

Шпаргалка: типы related-полей в сериализаторах

PrimaryKeyRelatedField

Связь по ID. Используется, когда нужно получить не связанные значения, а список ID этих значений. Работает и на чтение, и на запись.

Пример реализации:

```
from rest_framework import serializers

from .models import Achievement, Cat, Owner

class OwnerSerializer(serializers.ModelSerializer):
    cats = serializers.PrimaryKeyRelatedField(many=True, read_only=True)

    class Meta:
        model = Owner
        fields = ('first_name', 'last_name', 'cats')
```

Пример ответа:

```
{
    ...
    'cats': [
        74,
        75,
        76,
        ...
    ]
}
```

HyperlinkedRelatedField

Работает так же, как `PrimaryKeyRelatedField`, но отдаёт полный URL (гиперссылку) на ID; по ней можно перейти и запросить информацию об объекте. Работает и на чтение, и на запись.

Для работы в режиме записи необходимо дополнительно передать аргумент

`queryset`, например `cats = serializers.HyperlinkedRelatedField(many=True, queryset=Cat.objects.all())`

Пример реализации:

```
from rest_framework import serializers

from .models import Achievement, Cat, Owner

class OwnerSerializer(serializers.ModelSerializer):
    cats = serializers.HyperlinkedRelatedField(many=True, read_only=True)

    class Meta:
        model = Owner
        fields = ('first_name', 'last_name', 'cats')
```

Пример ответа:

```
{
  ...
  'cats': [
    'http://127.0.0.1:8000/cats/74',
    'http://127.0.0.1:8000/cats/75',
    'http://127.0.0.1:8000/cats/76',
    ...
  ]
}
```

SlugRelatedField

`SlugRelatedField` принимает аргумент `slug_field` и делает из него `slug`, который и отдает в поле. Работает и на чтение, и на запись.

Для работы в режиме записи необходимо дополнительно передать аргумент

`queryset`, например `cats = serializers.SlugRelatedField(many=True, queryset=Cat.objects.all())`

Пример реализации:

```
from rest_framework import serializers

from .models import Achievement, Cat, Owner
```

```
class OwnerSerializer(serializers.ModelSerializer):
    cats = serializers.SlugRelatedField(
        many=True,
        read_only=True,
        slug_field='name'
    )

    class Meta:
        model = Owner
        fields = ('first_name', 'last_name', 'cats')
```

Пример ответа:

```
{
  ...

  'cats': [
    'Leopold',
    'Begemot',
    'Matroskin',
    ...
  ]
}
```