

# OpenAPI широкого назначения

Александр Клименко

# Что такое ЖКХ?

$$P_i = \left( V_i + \frac{S_i \times (V^{\text{д}} - \sum V_i)}{S_{\text{об}}} \right) \times T^T, \quad P_i = \left( V_i^{\text{н}} + V_i^{\text{одн}} \times \frac{S_i}{S_{\text{об}}} \right) \times T^T,$$
$$V_i^{\text{одн}} = V^{\text{д}} - \sum_i V_i^{\text{н}},$$

$$P_i^{\text{о}} = \sum_v \left[ q_v^{\text{кр}} \times \left( V_i^{\text{н}} + V_i^{\text{одн}} \times \frac{S_i}{S_{\text{об}}} \right) \times T_v^{\text{кр}} \right],$$

Потребление воды

$$P_i = S_i \times N^T \times T^1, \quad P_i = \left( V_i + \frac{S_i \times S^{\text{он}} \times N^T \times (S^{\text{об}} - S^{\text{инд}})}{S^{\text{об}} \times (S^{\text{об}} - S^{\text{инд}} + S^{\text{он}})} \right) \times T^T,$$

$$V_i = S_i \times \frac{N^T \times (S^{\text{об}} - S^{\text{инд}})}{S^{\text{об}} - S^{\text{инд}} + S^{\text{он}}},$$

Отопление





Управление  
ресурсами



Разнообразный стек  
технологий



Постоянный  
обмен данными



Взаимодействие с  
приложениями



# Задача — организовать взаимодействие разработчиков

## Требования

- документация как таковая
- актуальность
- минимум трудозатрат
- быстрое развертывание приложений
- человеко- и машино- читаемая
- соответствие корпоративным стандартам API (REST)



<http://bit.ly/2DTgZ1e>

# OpenAPI 3

```
paths:  
  /api/v1/accountingitem-list/  
    get: ...  
    post: ...  
    patch: ...  
    delete: ...
```

# OpenAPI 3: Пути

```
components:  
  schemas:  
    AccountingItem:  
      properties:  
        _uid: ...  
        _type: ...  
        _version: ...  
        name: ...
```

# OpenAPI 3: Компоненты

paths

/api/v1/building-list/{\_uid}/:

...

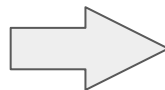
schema {"\$ref": "#/components/schemas/Building"}

# OpenAPI 3: Ссылки

# OpenAPI 3: Чего-то не хватает...

## Доработка вложенного ModelSerializer

```
class BuildingViewSet(...):  
    serializer_class = BuildingSerializer
```

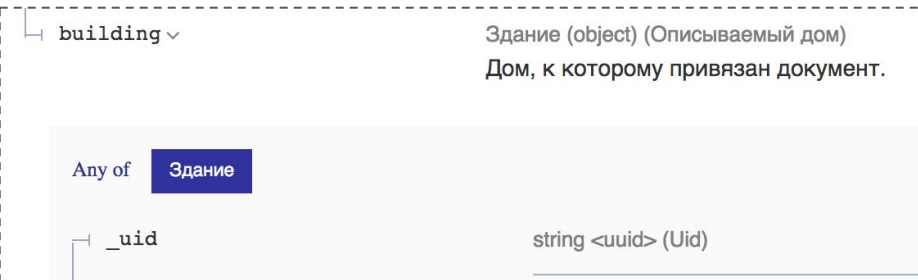
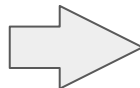


Получить объект Здание

GET /api/v1/building-list/{\_uid}/

Здание - разновидность наземного...

```
class DocumentSerializer(...):  
    building = BuildingSerializer()
```





