

Факультатив ФКН

# РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ НА БАЗЕ ПЛАТФОРМЫ



Преподаватель

Папулин Сергей Юрьевич ([papulin\\_hse@mail.ru](mailto:papulin_hse@mail.ru))



# Основные компоненты Django на простых примерах



## Пример 1. HttpResponse



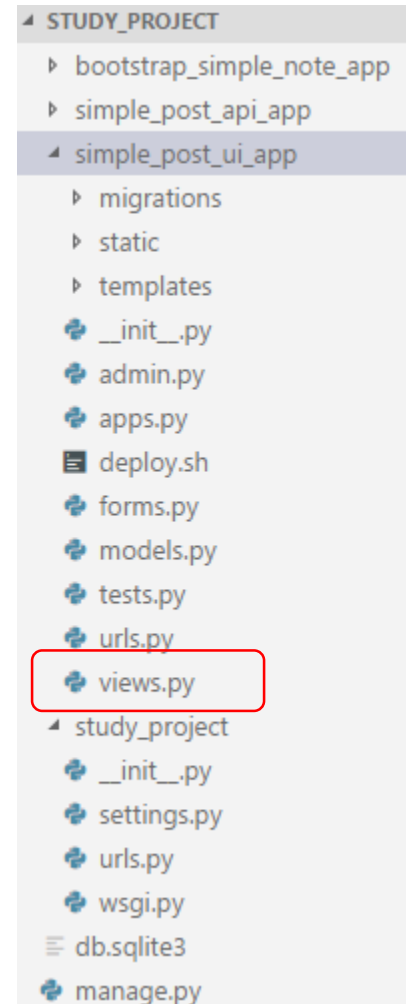
# View. Response

# View 1: HttpResponse

```
def view_http_response(request):  
    html_string = "<html><body><h1>Header 1</h1><p>Text 1</p></body></html>"  
    return HttpResponse(html_string)
```

# View 2: HttpResponse with variables

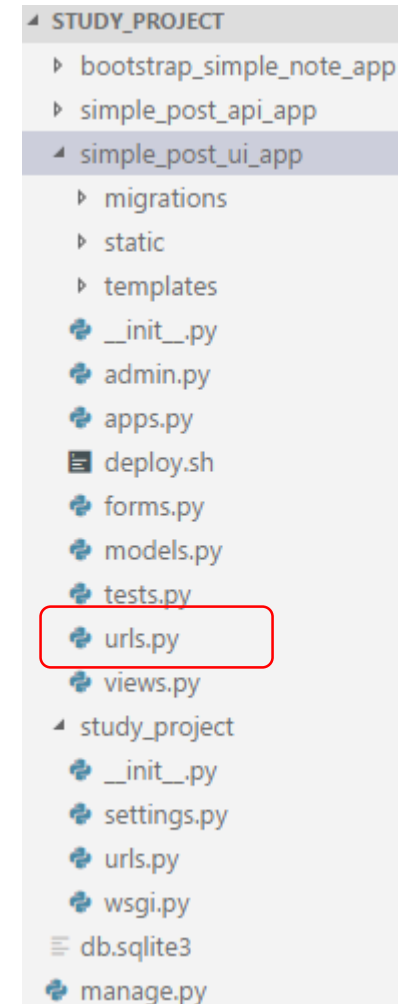
```
def view_http_response_with_var(request):  
    header = "Header 2"  
    content = "Text 2"  
    html_string = "<html><body><h1>%s</h1><p>%s</p></body></html>" % (header,  
content)  
    return HttpResponse(html_string)
```





# Url

```
urlpatterns = [  
  
    # view 1  
    url(    r'^http-response/$',  
           views.view_http_response,  
           name="testapp_view-http-response"),  
  
    # view 2  
    url(    r'^http-response_with_var/$',  
           views.view_http_response_with_var,  
           name="testapp_view-http-response-with-var"),  
  
    ...
```





## Пример 2. Использование шаблонов



# View. Response, Template

```
# View 3: HttpResponse with a template
```

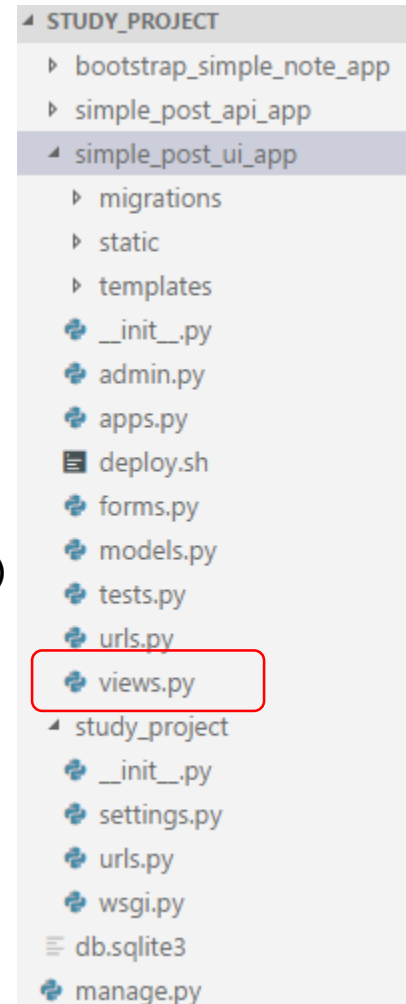
```
from django.template import loader
```

```
def view_http_response_with_template(request):
```

```
    template =
```

```
    loader.get_template("simple_post_ui_app/templates/html_response_with_template.html")
```

```
    return HttpResponse(template.render())
```





