МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ образовательное учреждениевысшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра вычислительной техники

**ОТЧЁТ по ПРАКТИЧЕСКОЙ работе №3**

**по дисциплине «ПРОГРАММИРОВАНИНЕ»**

**«Разработка сервиса сокращения ссылок»**

|  |  |
| --- | --- |
| Факультет: АВТ  Группа: АБ-220  Студент(ы): Лазарев Кирилл | Преподаватель: Исаев Г.А. |

Новосибирск, 2023 г.

**Задание к работе:**

Требуется реализовать сервис для сокращения ссылок. Сервис

должен принимать запросы по http протоколу.

Сервис должен иметь две конечные точки:

1. Сокращение ссылки

URL: /

Метод: POST

Запрос содержит ссылку которую нужно сократить.

Ответ содержит сокращенную ссылку.

2. Переход по сокращённой ссылке

URL: /<сокращение>

Метод: GET

Ответ должен перенаправлять пользователя на исходную

страницу (см. перенаправления)

Для хранения сокращенных ссылок используем СУБД из практики 2.

**Постановка задачи:**

Написать клиент-серверное приложение которое будет принимать POST запросы с ссылками на некоторые ресурсы, после чего отправлять запрос в базу данных на удаленном сервере где будет храниться сокращенная ссылка и полный адрес ресурса. После получения ответа от СУБД отправить сокращенную ссылку пользователю.

При переходе по сокращенной ссылке сервер отправляет запрос к СУБД и проверяет наличие ресурса по данной ссылке, если таковой нашелся то произвести редирект пользователя на заданный ресурс.

**Алгоритм решения задачи**

Серверное приложение создает сокет для общения с СУБД и привязывает его к адресу сервера (localhost или локальный адрес) и нужному порту. Создать веб-приложение на фреймворке Flask и настроить роуты для нужных запросов (GET и POST). При получении POST запроса с данными отправить по сокету обработанный запрос к СУБД и получить ответ. При получении GET запроса (если указан адрес короткой ссылки) отправить запрос к СУБД и получить полный адрес по сокращенной ссылке. Вернуть пользователю код редиректа (302) и полную ссылку. Сервер не должен падать при некорректных значениях.

**Текст программы**

**main.py:**

**import** **os**

**import** **socket**

**from** **flask** **import** Flask, redirect, request

**class** **\_Socket**():

**def** **\_\_init\_\_**(self):

self.host = '127.0.0.1'

self.port = **6379**

**def** **send\_request**(self, message: str) -> str:

self.client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

self.client.connect((self.host, self.port))

self.client.send(message.encode(encoding='utf-8', errors='ignore'))

response = self.client.recv(**4096**).decode(encoding='utf-8', errors='ignore').split('**\n**')

self.client.close()

**return** response[**0**]

**class** **UrlShortener**:

**def** **\_\_init\_\_**(self, database\_filename: str = 'links'):

self.db\_filename = database\_filename

self.id = **0**

self.client = \_Socket()

**def** **shorten\_url**(self, full\_url):

short\_url = self.client.send\_request(f"--file {self.db\_filename} --query HGET furl {full\_url}")[**5**::]

**if** short\_url != '(null)' **and** 'link' **in** short\_url:

**return** short\_url

short\_url = f'link{self.id}'

self.id += **1**

self.save\_data(full\_url, short\_url)

**return** short\_url

**def** **get\_full\_url**(self, short\_url):

response = self.client.send\_request(f"--file {self.db\_filename} --query HGET links {short\_url}")

print(response)

**return** response[**5**::]

**def** **save\_data**(self, full\_url, short\_url):

self.client.send\_request(f"--file {self.db\_filename} --query HSET links {short\_url} {full\_url}")

self.client.send\_request(f"--file {self.db\_filename} --query HSET furl {full\_url} {short\_url}")

# def read\_data(self):

# response = self.client.send\_request(f"--file {self.db\_filename} --query HGET links {short\_url}")

# print(response)

app = Flask(\_\_name\_\_)

shortener = UrlShortener()

