

**Министерство цифрового развития, связи и массовых коммуникаций  
Российской Федерации**  
**Ордена Трудового Красного Знамени федеральное государственное бюджетное  
образовательное учреждение высшего образования**  
**«Московский технический университет связи и информатики»**

**Кафедра математическая кибернетика и информационные технологии**  
**Отчёт по лабораторной работе №2**

«»

Выполнил: студент группы  
БИН2001 Фадеев А. М.  
Проверил: Аршинов Е. А.

В данной работе использованы возможности сервиса Microsoft Azure.  
Для более удобной работы с модулями создадим виртуальное окружение.

```
python3 -m venv env
```

После чего перейдём в него

```
arch@BTWUseArch ~/code/institut/vvedenie_v_it/IT_labs/lab2/sources main ± source ./env/bin/activate
(env) arch@BTWUseArch ~/code/institut/vvedenie_v_it/IT_labs/lab2/sources main ±
```

В файле app.py импортируем необходимые библиотеки, подгрузим переменные окружения из файла .env

```
2 import requests
3 import os
4 import uuid
5 import json
6 from dotenv import load_dotenv
7 load_dotenv()
8 from flask import Flask, render_template, request
```

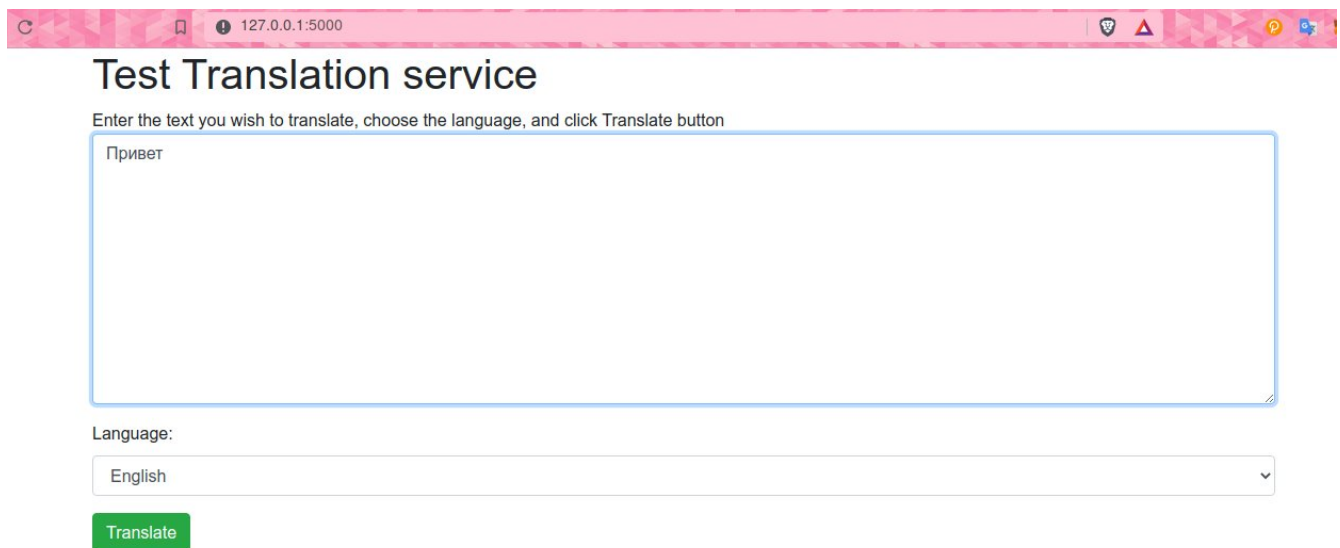
```
1 KEY=e1b1d244abf3403b91c26d9c5b75082f
2 ENDPOINT=https://api.cognitive.microsofttranslator.com/
3 LOCATION=westeurope
```

Опишем обработку POST/GET запросов

```
12 @app.route('/', methods=['GET'])
13 def index():
14     return render_template('index.html')
15
16 @app.route('/', methods=['POST'])
17 def index_post():
18     original_text = request.form['text']
19     target_language = request.form['language']
20     key = os.environ['KEY']
21     endpoint = os.environ['ENDPOINT']
22     location = os.environ['LOCATION']
23     path = '/translate?api-version=3.0'
24     target_language_parameter = '&to=' + target_language
25     constructed_url = endpoint + path + target_language_parameter
26     headers = {
27         'Ocp-Apim-Subscription-Key': key,
28         'Ocp-Apim-Subscription-Region': location,
29         'Content-type': 'application/json',
30         'X-ClientTraceId': str(uuid.uuid4())
31     }
32     body = [{'text': original_text}]
33     translator_request = requests.post(
34         constructed_url, headers=headers, json=body)
35     translator_response = translator_request.json()
36     translated_text = translator_response[0]['translations'][0]['text']
37     return render_template(
38         'results.html',
39         translated_text=translated_text,
40         original_text=original_text,
41         target_language=target_language
42     )
```

Результат запуска

```
(env) arch@BTWUseArch ~/code/institut/vvedenie_v_it/IT_labs/lab2/sources ❷ main ± flask run
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```



Test Translation service

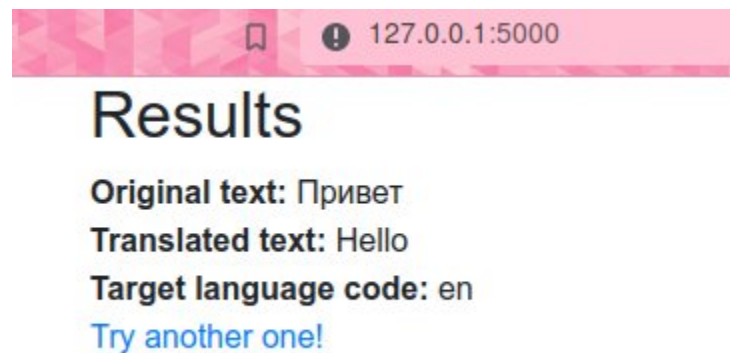
Enter the text you wish to translate, choose the language, and click Translate button

Привет

Language: English

Translate

По нажатию кнопки «Translate», был произведён запрос на Azure, откуда был получен ответ в виде переведённого слова и выведен в ответе скрипта на POST запрос.



Results

**Original text:** Привет

**Translated text:** Hello

**Target language code:** en

[Try another one!](#)

Перед добавлением файлов в git внесём изменения в .gitignore файл

```
103 # Environments
104 .env
105 .venv
106 env/
107 venv/
108 ENV/
109 env.bak/
110 venv.bak/
```