# Template

## 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

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Title</title>
  </head>
  <body>
    <h1>Header 3</h1>
    <p>Text 3</p>
  </body>
</html>
```

## STUDY\_PROJECT

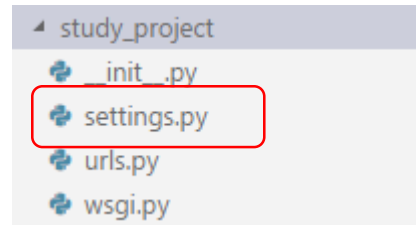
- bootstrap\_simple\_note\_app
- simple\_post\_api\_app
- simple\_post\_ui\_app
  - migrations
  - static
  - templates
- \_\_init\_\_.py
- admin.py
- apps.py
- deploy.sh
- forms.py
- models.py
- tests.py
- urls.py
- views.py
- study\_project
  - \_\_init\_\_.py
  - settings.py
  - urls.py
  - wsgi.py
- db.sqlite3
- manage.py





# Template

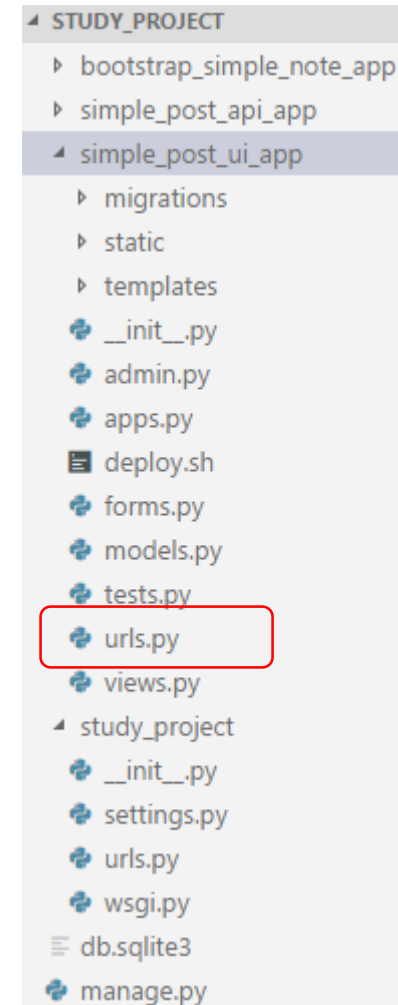
```
TEMPLATES = [  
    {  
        'BACKEND':  
        'django.template.backends.django.DjangoTemplates',  
        'DIRS': ['./'],  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.debug',  
                'django.template.context_processors.i18n',  
                'django.template.context_processors.request',  
                'django.contrib.auth.context_processors.auth',  
                'django.contrib.messages.context_processors.messages'  
            ],  
        },  
    },  
]
```





# Url

```
# view 3
url(    r'^http-response_with_template/$',
        views.view_http_response_with_template,
        name="testapp_view-http-response-with-template"),
```





## Пример 3. Использование переменных в шаблоне



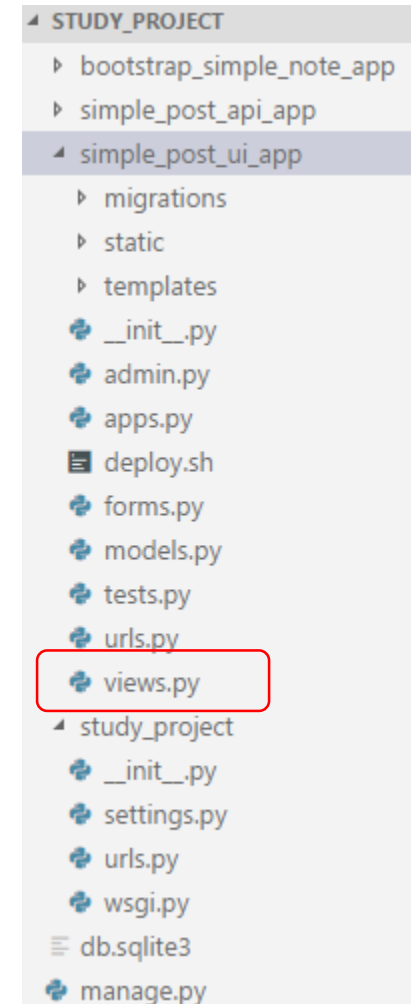
# View. Response, Context, Template

# View 4: HttpResponse with a template and variables

