# ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет «Инфокоммуникационных технологий» Направление подготовки «09.03.03 Прикладная информатика» Бакалаврская программа «Мобильные и сетевые технологии»

## Лабораторная работа №3

по дисциплине «Web программирование»

«Django Rest Framework, документирование API»

Выполнил		
	(подпись)	_/ <u>Комаров Г. Ю., К33402</u> (Фамилия И.О., группа)
Прородия		
Проверил		/ <u>Говоров А. И.</u>
Лата		

Санкт-Петербург 2021

## Выполнение работы

### Часть 1. Практическая работа №3.1

- 1. В рамках практической работы реализованы:
  - Методы для работы с профессиями



```
class Profession(models.Model):
20
21
22
23
24
           class Meta:
25
                verbose_name = 'Профессия'
                verbose_name_plural = 'Профессии'
26
27
            title = models.CharField(max_length=120, unique=True, verbose_name='Название')
28
            description = models.TextField(verbose_name='Описание')
29
```

```
6 class ProfessionSerializer(serializers.ModelSerializer):
7 class Meta:
8 model = Profession
9 fields = '__all__'
10
```

```
@extend_schema_view(
           create=profession_create_schema,
10
11
           retrieve=profession_retrieve_schema,
12
           list=profession_list_schema,
           update=profession_update_schema,
13
14
           partial_update=profession_update_schema,
           destroy=profession_delete_schema,
15
16
       class ProfessionViewSet(ModelViewSet):
17
           serializer_class = ProfessionSerializer
18 🌖
           queryset = Profession.objects.all()
```

• Методы для работы с умениями



```
class Skill(models.Model):
 5
6
           class Meta:
               verbose_name = 'Умение'
10
11
                verbose_name_plural = 'Умения'
12
           title = models.CharField(max_length=120, unique=True, verbose_name='Haименование')
14
15
  ©
           def __str__(self):
16
                return self.title
```

```
12 class SkillSerializer(serializers.ModelSerializer):
13 class Meta:
14 model = Skill
15 fields = '__all__'
```

```
22
       @extend_schema_view(
23
            create=skill_create_schema,
24
           retrieve=skill_retrieve_schema,
25
            list=skill_list_schema,
26
           update=skill_update_schema,
27
            partial_update=skill_update_schema,
28
           destroy=skill_delete_schema,
       )
29
      class SkillViewSet(ModelViewSet):
30
31 0
            serializer_class = SkillSerializer
            queryset = Skill.objects.all()
32 0
```

#### • Методы для работы с воинами



```
class SkillOfWarrior(models.Model):
37
                    class Meta:
                           verbose_name = 'Умение воина'
                           verbose_name_plural = 'Умение воинов'
                   skill = models.ForeignKey('Skill', verbose_name='Умение', on_delete=models.CASCADE)
warrior = models.ForeignKey('Warrior', verbose_name='Воин', on_delete=models.CASCADE)
level = models.IntegerField(verbose_name='Уровень освоения умения')
43
            class WarriorRaces(models.TextChoices):
                    STUDENT = 'student', 'Студент'
                    DEVELOPER = 'developer', 'Разработчик'
52
                    TEAMLEAD = 'teamlead', 'Тимлид'
56
           class Warrior(models.Model):
58
                    class Meta:
61
                           verbose_name = 'Воин'
                           verbose_name_plural = 'Воины'
                   race = models.CharField(max_length=10, choices=WarriorRaces.choices, verbose_name='Paca')
name = models.CharField(max_length=120, verbose_name='Имя')
level = models.IntegerField(default=0, verbose_name='Уровень')
skills = models.ManyToManyField(Skill, through=SkillOfWarrior, related_name='warrior_skils', verbose_name='Умения')
profession = models.ForeignKey(Profession, on_delete=models.CASCADE, blank=True, null=True, verbose_name='Профессия')
67
```

```
class SkillOfWarriorSerializer(serializers.ModelSerializer):
           class Meta:
20
               model = SkillOfWarrior
               fields = ['title', 'level']
21
               read_only_fields = fields
22
23
           title = serializers.CharField(source='skill', read_only=True)
27
       class WarriorSerializer(serializers.ModelSerializer):
28
           class Meta:
               model = Warrior
30
               fields = '__all__'
31
           profession = serializers.SlugRelatedField(slug_field='title', gueryset=Profession.objects.all())
32
           skills = SkillOfWarriorSerializer(source='skillofwarrior_set', many=True, read_only=True)
```

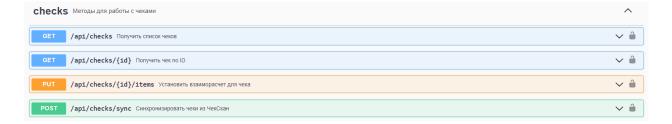
```
35
       @extend_schema_view(
           create=warrior_create_schema,
36
37
           retrieve=warrior_retrieve_schema,
38
           list=warrior_list_schema,
39
           update=warrior_update_schema,
           partial_update=warrior_update_schema,
40
41
           destroy=warrior_delete_schema,
       )
42
43
      ⇔class WarriorViewSet(ModelViewSet):
44 💿
            serializer_class = WarriorSerializer
45 💇
           queryset = Warrior.objects.all()
```

Так же была создана документация в формате OpenAPI v3 с помощью **drf-spectacular**, т. к. библиотека drf-yasg более не поддерживается и не всегда генерирует точную документацию.

