

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

Ст. преп.

М.Д. Поляк

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЁТ ПО ПРАКТИКЕ

вид практики      производственная

тип практики      научно-исследовательская работа

на тему индивидуального задания      Разработка серверного приложения для прямого гео-  
кодирования в рамках географической области Санкт-Петербурга

выполнен      Бабковым Савелием Викторовичем

фамилия, имя, отчество обучающегося в творительном падеже

по направлению подготовки      09.03.04

код

Программная инженерия

наименование направления

направленности

наименование направления

Проектирование программных систем

код

наименование направленности

наименование направленности

Обучающийся группы №      4031

номер

подпись, дата

С.В. Бабков

инициалы, фамилия

Санкт-Петербург 2023

## Содержание

Содержание .....	2
Цель работы .....	3
Исходные данные .....	4
Разработка приложения .....	5
Конфигурация сервера .....	6
Тестирование .....	7
Выводы .....	9
Список использованной литературы.....	10
Приложение .....	11

## **Цель работы**

Разработать клиент-серверное веб-приложение для прямого геокодирования.

Под прямым геокодированием понимается процедура получения географических координат используя административный адрес.

В качестве данных для геокодирования воспользоваться открытым банком данных [openstreetmaps.org](https://openstreetmaps.org).

## **Исходные данные**

Для геокодирования используется Nominatim. Nominatim – это сервис, предоставляющий возможность поиска адресов в базе данных open-streetmaps.org и получения развернутой информации о них (в т.ч. географических координат). Он может использоваться через API для получения данных с удаленных серверов, но также предоставляет возможность импорта базы данных на собственный сервер. Важной особенностью является то, что существует возможность масштабирования данных (работы только с определенным регионом), так как работа с базой данных адресов со всего мира требует мощных аппаратных ресурсов. В рамках данной работы обрабатываются только адреса в Северо-Западном Федеральном округе России.

Для реализации веб интерфейса используется Python-приложение с использованием фреймворка Flask.

## **Разработка приложения**

Разработанное приложение представляет собой простейшее Flask приложение. Оно состоит из функции-обработчика GET запросов и функции геокодирования. Функция-обработчик принимает запрос с параметрами «улица» и «номер дома», и передает эти данные в функцию геокодирования. Эта функция, в свою очередь, формирует запрос к Nominatim согласно Nominatim API, выбирает из возвращенных данных географические координаты, и возвращает их функции-обработчику, которая возвращает их пользователю в простейшем (два десятичных числа), или в «красивом» (html-страница, собранная из шаблона) виде.

По сути, приложение представляет собой прокси-сервер между пользователем и nominatim, который позволяет передавать и получать данные в более простом виде.

## Конфигурация сервера

Сервер для геокодирования разворачивается в облаке Yandex.Cloud[1]. Этот сервис предоставляет внешний IP-адрес, по которому можно обратиться к виртуальной машине.

Конфигурацию сервера, в соответствии с официальным руководством [2], можно разделить на несколько этапов:

1. Установка необходимого ПО, куда входит:

- СУБД PostgreSQL;

- Система сборки CMake

- Интерпретатор Python

- Непосредственно сервер Nominatim

2. Сборка и конфигурация Nominatim

Для сборки используется инструмент CMake, исходные файлы получены с официального сайта [3]

3. Импорт данных об адресах в локальную базу данных

Данные в виде .osm файлов доступны на сайте Geofabrik[4], после скачивания nominatim автоматически импортирует их в базу данных и подготавливает к использованию.

4. Конфигурация веб -сервера.

Вместе с nominatim поставляются .php файлы, которые реализуют nominatim API. Необходимо настроить веб-сервер так, чтобы к этим файлам можно было получить доступ по протоколу HTTP. В данной работе используется веб сервер nginx[5].

## Тестирование

Для ручного тестирования применялась html-страница с формой:

### Get coordinates from address

Street:

Number:

Pretty: ☐

Результаты тестирования:

### Get coordinates from address

Street:

Number:

Pretty: ☒

#### Coordinates of the given address:

Latitude: 59.877995

Longitude: 30.313824948349644

Full address: Общежитие ГУАП, 8, Варшавская улица, округ Московская застава, Санкт-Петербург, Северо-Западный федеральный округ, 196128, Россия

**Варшавская улица, 8**  
Варшавская улица, 8, Санкт-Петербург, 196105  
Координаты: **59.877935, 30.313910** ...

### Get coordinates from address

Street:

Number:

Pretty: ☐

59.9301179 30.2975855

## Большая Морская улица, 67

Большая Морская улица, 67, Санкт-Петербург,  
190000

Координаты: **59.929560, 30.296671** ...



## **Выводы**

В ходе работы было разработано клиент-серверное приложение-геокодер. Для реализации геокодирования был использован собственный сервер на виртуальной машине, использующий сервис nominatim. при геокодировании использовался открытый банк данных openstreetmaps. Для реализации собственного клиент-серверного приложения использовался python и фреймворк Flask.

Критический анализ:

Преимущества:

1. Автономность. Приложение не использует никакие сторонние сервисы, все необходимые программы и данные локализованы на собственном сервере;
2. Простота в использовании. В соответствии с заданием программа реализует только примитивный функционал, не отвлекая внимание на лишнюю информацию.

Недостатки:

1. Неопределенность. В базе данных адреса ищутся простым поиском по тексту, что может привести к неожиданным результатам при неточном вводе адреса;
2. Быстродействие. Время выполнения запросов к базе данных довольно велико при сравнительно небольшом объеме данных. При возникновении необходимости масштабирования на большую территорию будет необходима более точная настройка базы данных.

### Список использованной литературы

1. Яндекс.Облако – URL: <https://cloud.yandex.ru/>(дата обращения: 21.07.2023).
2. Официальное руководство по установке nominatim – URL: <https://nominatim.org/release-docs/latest/appendix/Install-on-Ubuntu-22/> (дата обращения: 21.07.2023).
3. Официальный сайт nominatim – URL: <https://nominatim.org/> (дата обращения: 21.07.2023).
4. Сервер загрузки Geofabrik – URL: <https://download.geofabrik.de/> (дата обращения: 21.07.2023).
5. Официальный сайт nginx – URL: <https://nginx.org/ru/>(дата обращения: 21.07.2023).
6. Официальное руководство по началу работы с Flask – URL: <https://flask-docs.readthedocs.io/en/latest/quickstart/> (дата обращения: 21.07.2023).

## Приложение

Программный код разработанного веб-приложения:

web.py:

```
from flask import Flask
from flask import render_template
from flask import request
```

```
from mycoder import incode
```

```
app = Flask(__name__)
```

```
@app.route('/')
def index():
```

```
    return render_template('index.html')
```

```
@app.route('/coder')
def coder():
```

```
    try:
```

```
        ans = incode(request.args['street'], request.args['number'])
```

```
    except Exception as e:
```

```
        return e.args[0], 400
```

```
    if 'pretty' in request.args and request.args['pretty'] == 'true':
```

```
        return render_template('ans.html', lat=ans[0], lon=ans[1], address=ans[2])
```

```
    else:
```

```
        return str(ans[0]) + ' ' + str(ans[1])
```

mycoder.py:

```
import requests
```

```
import json
```

```
def incode(street, number):
```

```
    q = "Санкт-Петербург, " + street + ", " + str(number)
```

```
resp = requests.get('http://51.250.101.212/search?q=' + q + '&format=json&limit=1')

j = json.loads(resp.content)
try:
    loc = j[0]
except IndexError:
    raise Exception('Cannot find given address')

return [loc['lat'], loc['lon'], loc['display_name']]
```