

# Yandex cloud function

Примечание. Точка входа должна называться по названию файла и главной функции в данном случае `index.handler` и должна стоять включена галочка напротив `Напротив публичная функция`

Файл `index.py`. Для перевода используется <https://api.mymemory.translated.net/get> единственный на данный момент работающий API.

```
import telebot
import requests
import json

API_TOKEN = 'я вам не скажу ключ, прошу прощения'
bot = telebot.TeleBot(API_TOKEN)

def translate(text):
    url = "https://api.mymemory.translated.net/get"
    params = {
        'q': text,
        'langpair': 'ru|en'
    }
    try:
        response = requests.get(url, params=params)
        translation = response.json()
        return translation['responseData']['translatedText']
    except Exception as e:
        print(f"Translation error: {e}")
        return "Translation service unavailable"

@bot.message_handler(commands=['start'])
def send_welcome(message):
    bot.reply_to(message, """\
Привет я бот переводчик с Русского на Английский. Напиши и я переведу!\
""")

@bot.message_handler(commands=['help'])
def send_help(message):
    bot.reply_to(message, """\
Помощи нет, просто напиши что нибудь!\
""")

@bot.message_handler(func=lambda message: True)
def echo_message(message):
    bot.reply_to(message, translate(message.text))

def handler(event, context):
    message = telebot.types.Update.de_json(event['body'])
    bot.process_new_updates([message])
    return {
```

```

    'statusCode': 200,
    'body': 'Hello World!',
}

```

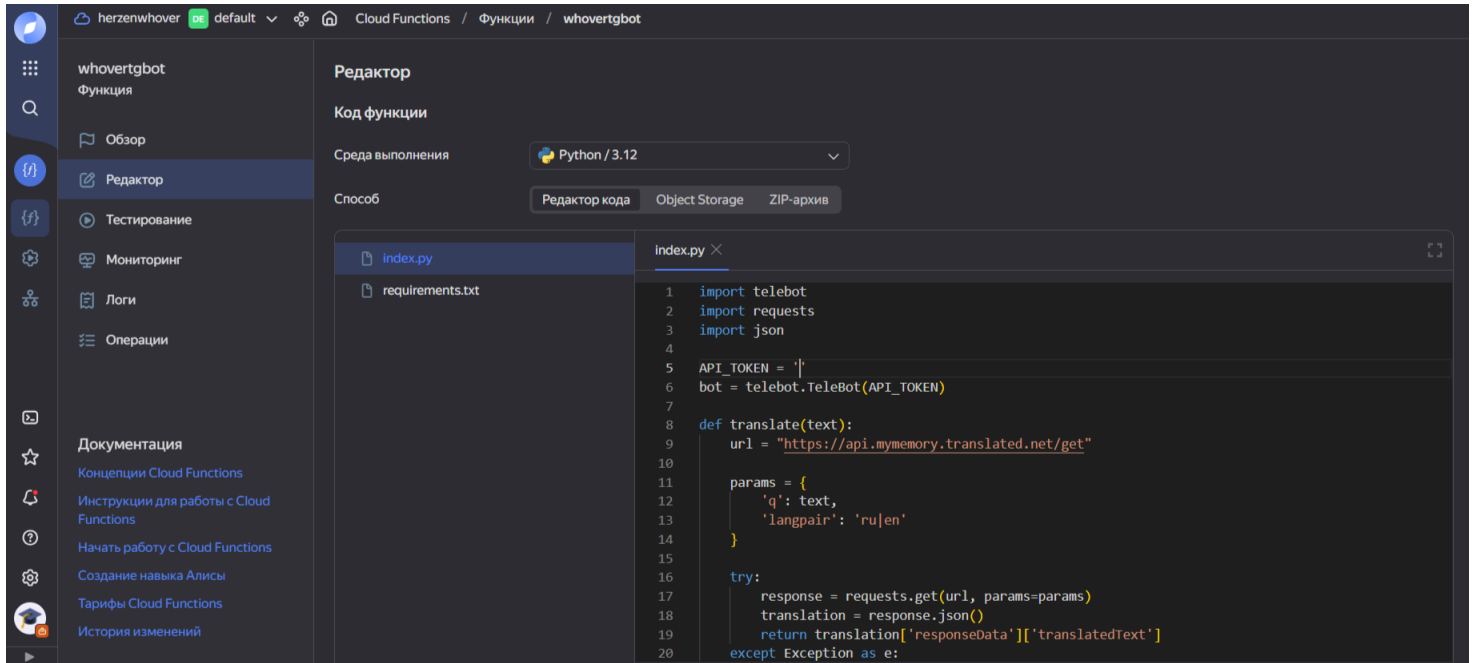
Файл requirements.txt

```

pyTelegramBotAPI
requests

```

Скрин подтверждение.