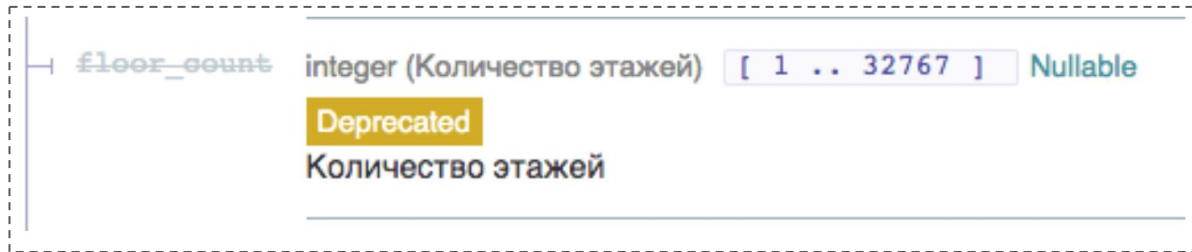
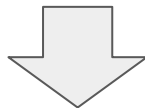
## Человеко-читаемые choices

```
{  
  "building_type": {  
    "type": "number",  
    "enum": [  
      0,  
      1  
    ],  
    "x-enumNames": [  
      "Многоквартирный дом",  
      "Частный дом"  
    ]  
  }  
}
```

```
└─ building_type    number  
    Enum: 0 1  
    enumNames: [ "Многоквартирный дом", "Частный дом" ]
```

## Доработка deprecated

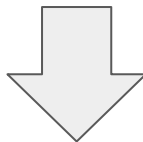
```
class SomeSerializer(...):  
    floor_count = ...  
    class Meta:  
        deprecated = {'floor_count': True}
```



## Доработка SerializerMethod type hints

```
class SomeSerializer(...):  
    is_deleted = SerializerMethod()  
    def get_is_deleted(*args, **kwargs) -> bool:
```

...



```
{  
  "is_deleted": {"type": "boolean"}  
  ...  
}
```



## Доработка ленивых полей сериализера

```
class SomeSerializer(LazyFieldHandler, ...):  
    some_field = LazyField('some_app.serializers.SomeSerializer', null=True)
```

## Доработка кастомизации схемы

```
class SomeSerializer(...):  
    update_schema = {  
        'description': 'Some object description'  
    }  
}
```

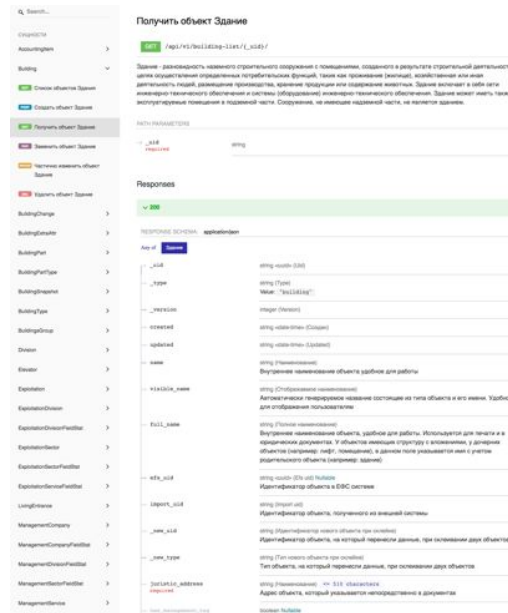
```
class SomeViewSet(...):  
    update_schema = {  
        'description': 'Some view-set description'  
    }  
}
```

```
class SomeSerializer(...):  
    def update_schema(self, schema):  
        ...
```

```
class SomeViewSet(...):  
    def update_schema(self, schema):  
        ...
```

## Преимущества

- всегда актуальная документация API
- механизм репликации структуры данных на уровне приложения
- упрощение разработки клиентских приложений



- Аріқа любить стандарты
- Будь как Аріқа!

# Спасибо!



[telegram.me/meteozond](https://t.me/meteozond)



[github.com/meteozond](https://github.com/meteozond)



[alex@erix.ru](mailto:alex@erix.ru)



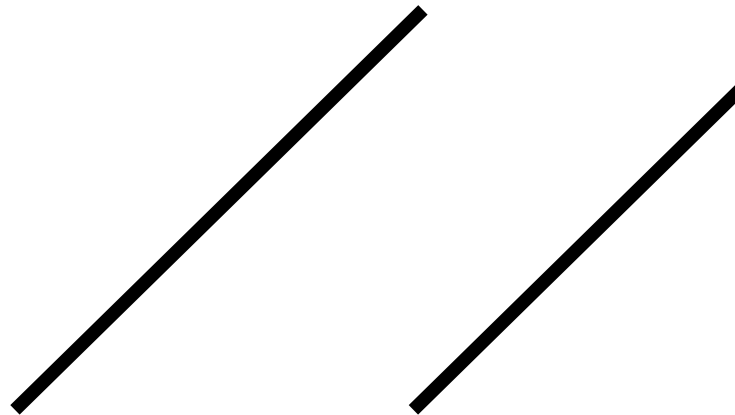
[bit.ly/2PghiZm](https://bit.ly/2PghiZm)



