

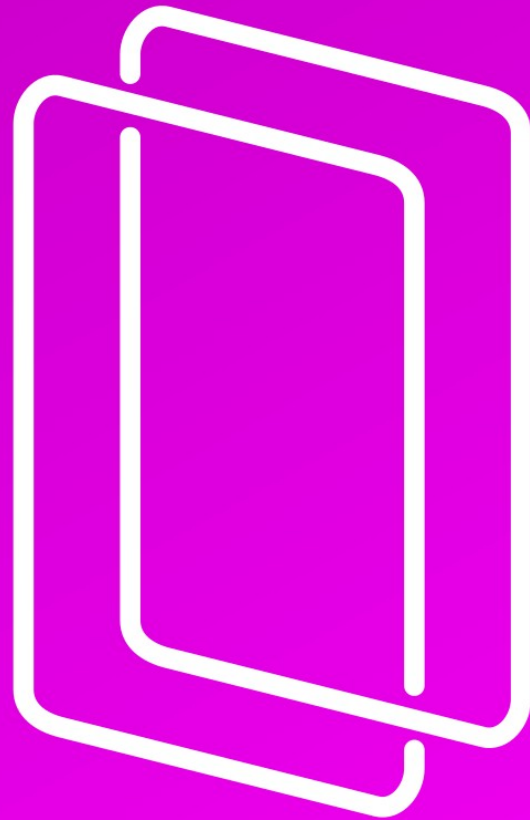
# Навык для Яндекс Алисы “Создай расписание”



Белозеров Ярослав

# Идея проекта:

Расширить возможности Алисы с помощью создания навыка





Приветствие

Создать

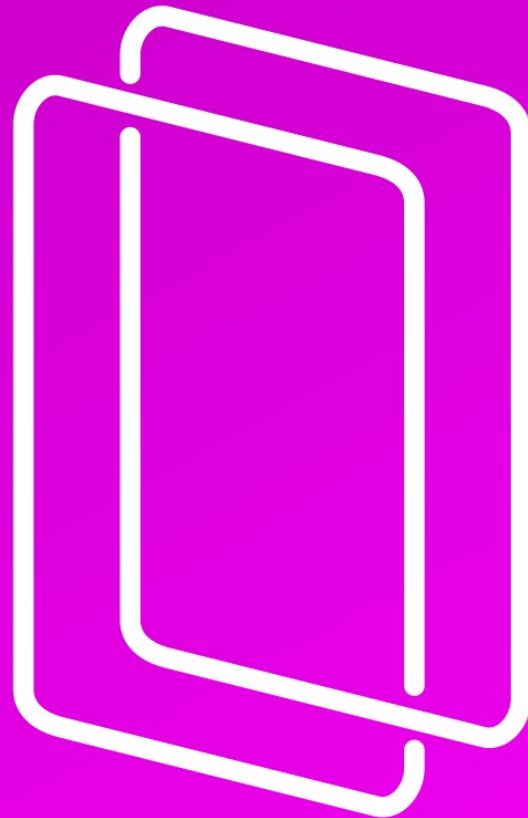
Показать

Удалить

Подтвердить

SQLAlchemy

sqlite



```
class Button:
    """Кнопка для api аlicy"""

    def __init__(self, title, url='', payload=None, hide=True):
        self.title = title # текст кнопки
        self.url = url # ссылка, которая откроется при нажатии
        # словарь, которые передаются навыку, если будет нажата кнопка
        self.payload = {} if payload is None else payload
        self.hide = hide # True - появится на вводе текста, False -

    def __iter__(self):
        # Ф-я нужна для вызова dict(), для отправки ответа к api, при
        builded = {'title': self.title,
                  'hide': self.hide}
        if self.url:
            builded['url'] = self.url
        if self.payload:
            builded['payload'] = self.payload
        for key, value in builded.items():
            yield (key, value)
```

class Message:

"""Обёртка для отправки и получения сообщений от api аlicy"""

```
def __init__(self, message):
    self.message = message
    self.request = message['request']
    self.meta = message['meta']
    self.session = message['session']
    self.version = message['version']
    self.text = ''
    self.tts = ''
    self.buttons = []
    self.is_end = False
```

```
def build_response(self):
    # строит ответ к api, просто складывает все внесенные значения в dict
    response = {
        "version": self.message['version'],
        "session": self.message['session'],
        "response": {
            "text": self.text,
            "end_session": self.is_end,
        }
    }
    if self.tts:
        response['response']['tts'] = self.tts
    if self.buttons:
        response['response']['buttons'] = [dict(button) for button in self.buttons]
    return response
```

```
def get_cmd(self):
    return self.request['command'] # возвращает команду пользователя
```

```
def get_orig_text(self):
    return self.request['original_utterance'] # оригинальный текст пользователя
```

# Возможности для развития:

- увеличить спектр воспринимаемых команд
- усовершенствовать озвучивание информации (синтезатор)
- добавить больше типов периодичности для событий

