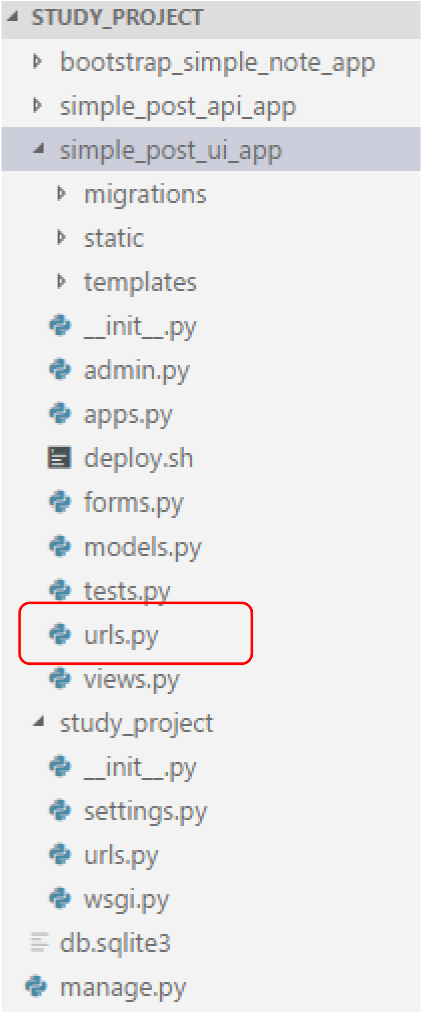
```
templates
    html_response_get_post_content.html
    html_response_get_post.html
    html_response_with_static.html
    html_response_with_template_var.html
    html_response_with_template.html
    simple_post_create_with_form_class.html
    simple_post_create_with_form.html
    simple_post_create.html
    simple_post_detail.html
    simple_post_list.html
```



```
STUDY_PROJECT
  bootstrap_simple_note_app
  simple_post_api_app
  simple_post_ui_app
    migrations
    static
    templates
    __init__.py
    admin.py
    apps.py
    deploy.sh
    forms.py
    models.py
    tests.py
    urls.py
    views.py
  study_project
    __init__.py
    settings.py
    urls.py
    wsgi.py
  db.sqlite3
  manage.py
```

Url. Создать пост

```
# ==================================
# Forms
# ==================================
# create
url(   r'^create-simple-post-form/create/$’,
       views.simple_post_form,
       name="testapp_create-simple-post-form"),
```

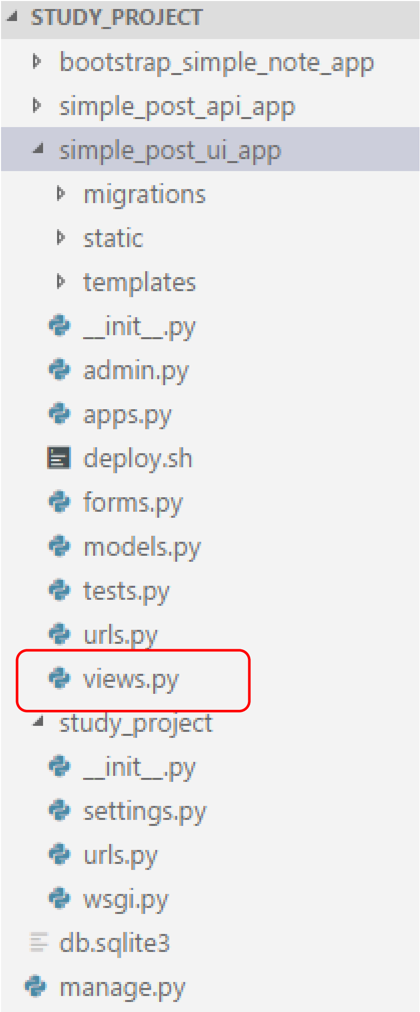


```
STUDY_PROJECT
  bootstrap_simple_note_app
  simple_post_api_app
  simple_post_ui_app
    migrations
    static
    templates
    __init__.py
    admin.py
    apps.py
    deploy.sh
    forms.py
    models.py
    tests.py
    urls.py
    views.py
  study_project
    __init__.py
    settings.py
    urls.py
    wsgi.py
  db.sqlite3
  manage.py
```

Class-based View

Class-based View. Создать пост

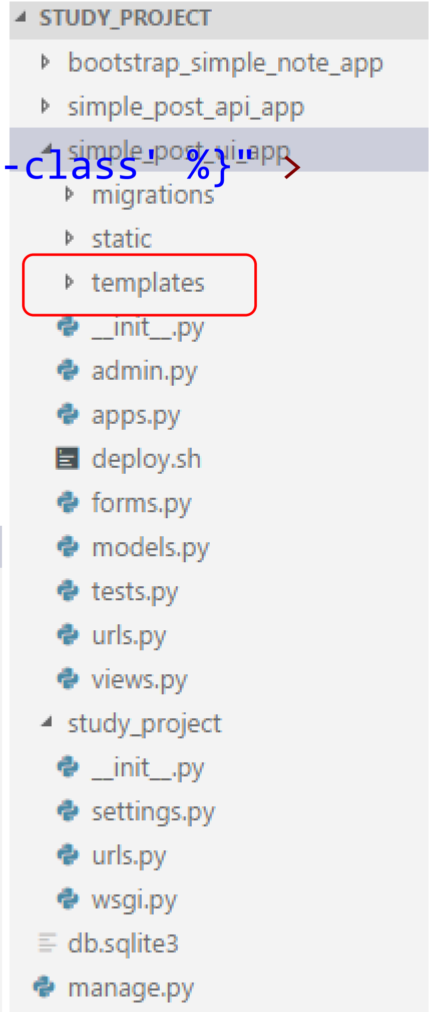
```
class SimplePostView(View):

    def get(self, request):
        form = SimplePostModelForm()
        return render(request,
        "simple_post_ui_app/templates/simple_post_create_with_form_class.ht
        ml", {'form': form})

    def post(self, request):

        form = SimplePostModelForm(request.POST)

        if form.is_valid():
            form.save()
            return redirect("testapp_simple-post-list")

        return render(request,
        "simple_post_ui_app/templates/simple_post_create_with_form_class.ht
        ml", {'form': form})
```