[github.com/pik-software](https://github.com/pik-software)



# Бонус-трек



Обычное поле:

```
Unit: {  
  title: "Подразделение",  
  properties: {  
    name: {  
      type: "string",  
      title: "Название"}}}
```

## Как описать вложенный объект?

```
Unit: {  
  title: "Подразделение",  
  properties: {  
    leader: <object> }}
```

## Как объект

Unit: {

title: "Подразделение", properties: {

leader:

```
{title: "Родительское подразделение",  
  properties: {  
    name: {type: "string", title: "Лидер"}}}}
```

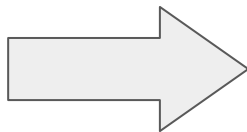
}}

## Вложенный объект как ссылка

```
Unit: {  
  title: "Подразделение",  
  properties: {  
    leader: { $ref: 'Unit' }  
  }  
}
```

## Поведение \$ref

```
{  
  "$ref": "Unit",  
  "title": "Лидер"  
}
```



```
{  
  "title": "Подразделение",  
  "properties": { ... }  
}
```

# Как же описывать поля- ссылки?

## Новые директивы OpenAPI 3

- anyOf - любой из...
- allOf - сумма
- oneOf - только один из (не совпадает с другими) ...



Что использовать  
`anyOf`, `allOf` или `oneOf`?

## Описание через allOf



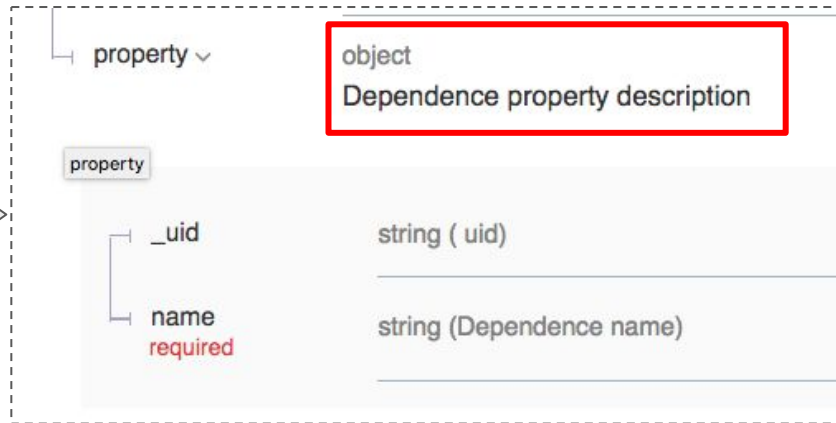
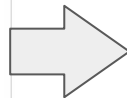
m-mohr on Oct 10 • edited

Contributor Author +😊 ...

@meteozone Oh, sorry, my mistake. I meant allOf instead of anyOf:

```
{
  "properties": {
    "property": {
      "allOf": [
        {
          "$ref": "#/components/schemas/SomeDefinition"
        },
        {
          "title": "Property title",
          "description": "Property description"
        }
      ]
    }
  }
}
```

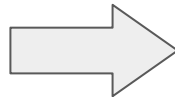
Not sure how that's shown in the UI though.



- <https://github.com/Redocly/redoc/issues/810#issuecomment-540459048>

## Описание через anyOf

```
{  
  "properties":  
  {  
    "property": {  
      "title": "Dependence property title",  
      "description": "Dependence property description",  
      "anyOf": [  
        {  
          "$ref": "#/components/schemas/Dependence"  
        }  
      ]  
    }  
  }  
}
```



property ▾ Dependence title (object) (Dependence property title)  
Dependence property description

Any of **Dependence title**

\_uid string ( uid)

name string (Dependence name)  
required

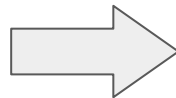
<https://github.com/Redocly/redoc/issues/810#issuecomment-535511029>

## Описание через allOf\*



RomanHotsiy on Oct 10

```
"property": {  
  "title": "Dependence property title",  
  "description": "Dependence property description",  
  "allOf":  
  {  
    "$ref": "#/components/schemas/Dependence"  
  }  
},  
],  
}
```



```
property ▾ object (Dependence property title)  
Dependence property description
```

```
_uid string (uid)  
name string (Dependence name)  
required
```

<https://github.com/Redocly/redoc/issues/1060#issuecomment-540590515>