

Информация для изучения

1. Изучение Python 3+

- Структуры и типы данных
- Отлов ошибок
- Тестирование с помощи библиотеки unittest

2. Классы в Python

- Переопределение методов и метод super
- «Магические методы»

3. HTTP протокол

- Заголовки: Allow, Content-Type, Authorization, Accept, User-Agent
- Статусы ответа сервера: 400, 401, 403, 404, 500, 501
- Методы запроса GET, POST, DELETE, PUT, PATCH

4. Изучение Django 2.2 (только эти разделы)

- Модели. Типы полей и их свойства. Работа с моделями. Можно прочитать про менеджеры моделей
- Миграции
- Представления
- Роутеры

5. Изучение Django Rest Framework

- Сериализаторы данных(Serializers)
- Валидаторы
- Viewsets, Views, Generic views

Задание

Разработать сервер предоставляющий REST API сервисы, реализующее функционал для работы с данными моделей: **Человек, Студент, Преподаватель, Предмет обучения, Группа обучаемых, Занятие**. Все сервисы покрыть тестами.

Описание:

Для разработки рекомендуется использовать Pycharm

Описать модели на фреймворке Django 2.2 на Python 3+ со следующими полями.

«Человек» Абстрактная модель

- ФИО тип varchar длиной 255 символов

«Студент» (наследуется от класса «Человек»)

- Наследуемые поля

«Преподаватель» (наследуется от класса «Человек»)

- Наследуемые поля
- ученая степень тип varchar длиной 255 символов

«Предмет обучения»

- наименование тип varchar длиной 255 символов

«Группа обучаемых»

- наименование тип varchar длиной 255 символов
- перечень идентификаторов студентов(многие ко многим)

«Занятие»

- предмет обучения(внешний ключ)
- преподаватель(внешний ключ)
- группы обучаемых(внешний ключ)
- дата проведения – тип дата и время
- состояние занятия – тип логический. Отвечает за признак того что занятие завершено
- перечень идентификаторов отсутствующих студентов(многие ко многим)

Создать следующие REST API в соответствии с методами HTTP протокола:

CRUD - это сокращенная аббревиатура означающая что сервис должен предоставлять операции Создания, Чтения данных, Изменение, Удаление. При создании сервисов рекомендую использовать класс ModelViewSet(из Django Rest Framework)

Для работы с моделью «Студент»

1. Операции CRUD

Для работы с моделью «Преподаватель»

1. Операции CRUD

Для работы с моделью «Предмет обучения»

1. Операции CRUD
2. Получение списка занятий по одному предмету обучения

Для работы с моделью «Группа обучаемых»

1. Операции CRUD
2. Получения списка студентов в одной группе

Для работы с моделью «Занятие»

1. Создание
2. Получение списка
3. Получение одной записи
4. Изменение
5. Фильтрация по группе обучаемых, предмету, преподавателю
6. Получения списка актуальных занятий(занятие не завершено)

При отправке задания отсылать архив с проектом, внутри которого будет лежать виртуальное окружение.