## Часть 2. Выполнение лабораторной работы

В качестве проекта для работы №3 и №4 была идея создать сервис для взаиморасчетов, где информация о тратах подгружается с помощью QR кода чека.

Чеки загружаются с помощью приложения ЧекСкан (такой способ был выбран ввиду закрытого АРІ ФНС). После этого есть возможность указать в профиле токен ЧекСкан и синхронизировать чеки с их АРІ. В чеке содержится общая сумма, список товаров, их колво и цена. После загрузки чеков в систему можно разбить каждый товар на несколько пользователей.



Методы для работы с чеками

```
Check v {
    id*
                           integer
                           title: ID чека
    items*
                           > [...]
string($date-time)
    createdAt*
                           readOnly: true
                           title: Дата создания
    updatedAt*
                           string($date-time)
                           readOnly: true
                           title: Дата обновления
   qrData
                           string
                           nullable: true
                           title: Данные QR кода
                           maxLength: 100
                           string
                           title: Наименование
                           maxLength: 1000
    address
                           string
                           nullable: true
                           title: Adpec
                           maxLength: 1000
    date*
                           string($date-time)
                           title: Дата покупки
    totalSum*
                           integer
                           title: Сумма чека
    fn
                           string
                           nullable: true
                           title: ФH
                           maxLength: 20
                           Номер фискального накопителя
    fp
                           string
                           nullable: true
                           title: ΦΠД
                           maxLength: 20
                           Фискальный признак документа
                           string
    fd
                           nullable: true
                           title: ФД
                           maxLength: 20
                           Номер фискального документа
                           string
    orgInn*
                           title: ИНН организации
                           maxLength: 20
    orgName*
                           string
                           title: Название организации
                           maxLength: 20
   orgAddress
                           string
                           nullable: true
title: Юр. адрес организации
                           maxLength: 20
    createdBy*
                           integer
                           readOnly: true
                           title: Кем создан
    updatedBy*
                           integer
                           readOnly: true
                           title: Кем обновлён
    users*
                           Пользователи > [...]
}
```

```
CheckItem ∨ {
   id*
                          integer
                           title: ID moβapa
   createdAt*
                          string($date-time)
                           readOnly: true
title: Дата создания
   updatedAt*
                           string($date-time)
                           readOnly: true
                           title: Дата обновления
                           string
                           title: Наименование
                           maxLength: 1000
   price*
                           integer
                           title: Цена
   quantity*
                          string($decimal)
                           pattern: ^\d{0,5}(?:\.\d{0,10})?$
title: Количество
   sum*
                           integer
                           title: Сумма
   createdBy*
                           integer
                           readOnly: true
                           title: Кем создан
   updatedBy*
                           integer
                           readOnly: true
                           title: Кем обновлён
   receipt*
                           integer
                           title: Чек
   parts*
                           Пользователи > [...]
```

Модель товара

Ассоциативная сущность: товар, пользователь, количество



Методы для работы с пользователем

```
Profile > {
   id*
                          integer
                          readOnly: true
                          string
   username*
                          title: Имя пользователя
                          readOnly: true
                          Обязательное поле. Не более 150 символов. Только буквы, цифры и символы @/./+/-/_.
   email*
                          string($email)
                          title: E-Mail
                          readOnly: true
   firstName
                          string
                          title: Имя
                          maxLength: 150
   lastName
                          string
                          title: Фамилия
                         maxLength: 150
   avatarUrl*
                         string($uri)
                          readOnly: true
   checkScanToken
                          string
                          nullable: true
                          title: Токен ЧекСкан
```

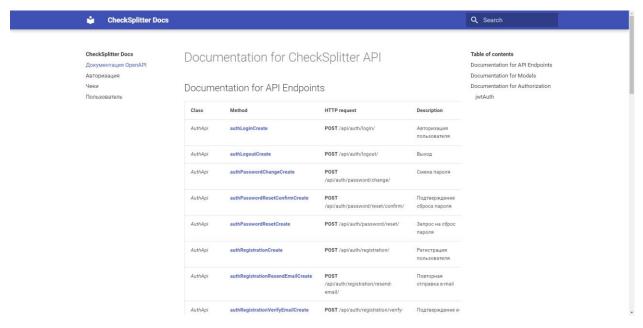
Модель пользователя



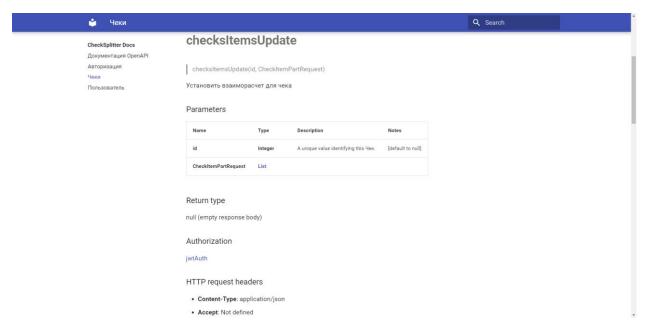
Методы библиотеки авторизации (dj-rest-auth)

#### **Часть 3.** Документация с помощью MkDocs

В результате выполненной работы была изучена система документации MkDocs. Ввиду того, что имелась полная API схема, было принято решение воспользоваться пакетом <u>openapi-generator</u> для формирования файлов в формате Markdown.



Документация MkDocs



Документация MkDocs

## Вывод

В результате выполненной работы были получены практические навыки по использованию DRF и MkDocs.