

Факультатив ФКН

РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ НА БАЗЕ ПЛАТФОРМЫ



Преподаватель

Папулин Сергей Юрьевич (papulin_hse@mail.ru)

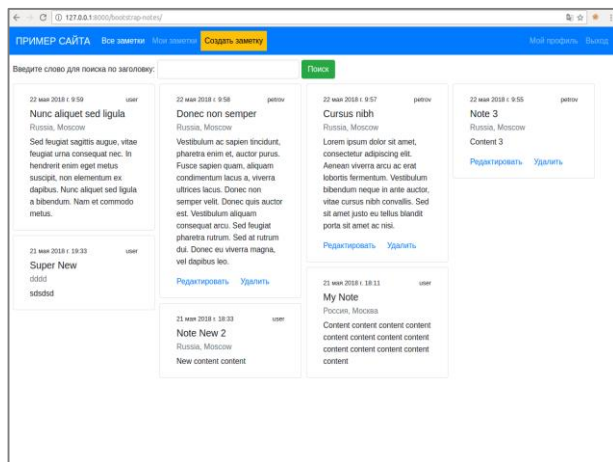


Приложение Bootstrap Simple Note

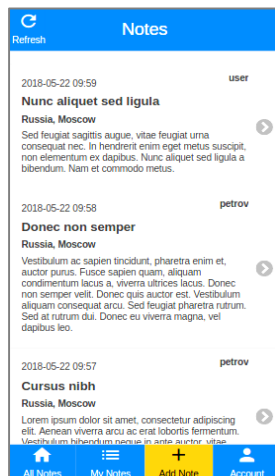


Основные функции

Браузер



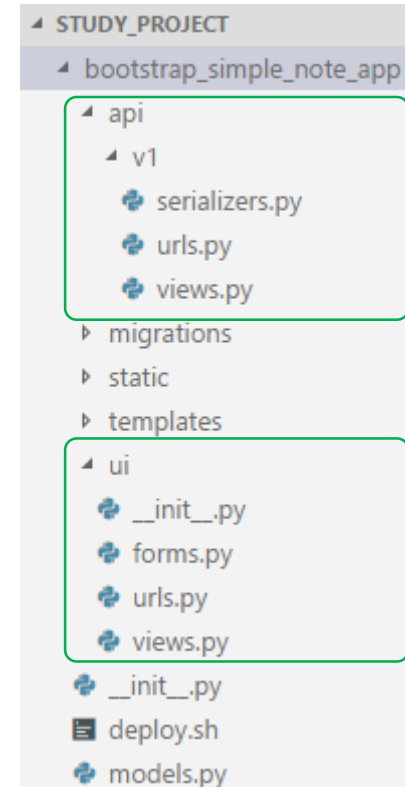
Tizen Mobile Web App



django
Приложение
bootstrap_simple_note_app

django
REST
framework

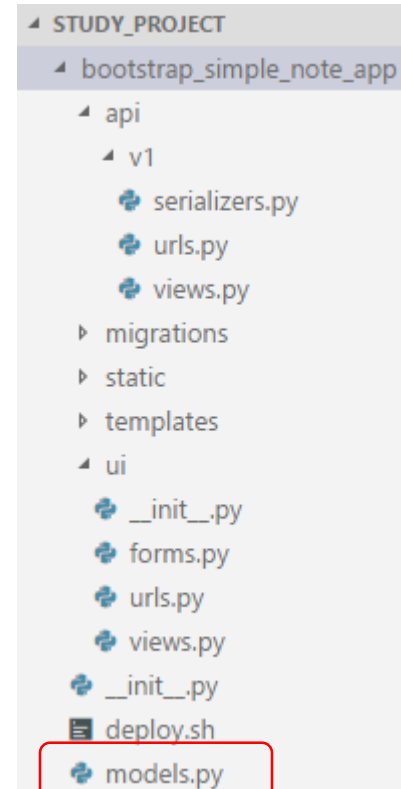
SQLite





Модель данных

```
class SimpleNote(models.Model):  
  
    # The id field will be auto-created  
  
    header = models.CharField(max_length=200, blank=False, null=False,  
                               verbose_name=_("Заголовок"))  
    content = models.TextField(blank=False)  
    created = models.DateTimeField(auto_now_add=True)  
    updated = models.DateTimeField(auto_now=True)  
    location = models.CharField(max_length=200, blank=True, null=True)  
    user = models.ForeignKey(User, on_delete=models.CASCADE)  
  
    def get_absolute_url(self):  
        from django.urls import reverse  
        return reverse("bootstrap-note:detail", kwargs={"pk": self.id})
```





Настройки приложения

См. настройки предыдущих приложений

settings.py

```
INSTALLED_APPS = [  
    ...  
    "bootstrap_simple_note_app" # UI + REST API  
]
```

urls.py

```
url(r'^bootstrap-notes/', include(('bootstrap_simple_note_app.ui.urls', "bootstrap_simple_note_app"),  
namespace="bootstrap-note")),  
  
url(r'^bootstrap-notes/api/v1/', include('bootstrap_simple_note_app.api.v1.urls'))),
```



make migrations

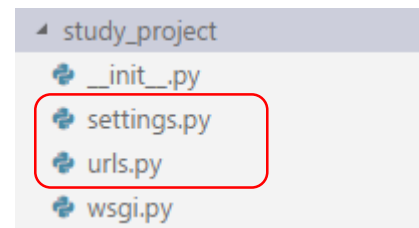
```
python manage.py makemigrations bootstrap_simple_note_app
```

migrate

```
python manage.py migrate bootstrap_simple_note_app
```

run server

```
python manage.py runserver
```





Версия для веб-браузера

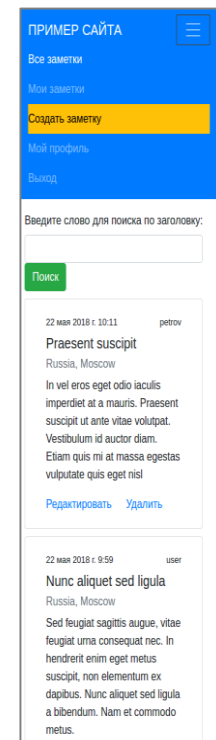
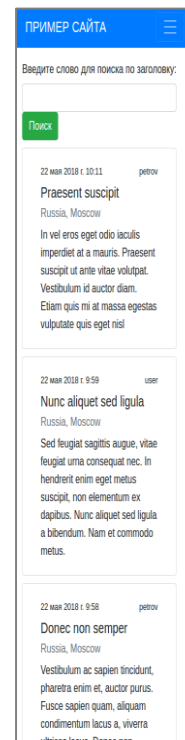
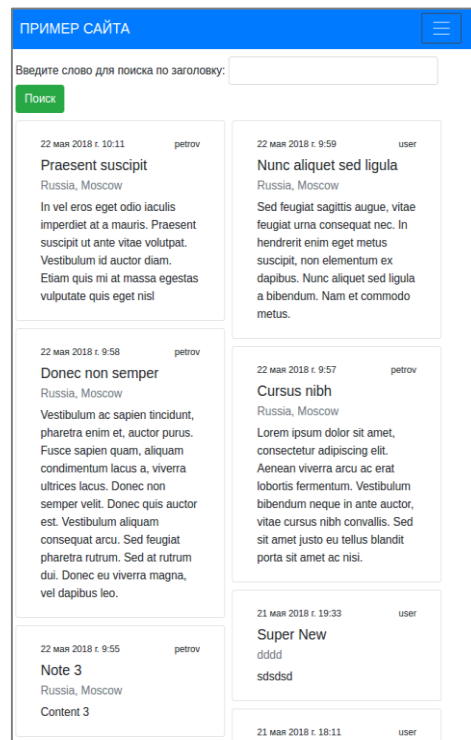
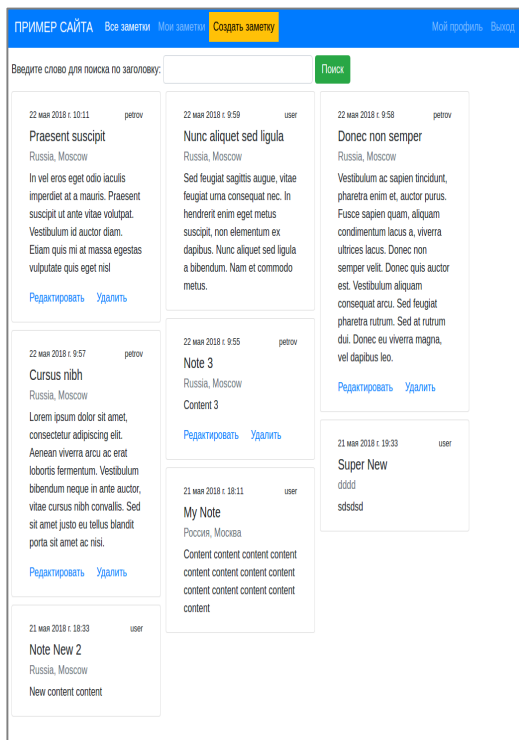
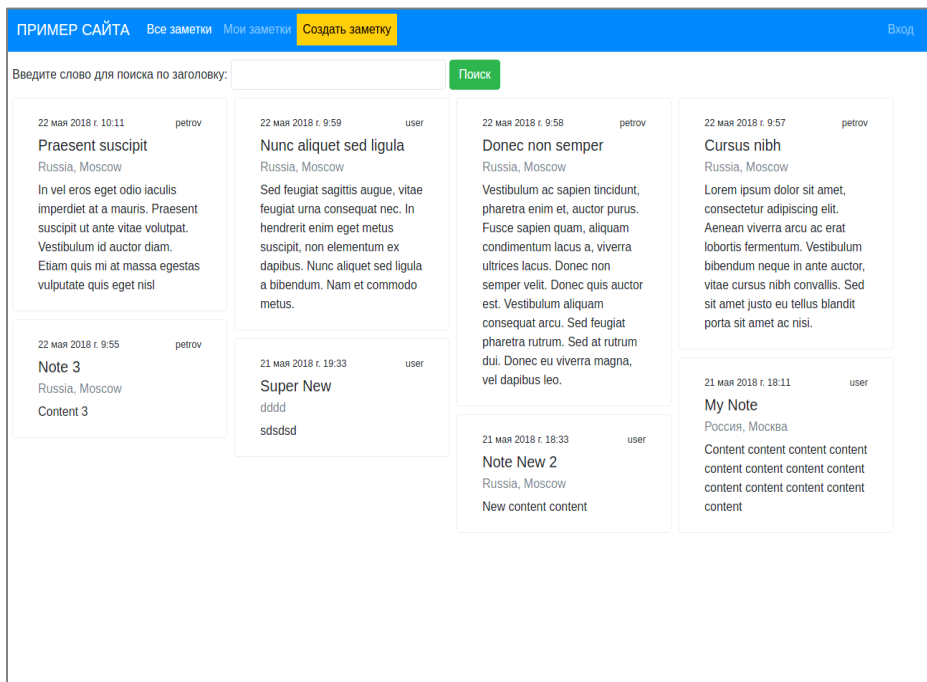