```
def view_http_response_with_template_var(request):  
  
    # comment: add the "./" path to the template DIRS in the settings.py  
    template =  
        loader.get_template("simple_post_ui_app/templates/html_response_with_template_var.html")  
  
    context_var = {"header": "Header 4", "content": "Text 4"}  
  
    return HttpResponse(template.render(context_var, request))
```

# View 5: HttpResponse with a template and variables. Approach 2 (main)

```
def view_http_response_with_template_var_2(request):  
  
    context_var = {"header": "Header 5", "content": "Text 5"}  
  
    return render(request,  
        "simple_post_ui_app/templates/html_response_with_template_var.html",  
        context_var)
```





# Template

## 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

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Title</title>
  </head>
  <body>
    <h1>{{ header }}</h1>
    <p>{{ content }}</p>
  </body>
</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



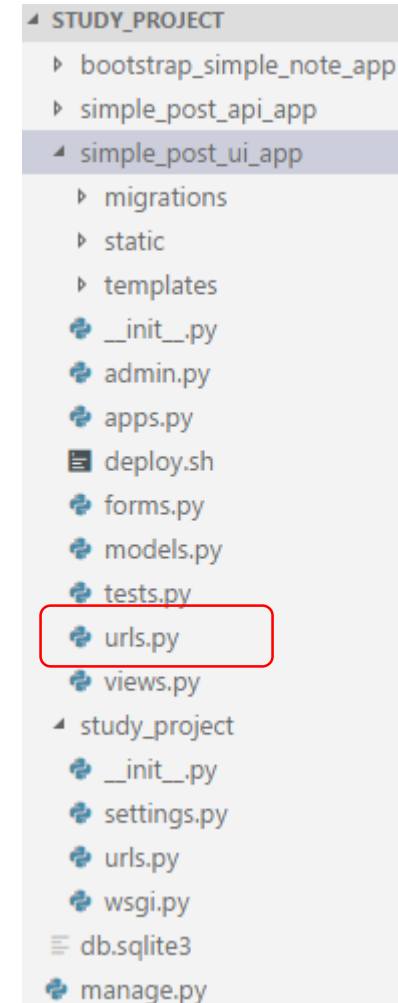
# Url

# view 4

```
url(    r'^http-response_with_template_var/$',  
      views.view_http_response_with_template_var,  
      name="testapp_view-http-response-with-template-var"),
```

# view 5

```
url(    r'^http-response_with_template_var_2/$',  
      views.view_http_response_with_template_var_2,  
      name="testapp_view-http-response-with-template-var-2"),
```



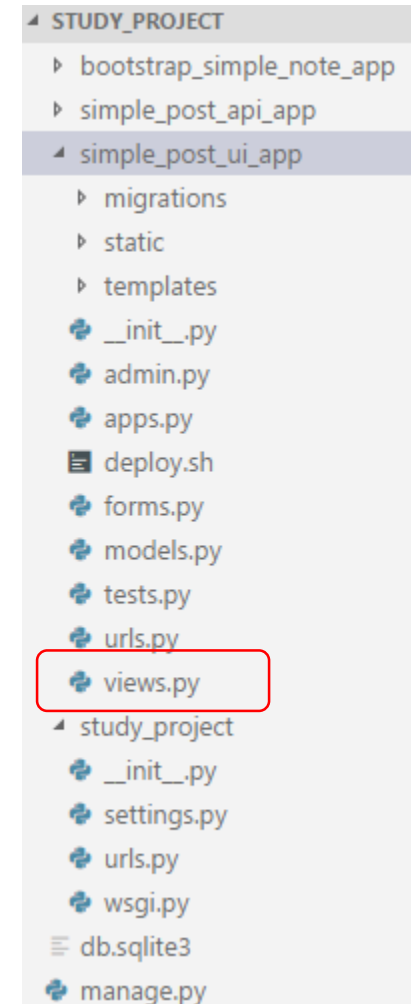


## Пример 4. Перенаправление с HttpResponseRedirect



# View. Redirect

