

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

ОТЧЁТ ПО ПРАКТИКЕ
ЗАЩИЩЁН С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

Старший преподаватель

должность, уч. степень, звание

М.Д. Поляк

подпись, дата

инициалы, фамилия

ОТЧЁТ ПО ПРАКТИКЕ

вид практики производственная практика

тип практики научно-исследовательская работа

на тему индивидуального задания «Разработка backend-приложения на языке python для
выполнения геокодирования в рамках географической области Санкт-Петербурга»

выполнен Валиевым Михаилом Робертовичем

фамилия, имя, отчество обучающейся в творительном падеже

по направлению подготовки 09.00.00

код

Информатика и вычислительная техника

наименование направления

направленности 09.03.04

код

Программная инженерия

наименование направленности

Проектирование программных систем

наименование направленности

Обучающаяся группы № 4031

номер

подпись, дата

М.Р. Валиев

инициалы, фамилия

Санкт–Петербург 2023

Введение

В рамках индивидуального задания производственной практики на тему «разработка backend-приложения на языке python для выполнения геокодирования в рамках географической области Санкт-Петербурга» необходимо установить сервер nominatim на Linux, скачать данные области Санкт-Петербург, импортировать их в БД postgresql, используя утилиту osm2pgsql; написать python-сервер, который будет получать координаты, запрашивать адрес по координатам установленный сервер nominatim и отправлять в ответ клиенту полученный адрес.

1. Установка Nominatim

Nominatim (лат. «по имени») — инструмент для поиска данных OSM по имени и адресу, а также для создания полных адресов точек OSM (обратное геокодирование).

Nominatim API в сети имеет (*nominatim.openstreetmap.org*) ограниченную мощность и, в связи с этим) ряд ограничений (например: максимум 1 запрос в секунду) [1]. Чтобы обойти эти ограничения, можно установить сервер nominatim на свой ЭВМ.

Nominatim API имеет несколько запросов. Для обратного геокодирования необходимо использовать `/reverse` [2].

Инструкцию по установке сервера Nominatim можно найти на официальном сайте Nominatim [3] (лучше выполнять по этой ссылке на youtube [4]. В описание к видео есть ссылка на команды терминала).

2. Импортирование данных

После установки Nominatim на Linux, нужно импортировать данные из osm в базу данных Postgres. Для этого нужно экспортировать данные нужной области в формате *.osm или *.pbf из bbbike [5].

Bbbike позволяет получить данные в выбранном формате определенной области из Open Street Map. В нашем случае область - Санкт-Петербург.

Для работы с географическими данными в Postgresql требуется расширение PostGIS. Следуя инструкции по установке Nominatim [3] (и [4]), это расширение будет установлено.

Импортирование данных в Postgresql происходит с помощью утилиты osm2pgsql. После установки Nominatim, это можно сделать с помощью команды *nominatim import --osm-file <data file> 2>&1 | tee setup.log*. Этот шаг также прописан в [3] и [4].

3. Разработка сервера на python

Сервер получает get-запрос с двумя параметрами: широты и долготы, и отправляет get-запрос `/reverse?lat=&lon=&format=json&addressdetails=1` [2] на сервер nominatim (<http://localhost:80/nominatim/>). Полученный текст преобразуется в json объект. Сервер отправляет ответ клиенту текст атрибута *display_name*, в котором содержится адрес по запрошенным координатам. Сервер реализован с помощью Flask API [6]. Код представлен в ПРИЛОЖЕНИЕ А, а также по ссылке [7].

Выводы

В ходе выполнения индивидуального задания были получены навыки работы с географическими данными. Я узнал о ресурсе Open Street Map, который представляет собой карту всего мира в открытом доступе. Научился пользоваться Nominatim API. Получил больше опыта работы в ОС Linux. Написал маленькое REST API приложение на языке python с использованием flask.

Список используемых источников

1. Политика использования Nominatim API. URL: <https://operations.osmfoundation.org/policies/nominatim/>
2. Nominatim API. URL: <https://nominatim.org/release-docs/develop/api/Overview/>
3. Инструкция по установке Nominatim. URL: <https://nominatim.org/release-docs/develop/admin/Installation/>
4. How to Install Nominatim 4.2.0 on Ubuntu Server 22.04 - Own Geocoder API. URL: https://www.youtube.com/watch?v=75Me-_3heM4
5. Bbbike. URL: <https://extract.bbbike.org/>
6. Flask. URL: <https://flask.palletsprojects.com/en/2.3.x/>
7. Ссылка на github проекта. URL: <https://github.com/1Mercury1/Production2023.git>

ПРИЛОЖЕНИЕ А

main.py

```
from flask import Flask, render_template, request
import requests

# границы импортированных данных
bllat = 59.634 # bottom left latitude
bllon = 29.525

trlat = 60.24 # top right latitude
trlon = 30.798

app = Flask(__name__)

@app.route("/")
def index():
    return render_template('index.html')

@app.route('/submit_coordinates', methods=['GET'])
def submit_coordinates():
    latitude = request.args.get('latitude')
    longitude = request.args.get('longitude')

    lat_num = float(latitude)
    lon_num = float(longitude)

    if(lat_num > trlat or lat_num < bllat or lon_num > trlon or lon_num < bllon):
        return "Заданные координаты за пределами Санкт-Петербурга"

    nominatim_reverse_url =
f'http://localhost:80/nominatim/reverse?lat={latitude}&lon={longitude}&format=json&addressdetails=1'
    response = requests.request("Get", nominatim_reverse_url).json()

    return response['display_name']

if __name__ == '__main__':
    app.run(debug=True)
```

index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Geocoder spb</title>
```



```
</head>

<body>
  <h1>Введите координаты:</h1>
  <form action="/submit_coordinates" method="GET">
    Широта: <input type="text" name="latitude"><br>
    Долгота: <input type="text" name="longitude"><br>
    <input type="submit" value="Получить адрес">
  </form>
  <div id="address"></div>

  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script>
    $('form').submit(function (e) {
      e.preventDefault();
      var latitude = $('input[name="latitude"]').val();
      var longitude = $('input[name="longitude"]').val();

      $.ajax({
        url: '/submit_coordinates',
        type: 'GET',
        data: { latitude: latitude, longitude: longitude },
        success: function (response) {
          $('#address').text(response);
        }
      });
    });
  </script>
</body>

</html>
```