Bootstrap

django

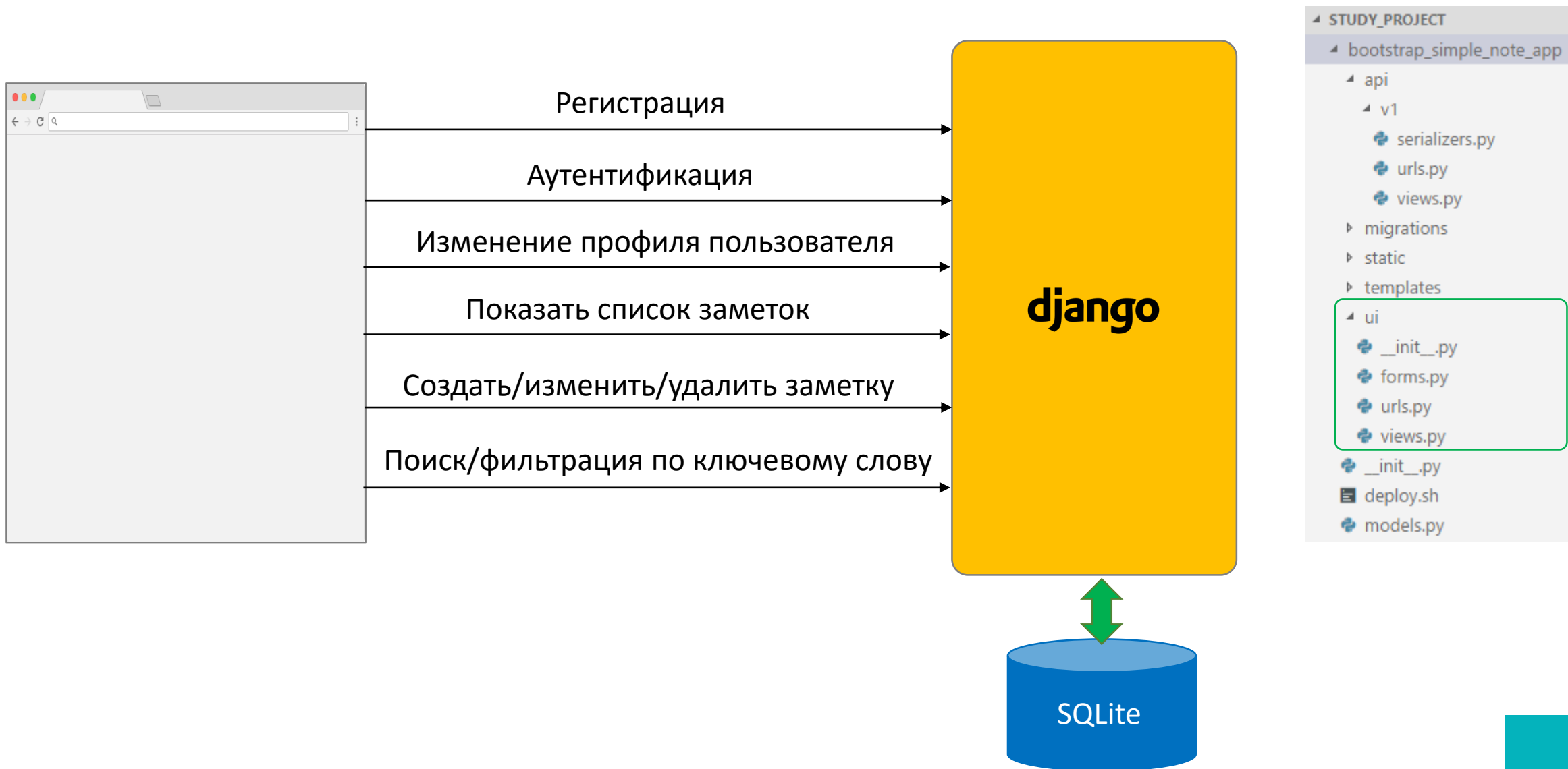


Bootstrap. Набор стилей, компонентов и адаптивный дизайн





Основные функции





Регистрация. Form, Template

```
class CustomSignUpForm(UserCreationForm):  
    """  
    def __init__(self, *args, **kwargs):  
        super().__init__(*args, **kwargs)  
        self.fields["username"].widget.attrs.update({"class": "form-control is-invalid"})  
        self.fields["password1"].widget.attrs.update({"class": "form-control"})  
        self.fields["password2"].widget.attrs.update({"class": "form-control"})  
    """  
    pass
```

<http://127.0.0.1:8000/bootstrap-notes/signup/>

ПРИМЕР САЙТА Все заметки Мои заметки Создать заметку Вход

Имя пользователя:

Пароль:

Подтверждение пароля:

Создать пользователя

```
templates  
  <> bootstrap_base.html  
  <> bootstrap_login.html  
  <> bootstrap_navbar.html  
  <> bootstrap_note_create.html  
  <> bootstrap_note_delete.html  
  <> bootstrap_note_list.html  
  <> bootstrap_note_profile.html  
  <> bootstrap_note_update.html  
  <> bootstrap_note_user_list.html  
  <> bootstrap_register.html
```

```
STUDY_PROJECT  
  bootstrap_simple_note_app  
    api  
      v1  
        serializers.py  
        urls.py  
        views.py  
    migrations  
    static  
    templates  
    ui  
      init.py  
      forms.py  
      urls.py  
      views.py  
      __init__.py  
      deploy.sh  
      models.py
```