```
# View 6: HttpResponseRedirect
def view_http_redirect(request):
    return HttpResponseRedirect("http://mail.ru", permanent=False) #True - HTTP
    Response Code 301; False - 302
    #return HttpResponseRedirect("/simple_post_ui_app/view-http-response/")
    #return HttpResponseRedirect(reverse("testapp_view-http-response"))
    #return redirect("testapp_view-http-response")
```

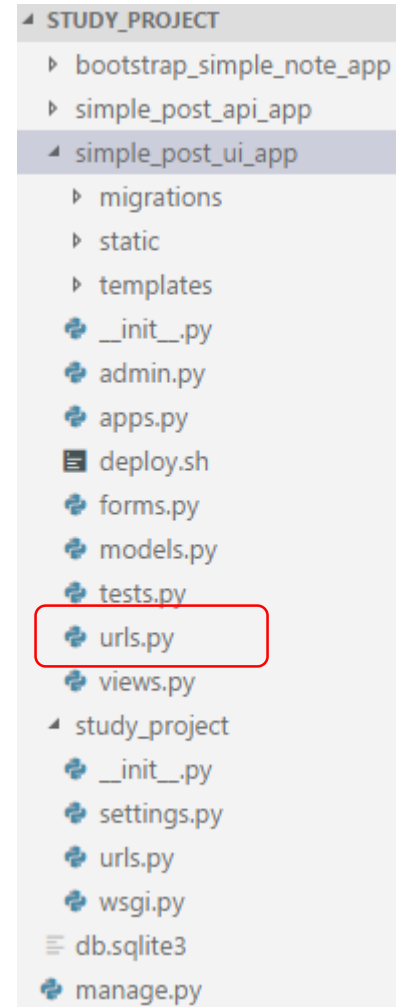






# Url

```
# view 6
url(    r'^http-redirect/$',
        views.view_http_redirect,
        name="testapp_view-http-redirect"),
```





## Пример 5. JSON ответ



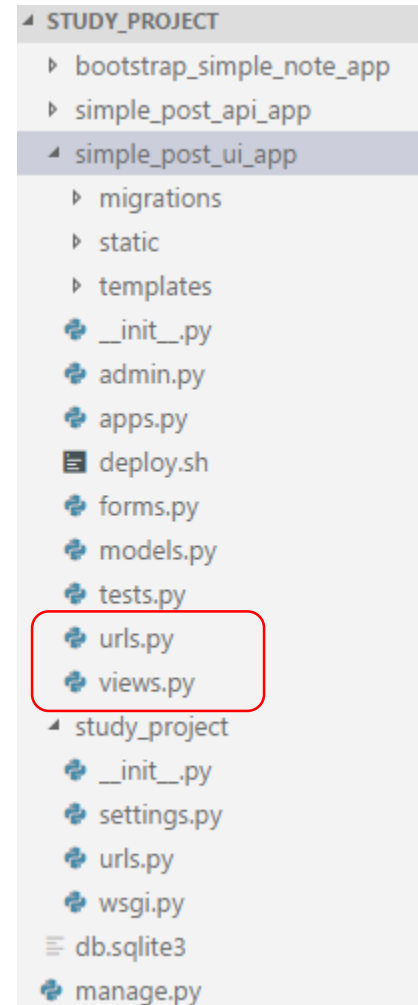
# View. Redirect

```
# View 7: JsonResponse
def view_json_response(request):