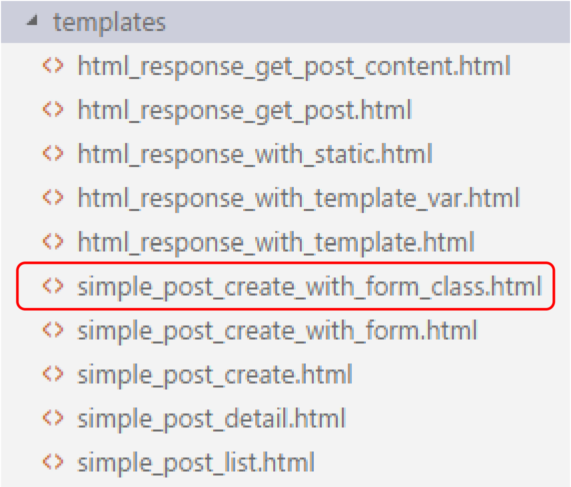
STUDY_PROJECT
- bootstrap_simple_note_app
- simple_post_api_app
- simple_post_ui_app
  - migrations
  - static
  - templates
  - __init__.py
  - admin.py
  - apps.py
  - deploy.sh
  - forms.py
  - models.py
  - tests.py
  - urls.py
  - views.py
- study_project
  - __init__.py
  - settings.py
  - urls.py
  - wsgi.py
- db.sqlite3
- manage.py

Template. Создать пост

```
<body>
    <form class="create" method="post" action="{% url 'testapp_create-simple-post-form-class' %}" >
    {% csrf_token %}
    {{ form }}
    <input type="submit" value="Submit" />
    </form>
</body>
```



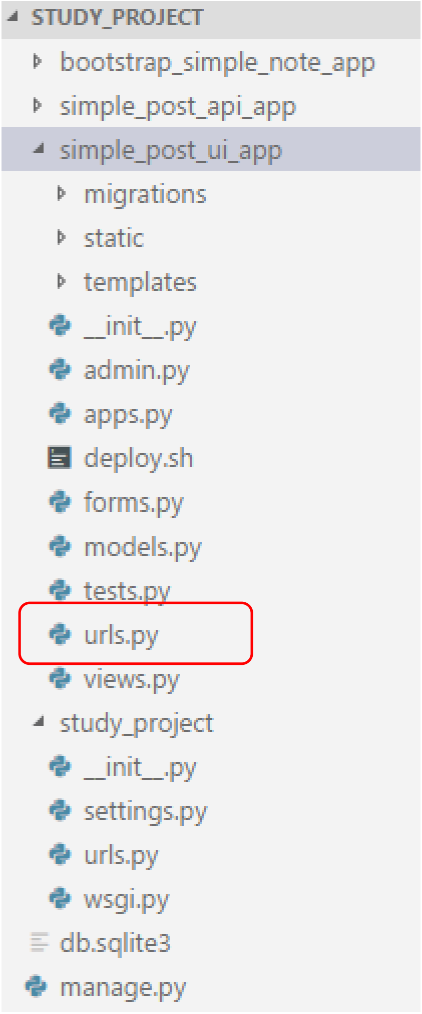
```
◢ STUDY_PROJECT
  ▷ bootstrap_simple_note_app
  ▷ simple_post_api_app
  ◢ simple_post_ui_app
    ▷ migrations
    ▷ static
    ▷ templates
      __init__.py
      admin.py
      apps.py
      deploy.sh
      forms.py
      models.py
      tests.py
      urls.py
      views.py
  ◢ study_project
      __init__.py
      settings.py
      urls.py
      wsgi.py
    db.sqlite3
    manage.py
```



```
◢ templates
  <> html_response_get_post_content.html
  <> html_response_get_post.html
  <> html_response_with_static.html
  <> html_response_with_template_var.html
  <> html_response_with_template.html
  <> simple_post_create_with_form_class.html
  <> simple_post_create_with_form.html
  <> simple_post_create.html
  <> simple_post_detail.html
  <> simple_post_list.html
```

Url. Создать пост

```
# ===================================
# Class-based View
# ===================================
# create
url(    r'^create-simple-post-from-class/create/$’,
        views.SimplePostView.as_view(),
        name="testapp_create-simple-post-form-class"),
```



```
STUDY_PROJECT
  bootstrap_simple_note_app
  simple_post_api_app
  simple_post_ui_app
    migrations
    static
    templates
    __init__.py
    admin.py
    apps.py
    deploy.sh
    forms.py
    models.py
    tests.py
    urls.py
    views.py
  study_project
    __init__.py
    settings.py
    urls.py
    wsgi.py
  db.sqlite3
  manage.py
```

Источники

https://docs.djangoproject.com/en/2.0/