Регистрация. View, Url

views.py

```
class SimpleNoteSignUpView(CreateView):

    form_class = CustomSignUpForm
    template_name = "bootstrap_register.html"
    success_url = reverse_lazy("bootstrap-note:list")

    def form_valid(self, form):

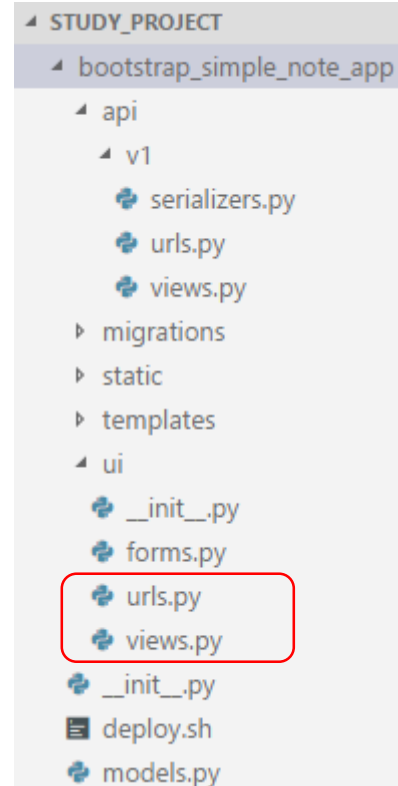
        self.object = form.save()

        new_user = authenticate(username=self.object.username,
                                password=form.cleaned_data["password1"],)
        login(self.request, new_user)

        return redirect(self.get_success_url())
```

urls.py

```
url(r'^signup/$', SimpleNoteSignUpView.as_view(), name="signup"),
```





Аутентификация. Form, Template

```
class CustomAuthenticationForm(AuthenticationForm):
```

```
    username = forms.CharField(
        label="Логин",
        widget=forms.TextInput(attrs={
            "placeholder": "Логин",
            "class": "form-control"
        })
    )
    password = forms.CharField(
        label="Пароль",
        strip=False,
        widget=forms.PasswordInput(attrs={
            "placeholder": "Пароль",
            "class": "form-control"
        })
    )
```

```
templates
├── bootstrap_base.html
├── bootstrap_login.html
├── bootstrap_navbar.html
├── bootstrap_note_create.html
├── bootstrap_note_delete.html
├── bootstrap_note_list.html
├── bootstrap_note_profile.html
├── bootstrap_note_update.html
├── bootstrap_note_user_list.html
└── bootstrap_register.html
```

```
STUDY_PROJECT
├── bootstrap_simple_note_app
│   ├── api
│   │   └── v1
│   │       ├── serializers.py
│   │       ├── urls.py
│   │       └── views.py
│   ├── migrations
│   ├── static
│   ├── templates
│   └── ui
│       ├── init_.py
│       ├── forms.py
│       ├── urls.py
│       ├── views.py
│       ├── __init__.py
│       ├── deploy.sh
│       └── models.py
```

<http://127.0.0.1:8000/bootstrap-notes/login/?next=/bootstrap-notes/>

Вход Регистрация

Логин:

Пароль: [Забыли?](#)

☐ Сохранить пароль

Войти



Аутентификация. View, Url

views.py

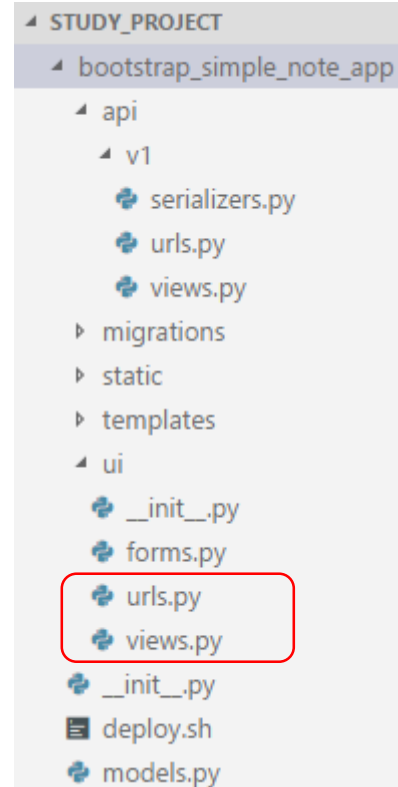
```
class SimpleNoteLoginView(LoginView):

    template_name = "bootstrap_login.html"
    authentication_form = CustomAuthenticationForm

    def get_success_url(self):
        url = self.request.POST.get(REDIRECT_FIELD_NAME, None)
        return url or reverse_lazy("bootstrap-note:list")
```

urls.py

```
url(r'^login/$', SimpleNoteLoginView.as_view(), name="login"),
```





Изменение профиля пользователя. Form, Template

```
class UserProfileForm(forms.ModelForm):  
  
    class Meta:  
        model = User  
        fields = ["username", "first_name",  
                  "last_name", "email"]
```

```
templates  
├> bootstrap_base.html  
├> bootstrap_login.html  
├> bootstrap_navbar.html  
├> bootstrap_note_create.html  
├> bootstrap_note_delete.html  
├> bootstrap_note_list.html  
└> bootstrap_note_profile.html  
├> bootstrap_note_update.html  
├> bootstrap_note_user_list.html  
└> bootstrap_register.html
```

```
STUDY_PROJECT  
├> bootstrap_simple_note_app  
│   ├── api  
│   │   ├── v1  
│   │   │   ├── serializers.py  
│   │   │   ├── urls.py  
│   │   │   └── views.py  
│   ├── migrations  
│   ├── static  
│   ├── templates  
│   └── ui  
│       ├── init_.py  
│       ├── forms.py  
│       ├── urls.py  
│       ├── views.py  
│       ├── __init__.py  
│       ├── deploy.sh  
│       └── models.py
```

<http://127.0.0.1:8000/bootstrap-notes/profile/>

ПРИМЕР САЙТА

Все заметки

Мои заметки

Создать заметку

Мой профиль

Выход

Имя пользователя:

petrov

Имя:

Ivan

Фамилия:

Petrov

Адрес электронной почты:

petrov@petrov.ru

Изменить





Изменение профиля пользователя. View, Url

views.py

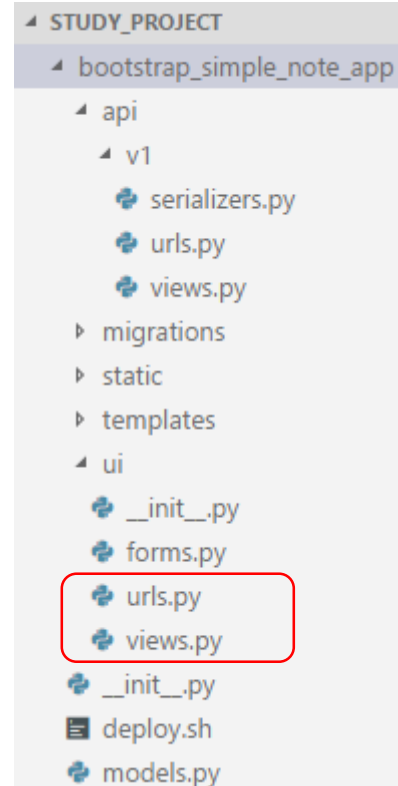
```
class SimpleNoteUserProfile(UpdateView):

    model = User
    form_class = UserProfileForm
    template_name = "bootstrap_note_profile.html"
    success_url = reverse_lazy("bootstrap-note:list")

    def get_object(self, queryset=None):
        return self.request.user
```

urls.py

```
url(r'^profile/$', SimpleNoteUserProfile.as_view(), name="profile")
```





Создание заметки. Form, Template

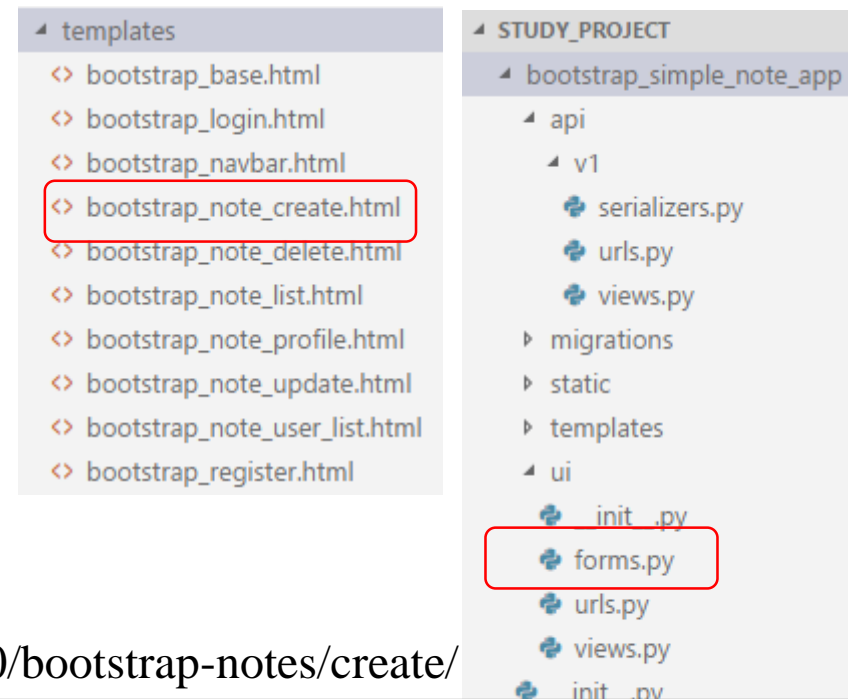
```
class SimpleNoteForm(forms.ModelForm):
```