## API gateway

```

openapi: 3.0.0
info:
  title: Telegram Bot API
  version: 1.0.0
servers:
  - url: https://d5dc3gvpnggqh85iks0l.trruwy79.apigw.yandexcloud.net
paths:
  /:
    post:
      x-yc-apigateway-integration:
        type: cloud_functions
        function_id: специально_удалена
        operationId: tgbot
  /login:
    get:
      x-yc-apigateway-integration:
        type: dummy
        content:
          '*': "1149912"

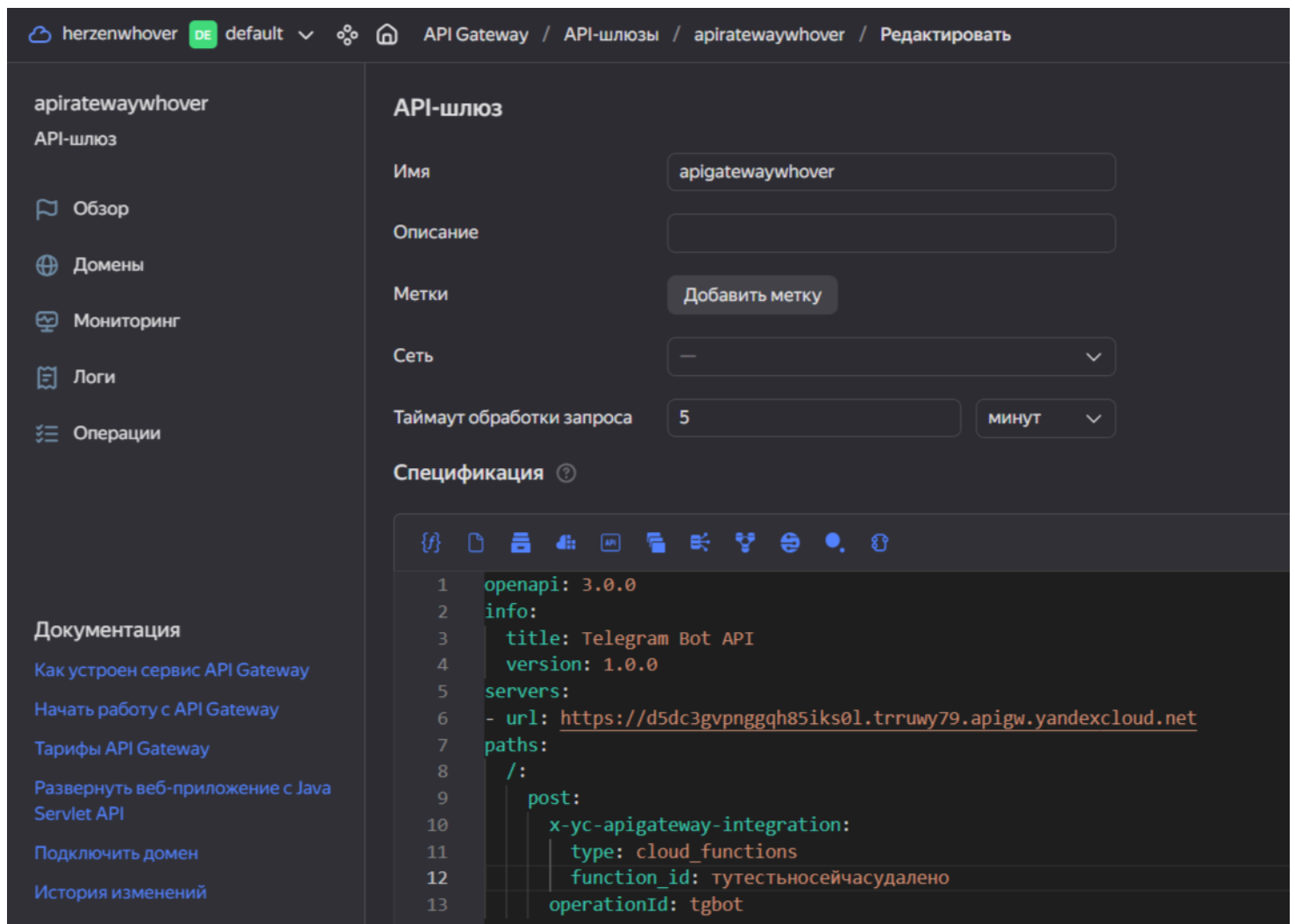
```

http\_code: 200

http\_headers:

Content-Type: "text/plain; charset=UTF-8"

Access-Control-Allow-Origin: "\*"



The screenshot shows the API Gateway console interface. On the left is a sidebar with navigation links: 'API-шлюз', 'Обзор', 'Домены', 'Мониторинг', 'Логи', 'Операции', and a 'Документация' section with links for service overview, getting started, pricing, Java Servlet API deployment, domain connection, and change history. The main panel is titled 'API-шлюз' and shows configuration for 'apigatewayerwhover'. Fields include 'Имя' (apigatewayerwhover), 'Описание' (empty), 'Метки' (Add tag button), 'Сеть' (dropdown), and 'Таймаут обработки запроса' (5 минут). Below is the 'Спецификация' (OpenAPI spec) editor showing a JSON configuration for a Telegram Bot API.

```
1 openapi: 3.0.0
2 info:
3   title: Telegram Bot API
4   version: 1.0.0
5 servers:
6   - url: https://d5dc3gvpnggqh85iks0l.trruwy79.apigw.yandexcloud.net
7   paths:
8     /:
9       post:
10         x-yc-apigateway-integration:
11           type: cloud_functions
12           function_id: тутестьносейчасудалено
13           operationId: tgbot
```

## Webhook

```
curl.exe `
  --request POST `
  --url https://api.telegram.org/bot<токенудален>/setWebhook `
  --header '"Content-type: application/json"' `
  --data '{" \url\":
\ "https://d5dc3gvpnggqh85iks0l.trruwy79.apigw.yandexcloud.net\" }"'
```

## Telegram бот

Работающего бота можно найти в телеграме по нику

@Egor\_hate\_this\_translatorKP\_Bot или по ссылке

