Московский государственный технический университет имени Н. Э. Баумана

Факультет «Информатика и системы управления»

Кафедра ИУ5

Отчёт по

Лабораторной работе 6

Выполнил:

Бахрамов Н.А

Группа ИУ5-53Б

Москва

2021

Задание:

В этой лабораторной работе Вы познакомитесь с популярной СУБД MySQL, создадите свою базу данных. Также Вам нужно будет дополнить свои классы предметной области, связав их с созданной БД. После этого Вы создадите свои модели с помощью Django ORM, отобразите объекты из БД с помощью этих моделей.

1. Создайте сценарий с подключением к БД и несколькими запросами, примеры рассмотрены в методических указаниях.
2. Реализуйте модели Вашей предметной области из предыдущей ЛР (минимум две модели, т.е. две таблицы).
3. Создайте представления и шаблоны Django для отображения списка данных по каждой из сущностей.

Urls

```python
"""
from django.contrib import admin
from media import views as media_views
from django.urls import include, path
from rest_framework import routers

router = routers.DefaultRouter()
router.register(r'films', media_views.StockViewSet)

# Wire up our API using automatic URL routing.
# Additionally, we include login URLs for the browsable API.
urlpatterns = [
    path('', include(router.urls)),
    path('api-auth/', include('rest_framework.urls', namespace='rest_framework')),

    path('admin/', admin.site.urls),]
```

Settings

```python
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    # DRF
    'rest_framework',

    # Наше приложение
    'media',
]
```

Views

```python
class StockViewSet(viewsets.ModelViewSet):
    """
    API endpoint, который позволяет просматривать и редактировать акции компаний
    """
    # queryset всех пользователей для фильтрации по дате последнего изменения
    queryset = Filem.objects.all().order_by('film_name')
    serializer_class = StockSerializer  # Сериализатор для модели
```

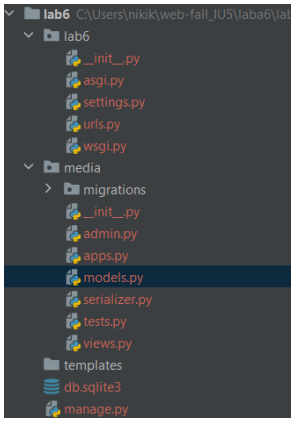
Models

```python
# Create your models here.
class Filem(models.Model):
    film_name = models.CharField(max_length=50, verbose_name="название фильма")
    description = models.CharField(max_length=255, verbose_name="описание фильма")
```

Serializers

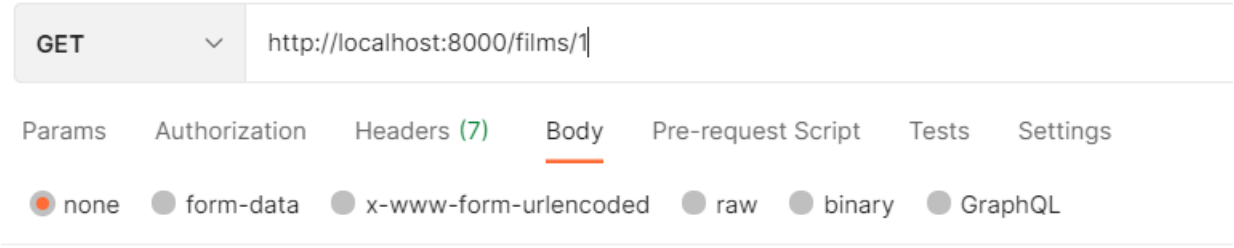
```python
class StockSerializer(serializers.ModelSerializer):
    class Meta:
        # Модель, которую мы сериализуем
        model = Filem
        # Поля, которые мы сериализуем
        fields = ["pk", "film_name", "description"]
```

Структура



Вид


```
[
    {
        "pk": 3,
        "film_name": "Красное уведомление",
        "description": "очень смешной фильм"
    },
    {
        "pk": 2,
        "film_name": "Суперсемейка",
        "description": "очень крутой мультфильм"
    },
    {
        "pk": 1,
        "film_name": "бойцовский клуб",
        "description": "очень крутой фильм"
    }
]
```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```
{
    "pk": 1,
    "film_name": "бойцовский клуб",
    "description": "очень крутой фильм"
}
```

POST http://localhost:8000/films/

Params Authorization Headers (9) Body Pre-request Script

none form-data x-www-form-urlencoded raw binary

```
{
    "film_name": "вечное сияние чистого разума",
    "description": "очень очень крутой фильм"
}
```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```
{
    "pk": 4,
    "film_name": "вечное сияние чистого разума",
    "description": "очень очень крутой фильм"
}
```