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

```
class Meta:
```

```
    model = SimpleNote
    fields = ("header", "content", "location")
    labels = {
        "header": _("Заголовок"),
        "content": _("Содержание"),
        "location": _("Местность")
    }
    widgets = {
        "header": forms.TextInput(attrs={"class": "form-control"}),
        "content": forms.Textarea(attrs={"class": "form-control"}),
        "location": forms.TextInput(attrs={"class": "form-control"}),
    }
```

<http://127.0.0.1:8000/bootstrap-notes/create/>



ПРИМЕР САЙТА Все заметки Мои заметки **Создать заметку** Мой профиль Выход

Заголовок:	<input type="text" value="Praesent suscipit"/>
Содержание:	<div>In vel eros eget odio iaculis imperdiet at a mauris. Praesent suscipit ut ante vitae volutpat. Vestibulum id auctor diam. Etiam quis mi at massa egestas vulputate quis eget nisi</div>
Местность:	<input type="text" value="Russia, Moscow"/>

Добавить



Создание заметки. View, Url

Для аутентифицированных пользователей

views.py

```
@method_decorator(login_required(login_url=reverse_lazy("bootstrap-note:login"))),
name="dispatch")
class SimpleNoteCreateView(CreateView):

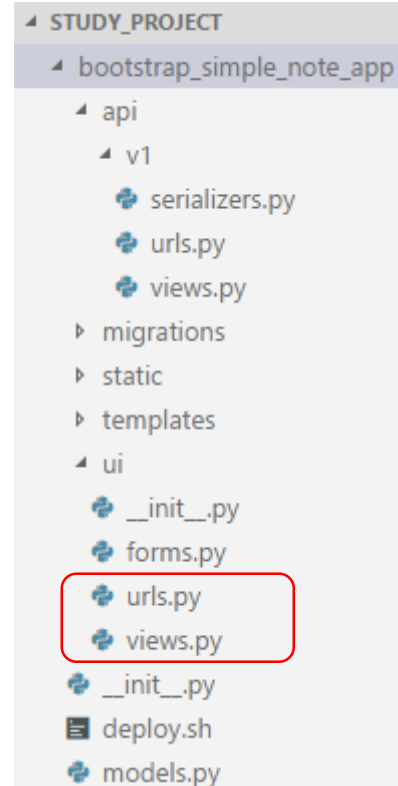
    form_class = SimpleNoteForm
    template_name = "bootstrap_note_create.html"
    success_url = reverse_lazy("bootstrap-note:user-list")

    def form_valid(self, form):

        obj = form.save(commit=False)
        obj.user = self.request.user
        return super().form_valid(form)
```

urls.py

```
url(r'^create/$', SimpleNoteCreateView.as_view(), name="detail-create"),
```





Список заметок пользователя. Form, Template

<http://127.0.0.1:8000/bootstrap-notes/by-user/>

[ПРИМЕР САЙТА](#) [Все заметки](#) [Мои заметки](#) [Создать заметку](#) [Мой профиль](#) [Выход](#)

22 мая 2018 г. 10:11

Praesent suscipit

Russia, Moscow

In vel eros eget odio iaculis imperdiet at a mauris. Praesent suscipit ut ante vitae volutpat. Vestibulum id auctor diam. Etiam quis mi at massa egestas vulputate quis eget nisi

[Редактировать](#) [Удалить](#)

22 мая 2018 г. 9:58

Donec non semper

Russia, Moscow

Vestibulum ac sapien tincidunt, pharetra enim et, auctor purus. Fusce sapien quam, aliquam condimentum lacus a, viverra ultrices lacus. Donec non semper velit. Donec quis auctor est. Vestibulum aliquam consequat arcu. Sed feugiat pharetra rutrum. Sed at rutrum dui. Donec eu viverra magna, vel dapibus leo.

[Редактировать](#) [Удалить](#)

22 мая 2018 г. 9:57

Cursus nibh

Russia, Moscow

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean viverra arcu ac erat lobortis fermentum. Vestibulum bibendum neque in ante auctor, vitae cursus nibh convallis. Sed sit amet justo eu tellus blandit porta sit amet ac nisi.

[Редактировать](#) [Удалить](#)

22 мая 2018 г. 9:55

Note 3

Russia, Moscow

Content 3

[Редактировать](#) [Удалить](#)

templates

<> bootstrap_base.html

<> bootstrap_login.html

<> bootstrap_navbar.html

<> bootstrap_note_create.html

<> bootstrap_note_delete.html

<> bootstrap_note_list.html

<> bootstrap_note_profile.html

<> bootstrap_note_update.html

<> bootstrap_note_user_list.html

<> bootstrap_register.html

STUDY_PROJECT

bootstrap_simple_note_app

api

v1

serializers.py

urls.py

views.py

migrations

static

templates

ui

__init__.py

forms.py

urls.py

views.py

__init__.py

deploy.sh

models.py



Список заметок пользователя. View, Url

Для аутентифицированных пользователей

views.py

```
@method_decorator(login_required(login_url=reverse_lazy("bootstrap-note:login")),
name="dispatch")
class SimpleNoteUserListView(ListView):

    model = SimpleNote
    template_name = "bootstrap_note_user_list.html"

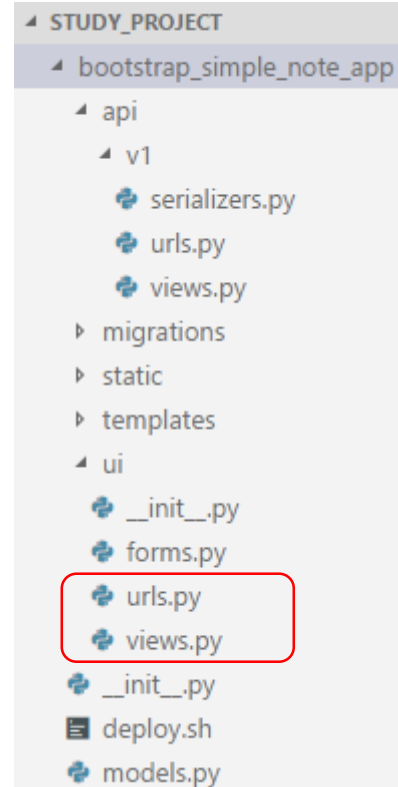
    def get_queryset(self):

        queryset = super().get_queryset()

        return queryset.filter(user=self.request.user).order_by("-created")
```

urls.py

```
url(r'^by-user/$', SimpleNoteUserListView.as_view(), name="user-list"),
```





ПРИМЕР САЙТА

Все заметки

Мои заметки

Создать заметку

Мой профиль

Выход

Введите слово для поиска по заголовку:

Поиск

22 мая 2018 г. 10:11

petrov

Praesent suscipit

Russia, Moscow

In vel eros eget odio iaculis imperdiet at a mauris. Praesent suscipit ut ante vitae volutpat. Vestibulum id auctor diam. Etiam quis mi at massa egetas vulputate quis eget nisl

Редактировать

Удалить

22 мая 2018 г. 9:59

user

Nunc aliquet sed ligula

Russia, Moscow

Sed feugiat sagittis augue, vitae feugiat urna consequat nec. In hendrerit enim eget metus suscipit, non elementum ex dapibus. Nunc aliquet sed ligula a bibendum. Nam et commodo metus.

22 мая 2018 г. 9:58

petrov

Donec non semper

Russia, Moscow

Vestibulum ac sapien tincidunt, pharetra enim et, auctor purus. Fusce sapien quam, aliquam condimentum lacus a, viverra ultrices lacus. Donec non semper velit. Donec quis auctor est. Vestibulum aliquam consequat arcu. Sed feugiat pharetra rutrum. Sed at rutrum dui. Donec eu viverra magna, vel dapibus leo.

