

**teletype**

@gimntut

Подписаться

20 октября 2022

# Вопросы по Django Rest Framework

## Вводная

### Модели (упрощено)

```
class Entity(Model):
    modified_by = ForeignKey('User', ...)
    value = IntegerField(...)
    properties = ManyToMany('Property')

class Property(Model):
    key = CharField(...)
    value = CharField(...)
```

User - стандартная модель django

### Сериализатор (упрощено)

```
class EntitySerializer(ModelSerializer):
    value = IntegerField(...)
    properties = ???(read_only=True)
```

## Требования к ответам

Общая задача: написать такое api, для которого может автоматически быть сгенерировано OpenApi-описание.

Ответы учитывающие совместимость с drf-spectacular, оцениваются более высоко

Все ответы должны быть полноценными фрагментами кода (желательно, рабочего кода)

Ожидаемые ответы полагаются на возможности DRF и не предусматривают сырую обработку данных в методах `to_internal_value`, `to_representation` или в `__init__`

## Вопросы

### 1. Как правильно сохранять `modified_by`?

Пояснение: при создании записи в запросе приходит только `value`, но в БД нужно записать не только `value`, но и того пользователя, который сделал POST-запрос.

Подсказка: Модели и сериализатор остаются неизменными

### 2. Для создания Entity на вход POST API подаётся json вида

```
{"data[value]": 10}
```

**Как исправить сериализатор так, чтобы он мог принять поле `"data[value]"` и сохранить его в поле `value` ?**

Пояснение: Python не позволит написать в сериализаторе `data[value] = IntegerField( ... )`, но есть другое решение

Подсказка: Модели остаются неизменными. Нужна возможность управлять, как обычным полем, например, чтобы в будущем добавить валидацию и другие возможности доступные через `IntegerField`

### 3. Как вывести `properties` в формате `{key:value, ... }`, если мы заранее не знаем сколько и каких `key` может быть?

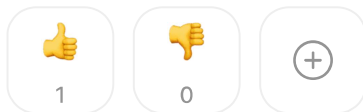
Пояснение: Иногда нужно вывести данные, когда имена полей заранее неизвестны. См. пример ниже. Не обращайте внимания на то, что `value` - строка, это всего лишь пример, как может выглядеть `properties`.

```
[
  {
    "value": "circle",
    "properties": {
      "center": "100, 100",
      "radius": "50"
    }
  },
  {
    "value": "line",
    "properties": {
      "start": "150, 50",
      "end": "50, 150"
    }
  },
]
```

```
{
  "value": "Медведь",
  "properties": {
    "класс": "Млекопитающие"
  }
},
{
  "value": "rectangle",
  "properties": {
    "corner_1": "50, 50",
    "corner_2": "150, 150"
  }
}
]
```

Есть много способов решить эту задачу, нужно выбрать максимально правильный способ

@gimntut · 20 октября 2022, 03:54 · 1027 просмотров · 1 реакция



## Оставьте комментарий