    data = {"header": "Header 7", "content": "Text 7"}

    return JsonResponse(data)
```

```
# view 7
url(    r'^json-response/$',
        views.view_json_response,
        name="testapp_view-json-response"),
```





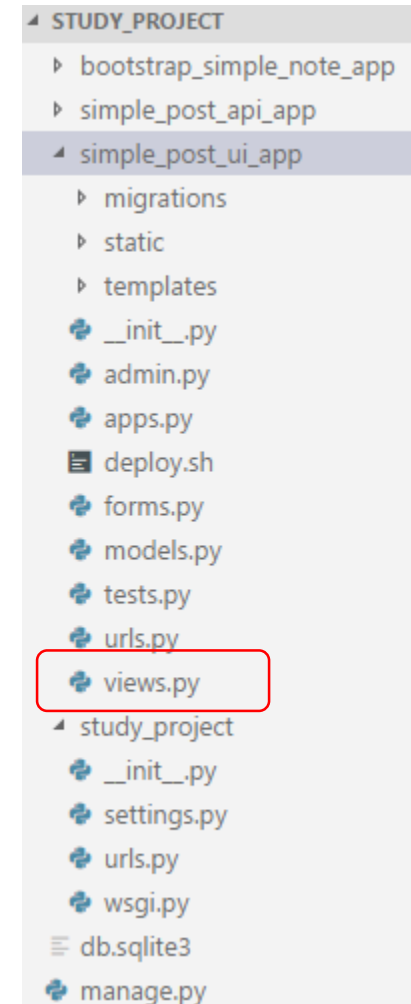
Пример 6. Шаблон со статическими файлами (css, js и пр.)



# View. Response, Context, Template

# View 8:

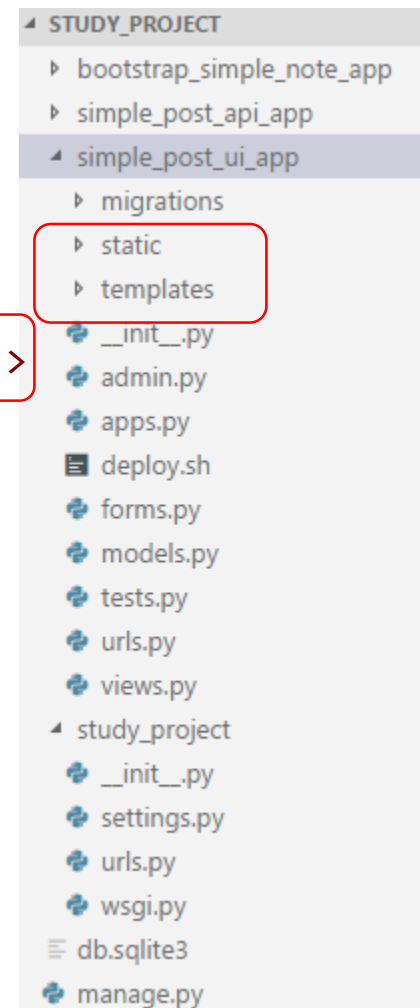
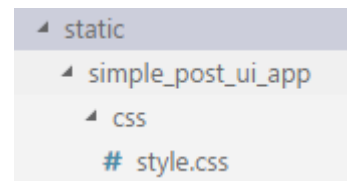
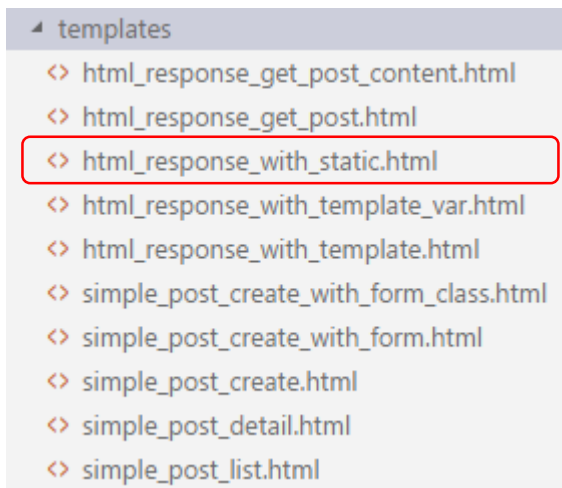
```
def view_http_response_with_static(request):  
    #print(STATIC_ROOT)  
    context_var = {"header": "Header 5", "content": "Text 5"}  
  
    return render(request,  
        "simple_post_ui_app/templates/html_response_with_static.html", context_var)
```





# Template

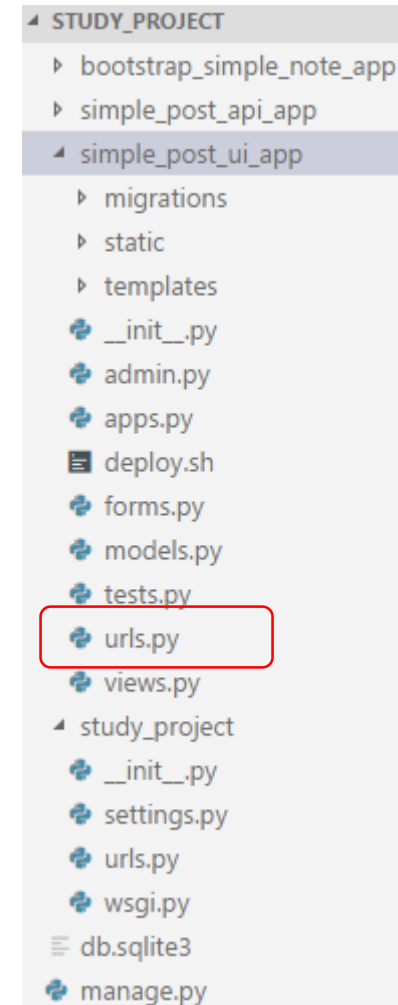
```
<!DOCTYPE html>
<html lang="en">
  <head>
    {% load static %}
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet" href="{% static 'simple_post_ui_app/css/style.css' %}">
  </head>
  <body>
    <h1>Header 3</h1>
    <p>Text 3</p>
  </body>
</html>
```





# Url

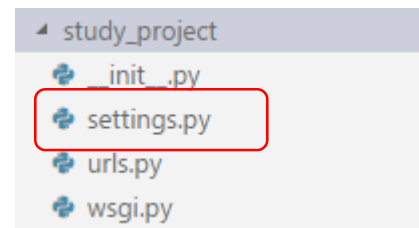
```
# view 8
url(    r'^http-response-with-static/$',
        views.view_http_response_with_static,
        name="testapp_view-http-response-with-static"),
```





## Указание директории для css и js файлов