Редактировать

Удалить

22 мая 2018 г. 9:57

petrov

Cursus nibh

Russia, Moscow

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean viverra arcu ac erat lobortis fermentum. Vestibulum bibendum neque in ante auctor, vitae cursus nibh convallis. Sed sit amet justo eu tellus blandit porta sit amet ac nisi.

Редактировать

Удалить

22 мая 2018 г. 9:55

petrov

Note 3

Russia, Moscow

Content 3

Редактировать

Удалить

21 мая 2018 г. 18:33

user

Note New 2

Russia, Moscow

New content content

21 мая 2018 г. 18:11

user

My Note

Россия, Москва

Content content content content content content content content content content

21 мая 2018 г. 19:33

user

Super New

dddd

sdsdsd

The screenshot shows a file explorer with two main sections. The left section, titled 'templates', lists several HTML files: bootstrap_base.html, bootstrap_login.html, bootstrap_navbar.html, bootstrap_note_create.html, bootstrap_note_delete.html, bootstrap_note_list.html (highlighted with a red rectangle), bootstrap_note_profile.html, bootstrap_note_update.html, bootstrap_note_user_list.html, and bootstrap_register.html. The right section, titled 'STUDY_PROJECT', shows a directory structure: bootstrap_simple_note_app (expanded to show api, v1, migrations, static, templates, and ui), migrations, static, templates, and ui. The v1 directory is further expanded to show serializers.py, urls.py, and views.py. The ui directory is expanded to show __init__.py, forms.py, urls.py, and views.py. The migrations directory is expanded to show __init__.py, deploy.sh, and models.py.



Поиск по ключевому слову. Form, Template

<http://127.0.0.1:8000/bootstrap-notes/?keyword=su>

[ПРИМЕР САЙТА](#) [Все заметки](#) [Мои заметки](#) [Создать заметку](#) [Вход](#)

22 мая 2018 г. 10:11	petrov	22 мая 2018 г. 9:57	petrov	21 мая 2018 г. 19:33	user
Praesent suscipit		Cursus nibh		Super New	
Russia, Moscow		Russia, Moscow		dddd	
In vel eros eget odio iaculis imperdiet at a mauris. Praesent suscipit ut ante vitae volutpat. Vestibulum id auctor diam. Etiam quis mi at massa egestas vulputate quis eget nisl		Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean viverra arcu ac erat lobortis fermentum. Vestibulum bibendum neque in ante auctor, vitae cursus nibh convallis. Sed sit amet justo eu tellus blandit porta sit amet ac nisi.		sdsdsd	

templates

<> bootstrap_base.html

<> bootstrap_login.html

<> bootstrap_navbar.html

<> bootstrap_note_create.html

<> bootstrap_note_delete.html

<> bootstrap_note_list.html

<> bootstrap_note_profile.html

<> bootstrap_note_update.html

<> bootstrap_note_user_list.html

<> bootstrap_register.html

STUDY_PROJECT

bootstrap_simple_note_app

api

v1

serializers.py

urls.py

views.py

migrations

static

templates

ui

__init__.py

forms.py

urls.py

views.py

__init__.py

deploy.sh

models.py



Список всех заметок. View, Url

views.py

```
class SimpleNoteListView(ListView):

    model = SimpleNote
    template_name = "bootstrap_note_list.html"
    ordering = ["-created"]

    def get_queryset(self):

        keyword = self.request.GET.get("keyword", None)

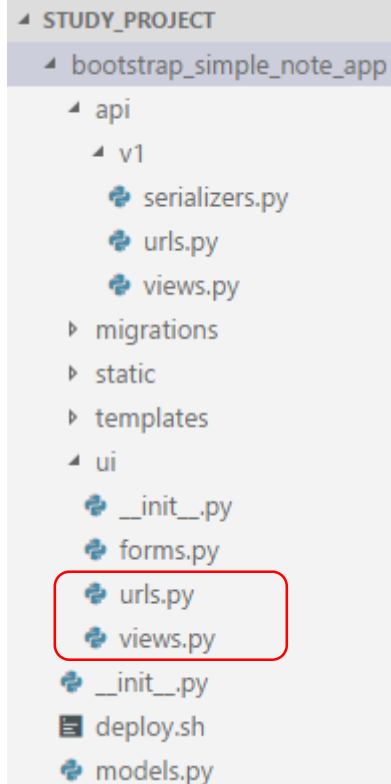
        queryset = super().get_queryset()

        if keyword is not None:
            queryset = queryset.filter(header__icontains=keyword)

        return queryset
```

urls.py

```
url(r'^$', SimpleNoteListView.as_view(), name="list"),
```





Редактирование заметки. Form, Template

SimpleNoteForm

<http://127.0.0.1:8000/bootstrap-notes/6/update/>

ПРИМЕР САЙТА

Все заметки

Мои заметки

Создать заметку

Мой профиль

Выход

Заголовок:

Donec non semper

Содержание:

Vestibulum ac sapien tincidunt, pharetra enim et, auctor purus. Fusce sapien quam, aliquam condimentum lacus a, viverra ultrices lacus. Donec non semper velit. Donec quis auctor est. Vestibulum aliquam consequat arcu. Sed feugiat pharetra rutrum. Sed at rutrum dui. Donec eu viverra magna, vel dapibus leo.

Местность:

Russia, Moscow

Изменить

- templates
 - bootstrap_base.html
 - bootstrap_login.html
 - bootstrap_navbar.html
 - bootstrap_note_create.html
 - bootstrap_note_delete.html
 - bootstrap_note_list.html
 - bootstrap_note_profile.html
 - bootstrap_note_update.html
 - bootstrap_note_user_list.html
 - bootstrap_register.html

- STUDY_PROJECT
 - bootstrap_simple_note_app
 - api
 - v1
 - serializers.py
 - urls.py
 - views.py
 - migrations
 - static
 - templates
 - ui
 - init_.py
 - forms.py
 - urls.py
 - views.py
 - __init__.py
 - deploy.sh
 - models.py



Редактирование заметки. View, Url

Для аутентифицированных пользователей

views.py

```
@method_decorator(login_required, name="dispatch")
class SimpleNoteUpdateView(UpdateView):

    model = SimpleNote
    form_class = SimpleNoteForm
    template_name = "bootstrap_note_update.html"
    success_url = reverse_lazy("bootstrap-note:user-list")

    def get_object(self, queryset=None):

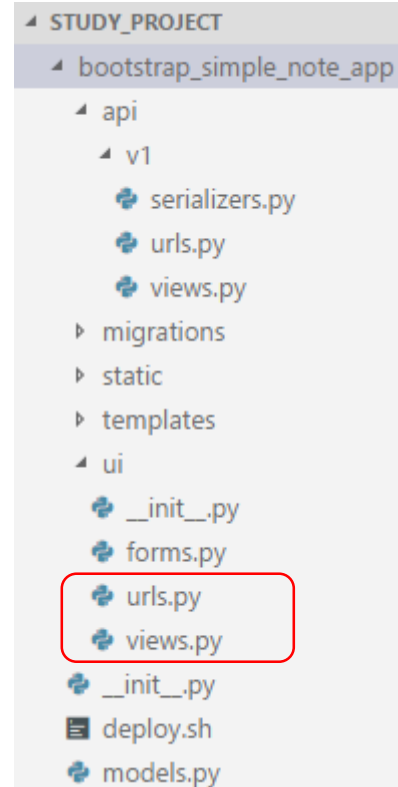
        pk = self.kwargs.get("pk")

        if pk is not None:
            try:
                return self.model.objects.get(pk=pk, user=self.request.user)
            except self.model.DoesNotExist:
                raise Http404

        raise Http404
```

urls.py

```
url(r'^(?P<pk>\d+)/update/$', SimpleNoteUpdateView.as_view(), name="detail-update"),
```





Удаление заметки. Form, Template

<http://127.0.0.1:8000/bootstrap-notes/7/delete/>

[ПРИМЕР САЙТА](#) [Все заметки](#) [Мои заметки](#) [Создать заметку](#) [Мой профиль](#) [Выход](#)

Вы действительно хотите удалить "Praesent suscipit"?

