

jsonschema

Используется для валидации json схемы. По умолчанию дополнительно указанные ключи (не существующие в схеме, но присутствующие в документе) не проверяются.

Установка

```
pip install jsonschema
```

Базовое использование

```
from jsonschema import validate

validate(instance=json_to_check, schema=schema)
```

Исключения

`jsonschema.exceptions.ValidationError` - если документ не соответствует структуре

`jsonschema.exceptions.SchemaError` - если сама схема некорректна

Пример схемы:

```
schema = {
    "type": "object",
    "properties": {
        "name": {"type": "string"},
        "age": {"type": "number"},
    },
    "required": ["name"],
}
```

Данная схема определяет json объект с 2 свойствами: `name` и `age`. Обязательное свойство `name`.

Ключевые слова

Для некоторых типов используются дополнительные ключевые слова.

Ключевое слово	Описание
type	Тип. Для корня часто object. string - строка number - число object - объект array - список
\$defs	Вложенный шаблон для случая, когда шаблон элемента встречается в нескольких местах.
\$ref	Подстановка вложенного шаблона.
\$schema	Ссылка на шаблон шаблона. При обновлении версии библиотеки будет использоваться новый шаблон шаблона, что может привести к проблемам. Желательно указывать.

Дополнительные ключевые слова для типов.

Тип array

Ключевое слово	Описание
items	Тип элементов списка. <div>"scores": { "type": "array", "items": {"type": "number"}, }</div>
minItems	Минимальное количество элементов

Тип object

Ключевое слово	Описание
required	Список обязательных ключей. <div>"required": ["name"]</div>

Ключевое слово	Описание
properties	Определяет ключи объекта и их тип. <div>"properties": { "name": {"type": "string"} },</div>
additionalProperties	Если True, то дополнительно указанные ключи приводят к исключению.

Локальные вложенные шаблоны.

Для определения используется переменная \$defs, для использования - \$ref.

```
schema = {  
  "type": "object",  
  "properties": {  
    "address": {"$ref": "#/$defs/address"},  
  },  
  "$defs": {  
    "address": {  
      "type": "object",  
      "properties": {  
        "street": {"type": "string"},  
      }  
    },  
  },  
}
```