```
STATIC_URL = '/static/'  
STATICFILES_DIRS = [  
    os.path.join(BASE_DIR, "simple_post_ui_app/static"),  
]
```







## Пример 7. Передача параметров пользователя в View

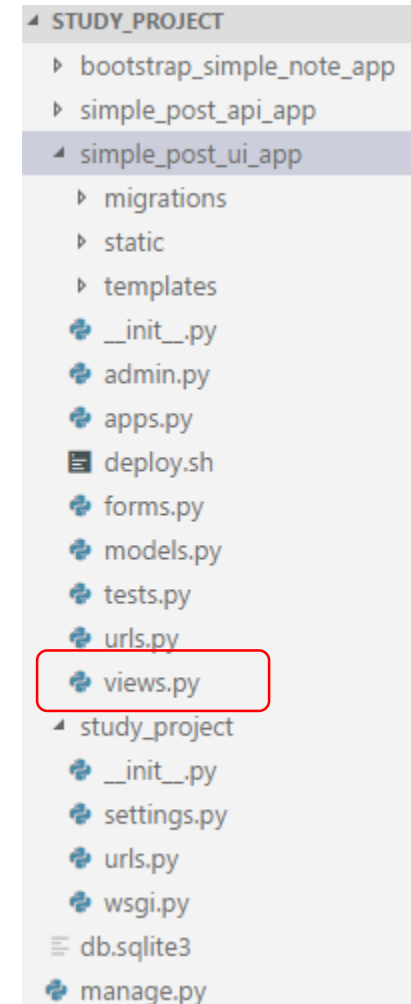
Способ 1. args



# View. Response, Context, Template

## # 9. View Args

```
def view_http_response_url_args(request, *args):  
  
    header = "Default Header"  
    content = "Default Content"  
  
    if len(args) > 0:  
        header = args[0]  
    if len(args) > 1:  
        content = args[1]  
  
    context_var = {"header": header, "content": content}  
  
    return render(request,  
        "simple_post_ui_app/templates/html_response_with_template_var.html",  
        context_var)
```





# Url

# view 9

```
url( r'^http-response-url-args/$',  
      views.view_http_response_url_args,  
      name="testapp_view-http-response-url-args"),  
  
url( r'^http-response-url-args/([+ \w]+)/$',  
      views.view_http_response_url_args,  
      name="testapp_view-http-response-url-args"),  
  
url( r'^http-response-url-args/([+ \w]+)/([+ \w]+)/$',  
      views.view_http_response_url_args,  
      name="testapp_view-http-response-url-args"),
```

Header

Header

Content

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



## Пример 7. Передача параметров пользователя в View

### Способ 2. kwargs



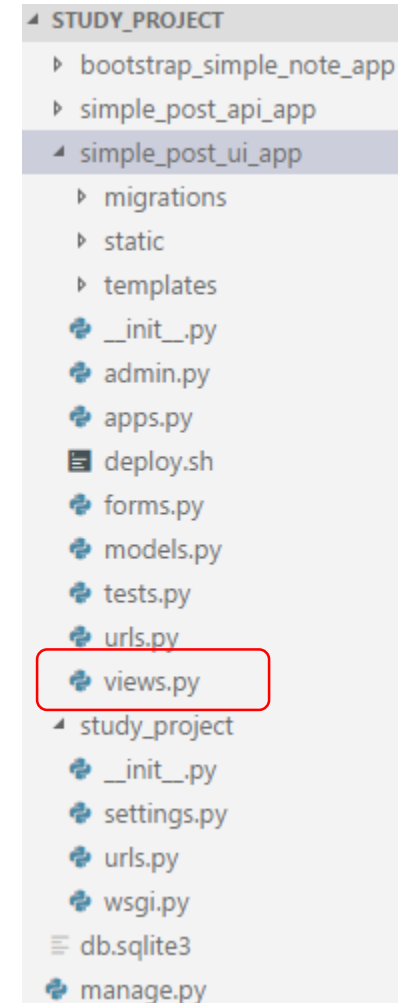
# View. Response, Context, Template

# 10. View Kwargs

```
def view_http_response_url_kwargs(request, header="Default Header",  
content="Default Content"):
```

```
    context_var = {"header": header, "content": content}
```

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





# Url

```
# view 10
```

```
url(    r'^http-response-url-kwarg/$',  
        views.view_http_response_url_kwarg,  
        name="testapp_view-http-response-url-kwarg"),  
  
url(    r'^http-response-url-kwarg/(?P<header>[+ \w]+)/$',  
        views.view_http_response_url_kwarg,  
        name="testapp_view-http-response-url-kwarg"),  
  
url(    r'^http-response-url-kwarg/(?P<header>[+ \w]+)/(?P<content>[+ \w]+)/$',  
        views.view_http_response_url_kwarg,  
        name="testapp_view-http-response-url-kwarg"),
```

Header

Header

Content