[Удалить](#)

- templates
 - bootstrap_base.html
 - bootstrap_login.html
 - bootstrap_navbar.html
 - bootstrap_note_create.html
 - bootstrap_note_delete.html
 - bootstrap_note_list.html
 - bootstrap_note_profile.html
 - bootstrap_note_update.html
 - bootstrap_note_user_list.html
 - bootstrap_register.html

- STUDY_PROJECT
 - bootstrap_simple_note_app
 - api
 - v1
 - serializers.py
 - urls.py
 - views.py
 - migrations
 - static
 - templates
 - ui
 - init_.py
 - forms.py
 - urls.py
 - views.py
 - __init__.py
 - deploy.sh
 - models.py



Удаление заметки. View, Url

Для аутентифицированных пользователей

views.py

```
@method_decorator(login_required, name="dispatch")
class SimpleNoteDeleteView(DeleteView):

    model = SimpleNote
    success_url = reverse_lazy("bootstrap-post:list")
    template_name = "bootstrap_note_delete.html"

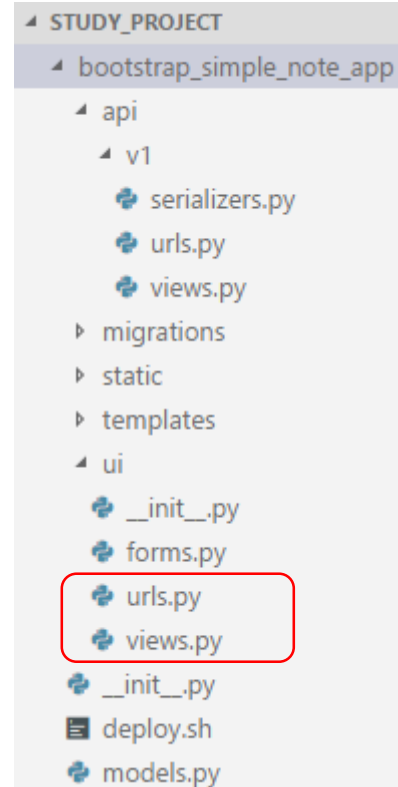
    def get_object(self, queryset=None):

        obj = super().get_object()
        if obj.user == self.request.user:
            return obj

        raise Http404()
```

urls.py

```
url(r'^(?P<pk>\d+)/delete/$', SimpleNoteDeleteView.as_view(), name="detail-delete"),
```



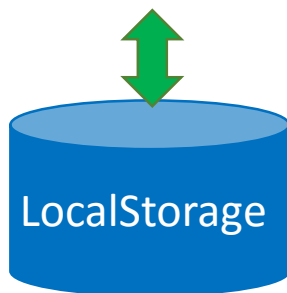
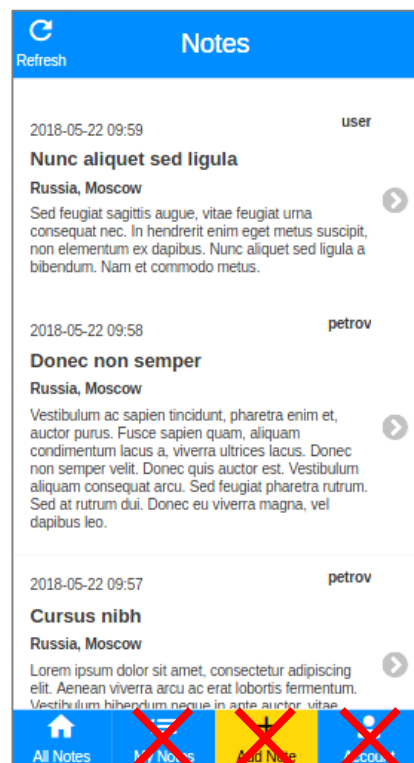


Версия для мобильного веб-приложения





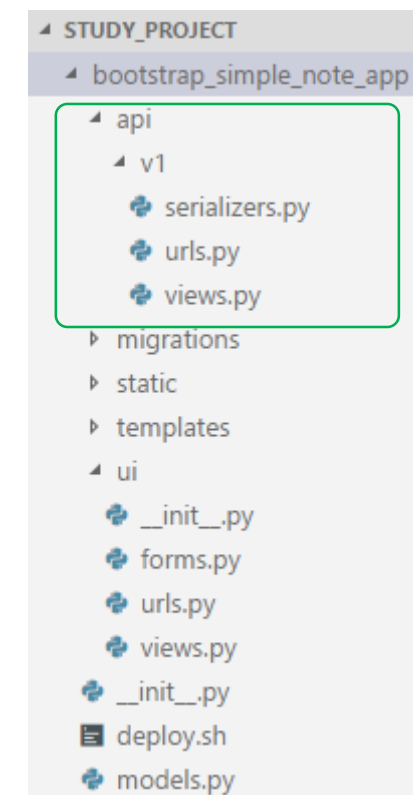
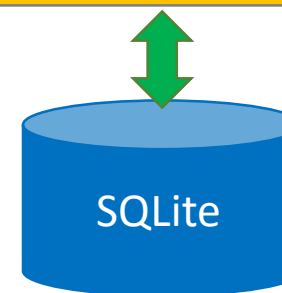
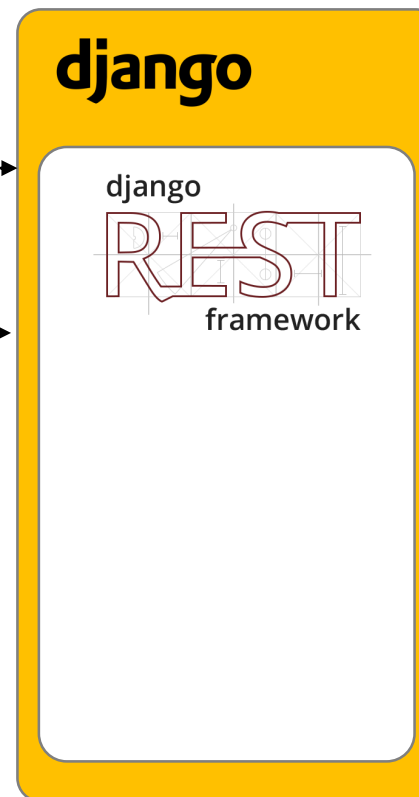
Мобильное веб-приложение



- Заметки
- Timestamp последнего обращения к серверу

Получить все заметки

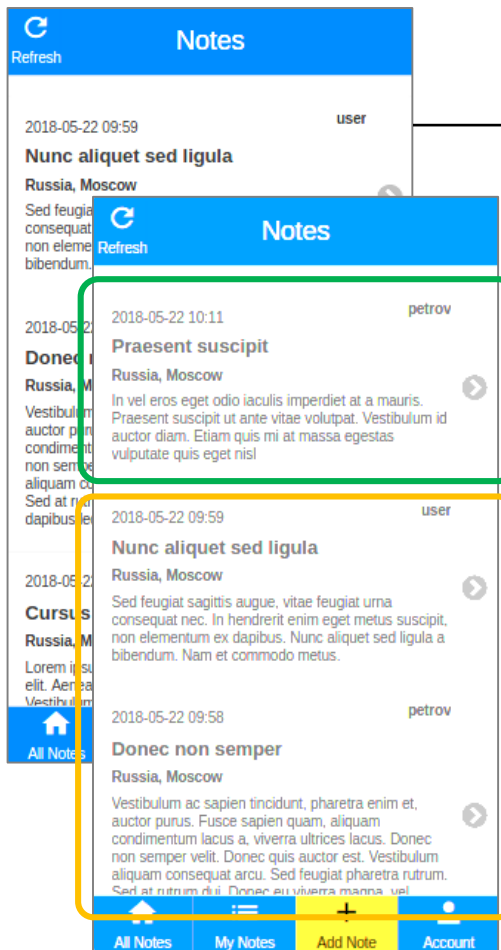
Получить новые заметки





Список всех заметок

Мобильное веб-приложение



LocalStorage

Получить новые заметки

GET
`http://127.0.0.1:8000/bootstrap-notes/api/v1/notes/?timestamp=1526927580.0`

Новая заметка

django

django
REST
framework



Material Design Icons



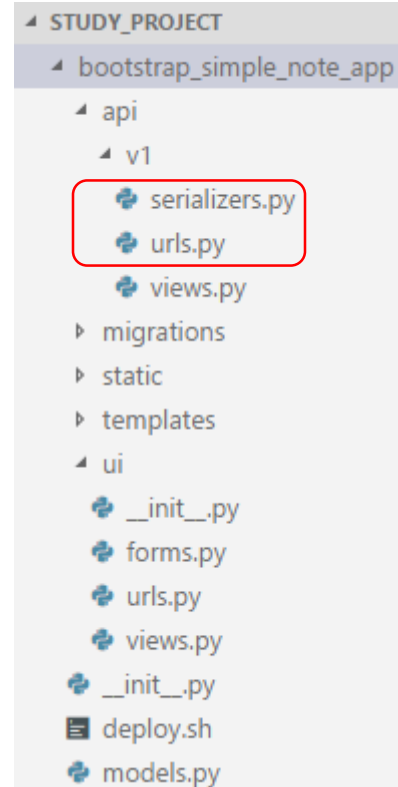
Сервер. Получение заметок. Serializer, Url