**@app**.route('/', methods=['POST'])

**def** **shorten\_url\_endpoint**():

full\_url = request.get\_data().decode(encoding='utf-8', errors='ignore')

**return** shortener.shorten\_url(full\_url)

**@app**.route("/", methods=["GET"])

**def** **index**():

html = '<html style="font-size: 200%; font-family: Verdana, Arial, sans-serif;">'

**return** f'{html}<h3>Сервис для сокращения ссылок<br>Для сокращения ссылки пользуйтесь POST запросом</h3>'

**@app**.route('/<short\_url>', methods=['GET'])

**def** **redirect\_endpoint**(short\_url):

full\_url = shortener.get\_full\_url(short\_url).replace('**\n**', '').strip()

print(f'{full\_url=}')

**if** full\_url == '(null)':

**return** redirect('/')

**if** len(full\_url.split('://')) == **1**:

**return** redirect(f'https://{full\_url}')

**return** redirect(full\_url)

**if** \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=**True**, host="127.0.0.1", port="5000")

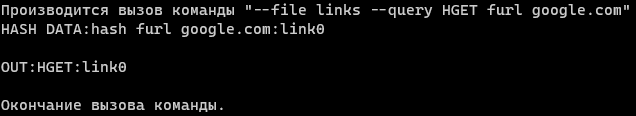
**Результат работы программы**



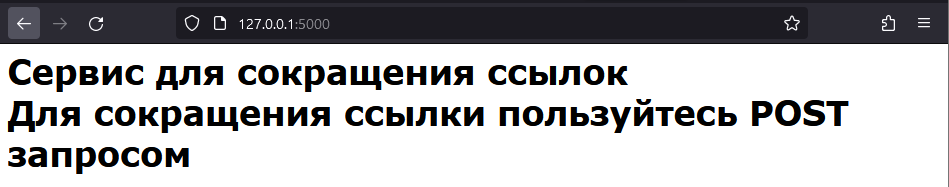
Отправляем POST запрос через curl



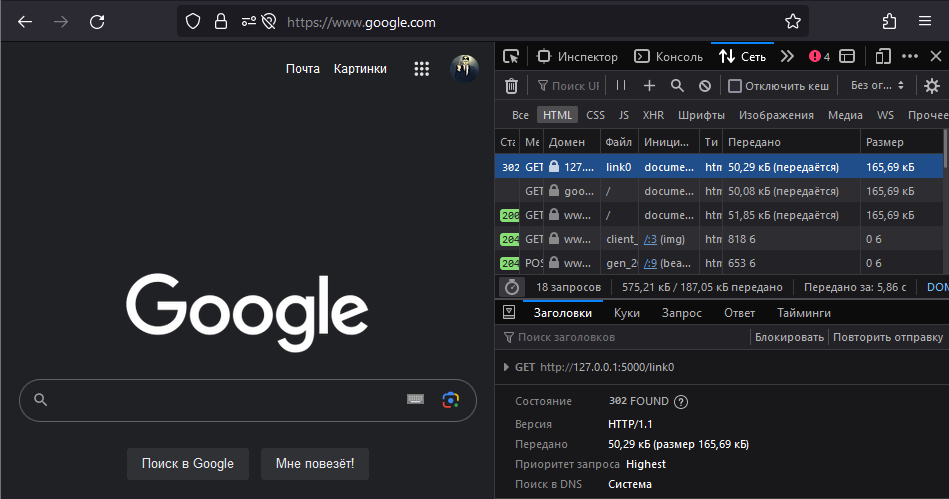
Сервер принял POST запрос



СУБД обработала запрос и вернула короткую ссылку



При переходе на главную страницу или по несуществующей ссылке мы попадаем на главную страницу



При переходе по ссылке /link0 мы попадаем на страницу google.com

*Программа СУБД взята в точности из первой практики и работает соответственно точно так же. Программа из второй практики была переделана (добавлен mutex)*

**Вывод**: Результат работы программы совпадает с ожидаемыми результатами, что говорит о корректной работе программы. Мы изучили как работать c GET и POST запросами, коды ответов от сервера, как происходит обработка запросов и научились применять эти знания на практике.