```
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  
study_project  
├── __init__.py  
├── settings.py  
├── urls.py  
├── wsgi.py  
├── db.sqlite3  
└── manage.py
```



Пример 7. Передача параметров пользователя в View

Способ 3. GET и POST



# View. Response, Context, Template

## # 11. View GET and POST

```
def view_http_response_get_post(request):
```

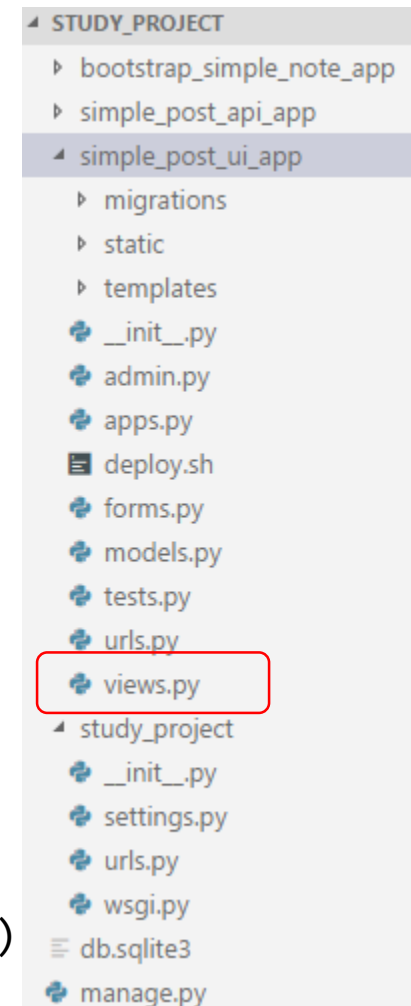
```
    context_var = {"header": None, "content": None}
```

```
    if request.method == "GET":  
        context_var["header"] = request.GET.get("header", None)  
        context_var["content"] = request.GET.get("content", None)
```

```
    if request.method == "POST":  
        context_var["header"] = request.POST.get("header", None)  
        context_var["content"] = request.POST.get("content", None)
```