serializers.py

```
class SimpleNoteSerializer(serializers.ModelSerializer):  
  
    user = serializers.ReadOnlyField(source="user.username")  
  
    class Meta:  
        model = SimpleNote  
        fields = "__all__"
```

urls.py

```
url(r'^notes/$', SimpleNoteListAPIView.as_view()),
```





Сервер. Получение заметок. View

```
class SimpleNoteListAPIView(ListAPIView):

    queryset = SimpleNote.objects.all()
    serializer_class = SimpleNoteSerializer

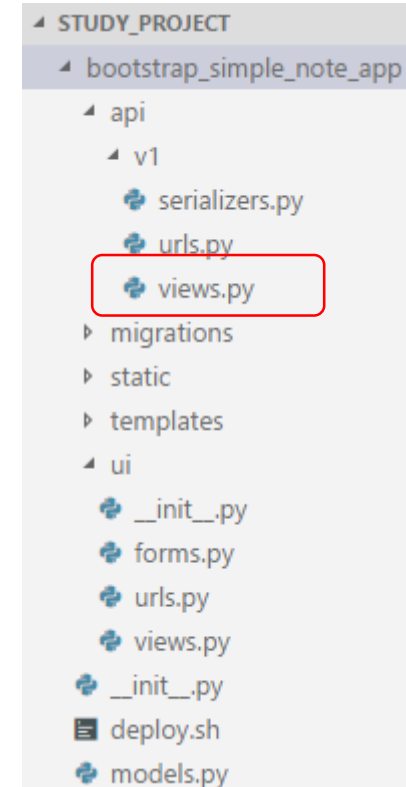
    def get_queryset(self):

        queryset = super().get_queryset()

        timestamp_param = self.request.GET.get("timestamp", None)

        if timestamp_param:
            datetime_timestamp =
                datetime.fromtimestamp(float(timestamp_param)).isoformat()
            queryset = queryset.filter(updated__gt=datetime_timestamp)

        return queryset.order_by("-updated")
```





Сервер. Получение заметок. View

```
def list(self, request, *args, **kwargs):

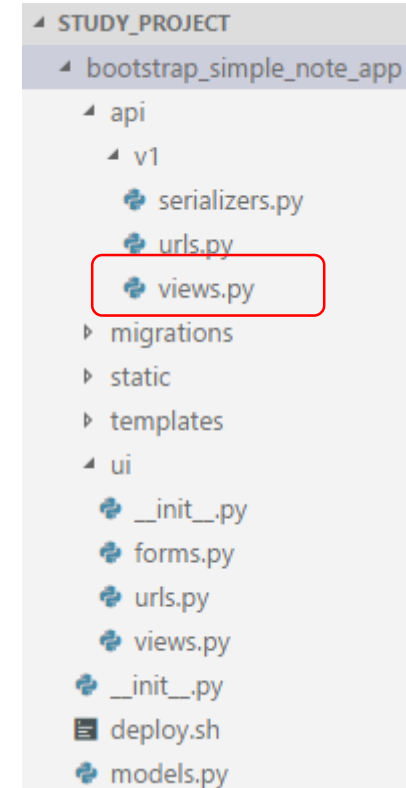
    queryset = self.filter_queryset(self.get_queryset())

    page = self.paginate_queryset(queryset)
    if page is not None:
        serializer = self.get_serializer(page, many=True)
        return self.get_paginated_response(serializer.data)

    serializer = self.get_serializer(queryset, many=True)

    response_data = {
        "status": "OK",
        "notes": serializer.data,
        "timestamp": datetime.now().timestamp()
    }

    return Response(response_data)
```





Мобильное веб-приложение. HTML

Refresh

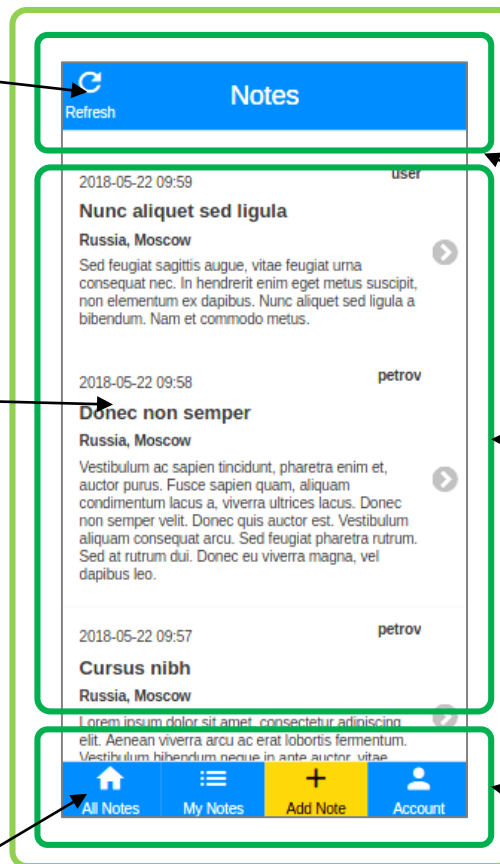
.notes-page-btn-id

ListView

#note-list-view-id

All Notes

. notes-page-btn-id



Page

#notes-page-id

Header

Content

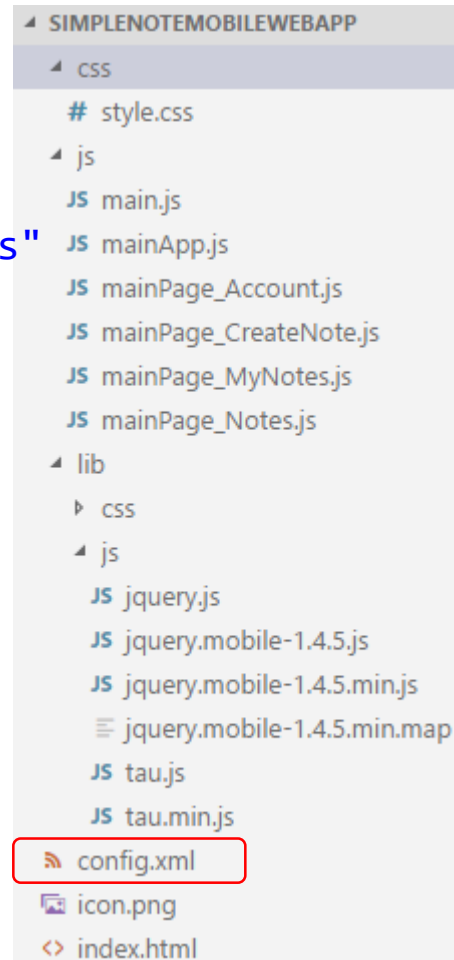
Footer

```
SIMPLENOTEMOBILEWEBAPP
├── css
│   └── style.css
├── js
│   ├── main.js
│   ├── mainApp.js
│   ├── mainPage_Account.js
│   ├── mainPage_CreateNote.js
│   ├── mainPage_MyNotes.js
│   └── mainPage_Notes.js
├── lib
│   ├── css
│   │   ├── jquery.js
│   │   ├── jquery.mobile-1.4.5.js
│   │   ├── jquery.mobile-1.4.5.min.js
│   │   └── jquery.mobile-1.4.5.min.map
│   ├── tau.js
│   └── tau.min.js
├── config.xml
├── icon.png
└── index.html
```