```
    if context_var["header"] is None and context_var["content"] is None:  
        return render(request,  
            "simple_post_ui_app/templates/html_response_get_post.html", context_var)
```

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

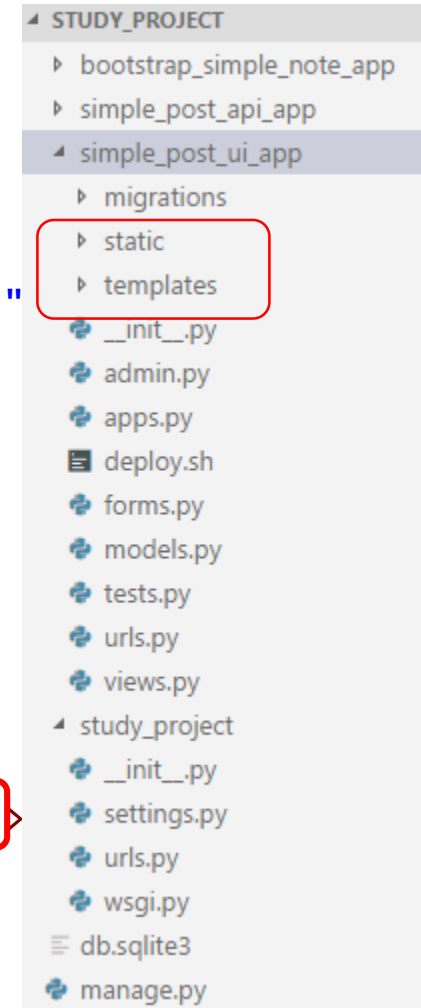
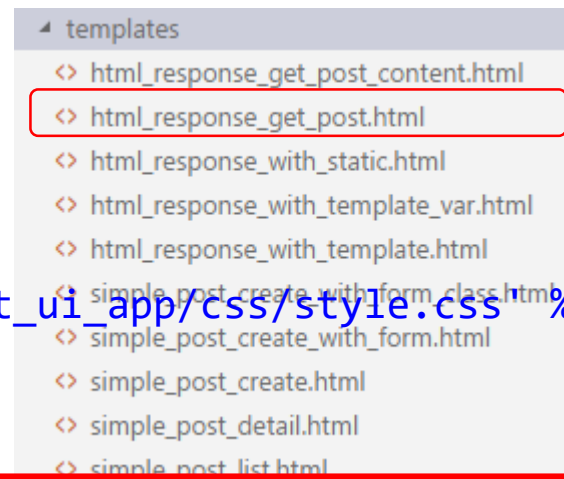






# Template. Форма для заполнения

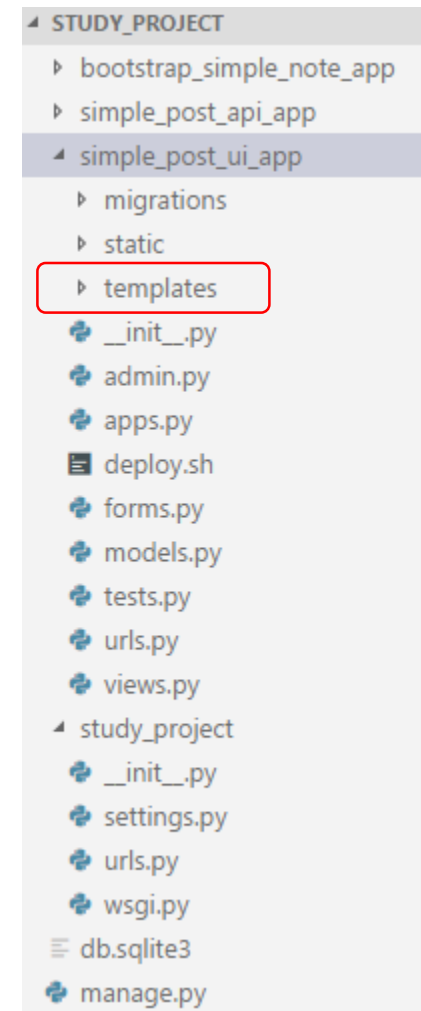
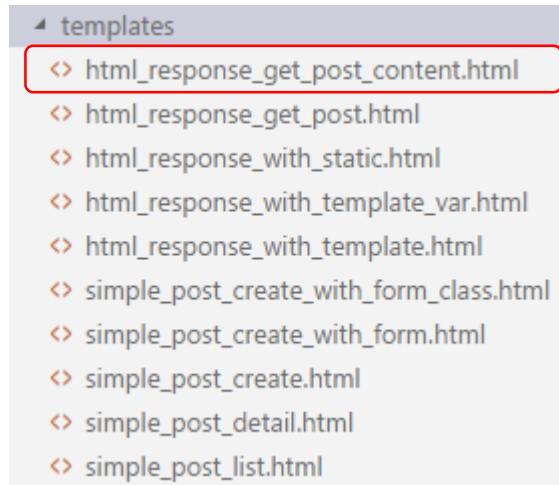
```
<!DOCTYPE html>
<html lang="en">
  <head>
    {% load static %}
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet" href="{% static 'simple_post_ui_app/css/style.css' %}" %}>
  </head>
  <body>
    <div>GET</div>
    <form method="get" action="{% url 'testapp_view-http-response-get-post' %}" %}>
      <input type="text" placeholder="Header" name="header">
      <textarea placeholder="Content" name="content"></textarea>
      <button type="submit">Submit</button>
    </form>
    <div>POST</div>
    <form method="post" action="{% url 'testapp_view-http-response-get-post' %}" %}>
      {% csrf_token %}
      <input type="text" placeholder="Header" name="header">
      <textarea placeholder="Content" name="content"></textarea>
      <button type="submit">Submit</button>
    </form>
  </body>
</html>
```





# Template. Отображение введенных данных

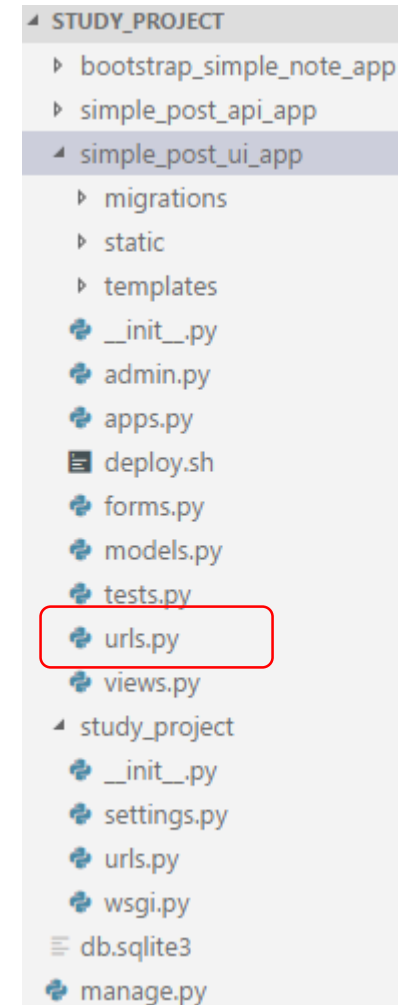
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Title</title>
  </head>
  <body>
    <h1>{{ header }}</h1>
    <p>{{ content }}</p>
    <a href="{% url 'testapp_view-http-response-get-post' %}">Back</a>
  </body>
</html>
```





# Url

```
# view 11
url(    r'^http-response-get-post/$',
        views.view_http_response_get_post,
        name="testapp_view-http-response-get-post"),
```





# Источники

---

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