Мобильное веб-приложение. Конфигурация

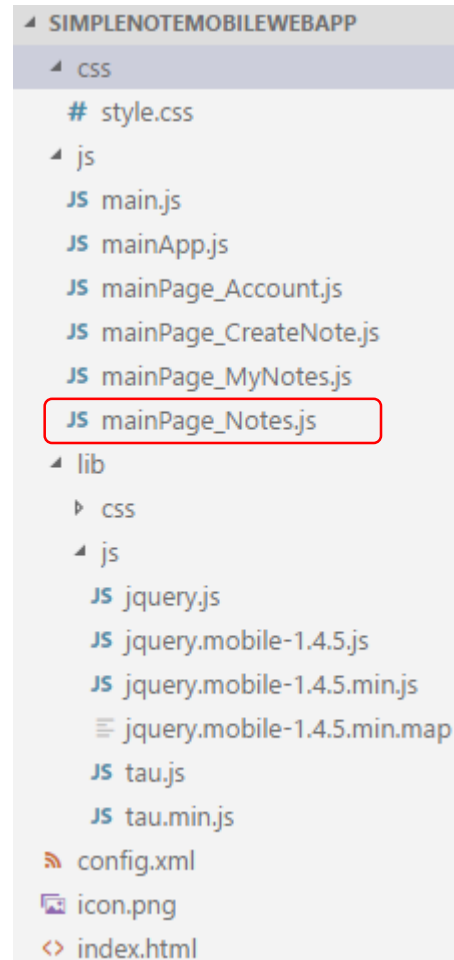
```
<?xml version="1.0" encoding="UTF-8"?>
<widget xmlns:tizen="http://tizen.org/ns/widgets" xmlns="http://www.w3.org/ns/widgets"
id="http://yourdomain/SimpleNoteMobileWebApp" version="1.0.0" viewmodes="maximized">
  <access origin="*" subdomains="true"/>
  <tizen:application id="vuFzFUvK0z.SimpleNoteMobileWebApp" package="vuFzFUvK0z"
required_version="2.4"/>
  <content src="index.html"/>
  <feature name="http://tizen.org/feature/screen.size.all"/>
  <icon src="icon.png"/>
  <name>SimpleNoteMobileWebApp</name>
  <tizen:profile name="mobile"/>
  <tizen:setting context-menu="disable"/>
</widget>
```





Мобильное веб-приложение. JS. Запрос к серверу

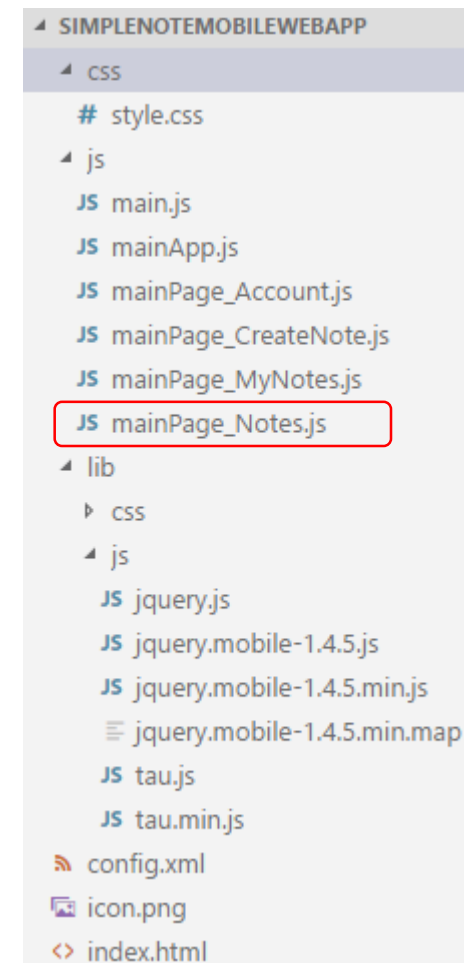
```
var defaults = {  
  url: "http://127.0.0.1:8000/bootstrap-notes/api/v1/notes/",  
  containerSelector: "#note-list-view-id",  
};
```





Мобильное веб-приложение. JS. Запрос к серверу

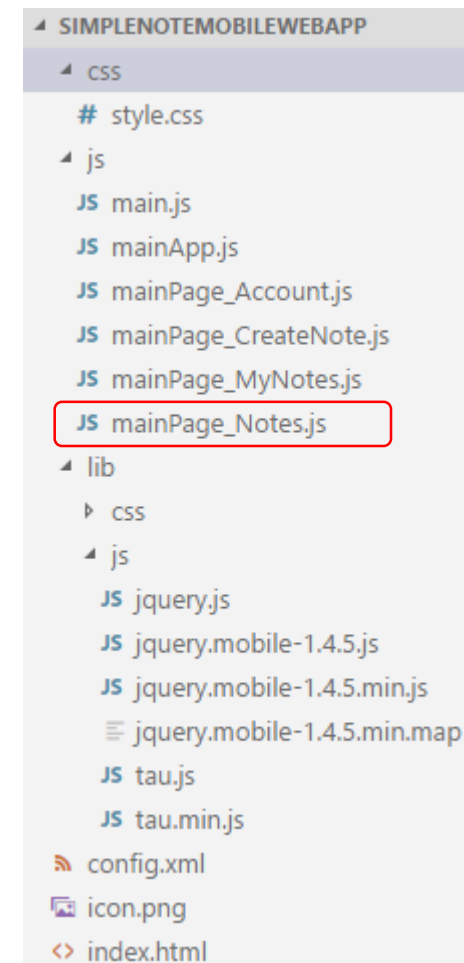
```
function requestData() {  
  
    // get local timestamp  
    var localTimestamp = pageStorage.getTimestamp();  
    // request all after localTimestamp  
    requestNotes(localTimestamp);  
}  
  
function requestNotes(timestamp) {  
  
    var url = options.url;  
  
    if (timestamp) url += "?timestamp=" + timestamp;  
  
    $.ajax({  
        type: "GET",  
        url: url,  
        success: onSuccessRequestNotes,  
        error: onErrorRequestNotes,  
        dataType: "json",  
    });  
}
```





Мобильное веб-приложение. JS. Запрос к серверу

```
function onSuccessRequestNotes(response) {  
  
    if (response.status === "OK") {  
  
        // set local timestamp  
        pageStorage.setTimestamp(response.timestamp);  
  
        if (response.notes.length > 0) {  
  
            // add new notes to the storage  
            pageStorage.addNotes(response.notes);  
            // render new notes  
            renderNewNotes(response.notes);  
        }  
    } else {  
        console.log("onErrorRequestNotes => Error");  
    }  
}
```

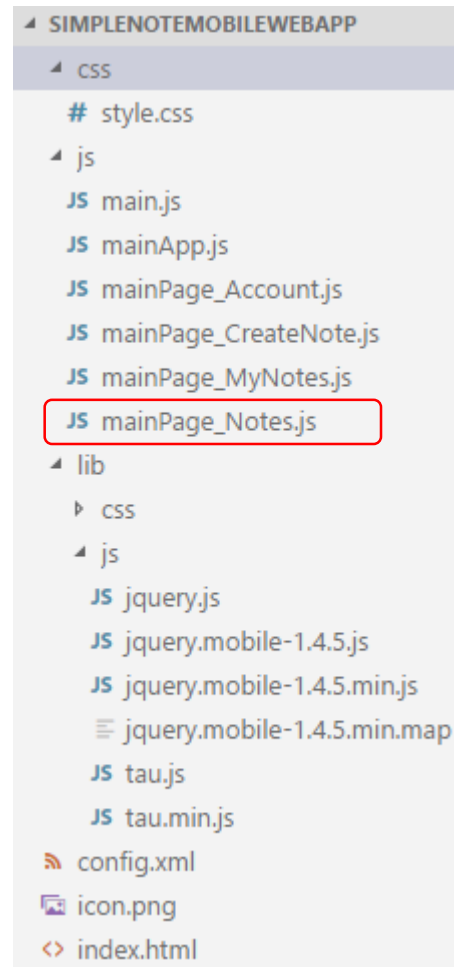




Мобильное веб-приложение. JS. Отображение новых заметок

```
function renderNewNotes(notes) {
    html2render = '';
    for(var i=0; i < notes.length; i++) html2render += getHtmlNoteItem(notes[i]);
    $container.prepend(html2render);
    $container.listview("refresh");
}
function getHtmlNoteItem(note) {

    updated = note.updated.split("T");
    if (updated.length == 2) {
        date = updated[0];
        time = updated[1].split(":");
        note.updated = date + " " + time[0] + ":" + time[1];
    }
    return '<li>' +
        '<a href="#">' +
            '<p>' + note.updated + '</p>' +
            '<h2>' + note.header + '</h2>' +
            '<p><strong>' + note.location + '</strong></p>' +
            '<p>' + note.content + '</p>' +
            '<p class="ui-li-aside"><strong>' + note.user + '</strong></p>' +
            '</a>' +
        '</li>';
}
```





ИСТОЧНИКИ

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

<http://www.django-rest-framework.org/>

<https://getbootstrap.com/docs/4.1/getting-started/introduction/>

<https://api.jquerymobile.com/>