

Краткий сценарий тестирования
приложения по сбору
статистики посещаемости веб-сайта “WEBSTAT”

Исполнитель: Анастасия Гарбуз

anastasiagrbz@gmail.com

16.07.2020

1. Создание базы данных

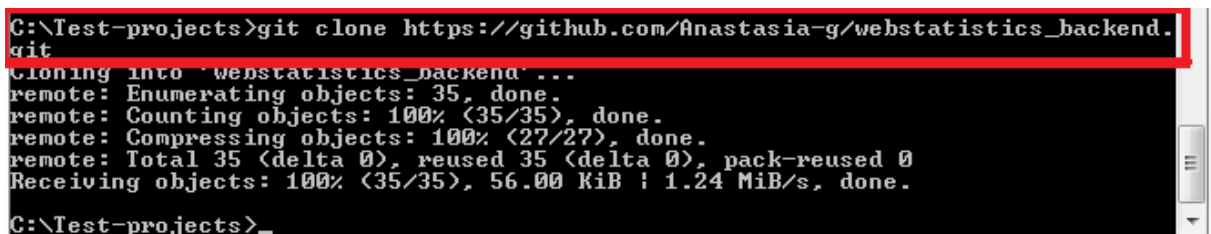
Необходимо создать базу данных PostgreSQL с названием “webstat”. К примеру, это можно сделать при помощи приложения “pgAdmin 4”, которое можно скачать по ссылке: <https://www.pgadmin.org/download/>

2. Установка приложения по сбору статистики посещаемости web-сайта.

В командной строке в нужную папку устанавливаем проект. Для этого вводим следующую команду:

```
git clone https://github.com/Anastasia-g/webstatistics\_backend.git
```

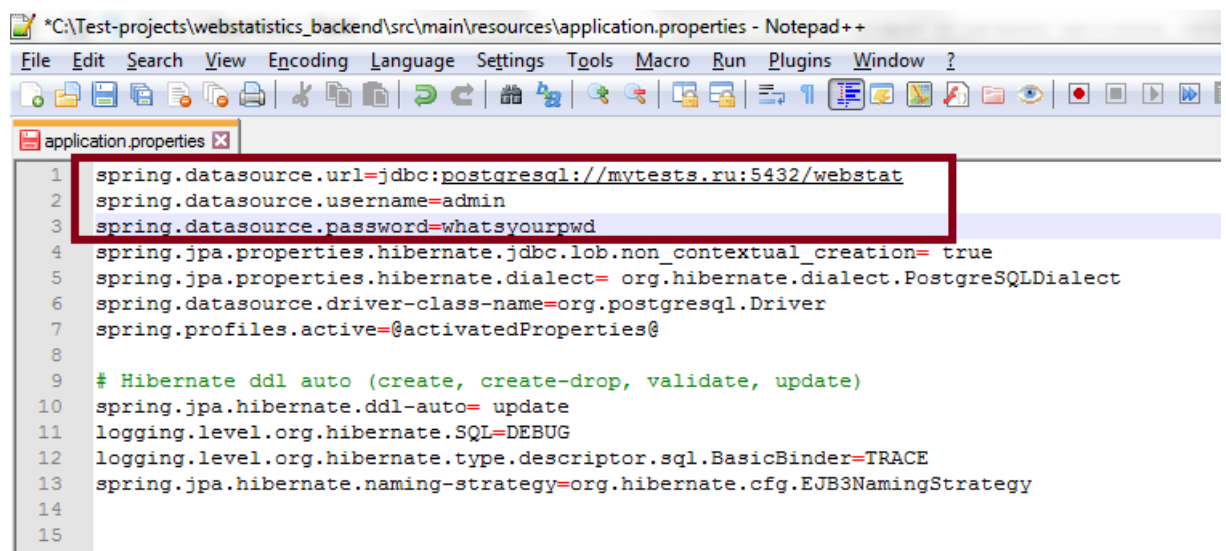
и нажимаем клавишу “Enter” (Рисунок 1):



```
C:\Test-projects>git clone https://github.com/Anastasia-g/webstatistics_backend.git
Cloning into 'webstatistics_backend'...
remote: Enumerating objects: 35, done.
remote: Counting objects: 100% (35/35), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 35 (delta 0), reused 35 (delta 0), pack-reused 0
Receiving objects: 100% (35/35), 56.00 KiB | 1.24 MiB/s, done.
C:\Test-projects>
```

Рисунок 1 – Установка проекта

В каталоге проекта по ссылке: `\src\main\resources` необходимо изменить значения трёх параметров соединения с базой данных в файле `application.properties` на необходимые (Рисунок 2):



```
*C:\Test-projects\webstatistics_backend\src\main\resources\application.properties - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
application.properties
1 spring.datasource.url=jdbc:postgresql://mytests.ru:5432/webstat
2 spring.datasource.username=admin
3 spring.datasource.password=whatsyourpwd
4 spring.jpa.properties.hibernate.jdbc.lob.non_contextual_creation= true
5 spring.jpa.properties.hibernate.dialect= org.hibernate.dialect.PostgreSQLDialect
6 spring.datasource.driver-class-name=org.postgresql.Driver
7 spring.profiles.active=@activatedProperties@
8
9 # Hibernate ddl auto (create, create-drop, validate, update)
10 spring.jpa.hibernate.ddl-auto= update
11 logging.level.org.hibernate.SQL=DEBUG
12 logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE
13 spring.jpa.hibernate.naming-strategy=org.hibernate.cfg.EJB3NamingStrategy
14
15
16
```

Рисунок 2 – Параметры соединения с базой данных для изменения

Затем переходим в папку с установленным проектом, вводим команду **mvn install** и нажимаем клавишу “Enter”:

```
C:\Test-projects>cd webstatistics_backend
C:\Test-projects\webstatistics_backend>mvn install
```

Далее вводим команду **mvn spring-boot:run** и нажимаем клавишу “Enter”:

```
C:\Windows\system32\cmd.exe - mvn spring-boot:run
C:\Test-projects\webstatistics_backend>mvn spring-boot:run
```

3. Установка приложения Postman для тестирования.

Необходимо перейти по ссылке: <https://www.postman.com/downloads/>

4. Тестирование заданий

Протестировать в приложении Postman задание №1 “Создание события посещения сайта пользователем”. Для этого необходимо указать следующие параметры и нажать кнопку “Send” (рисунок 3):

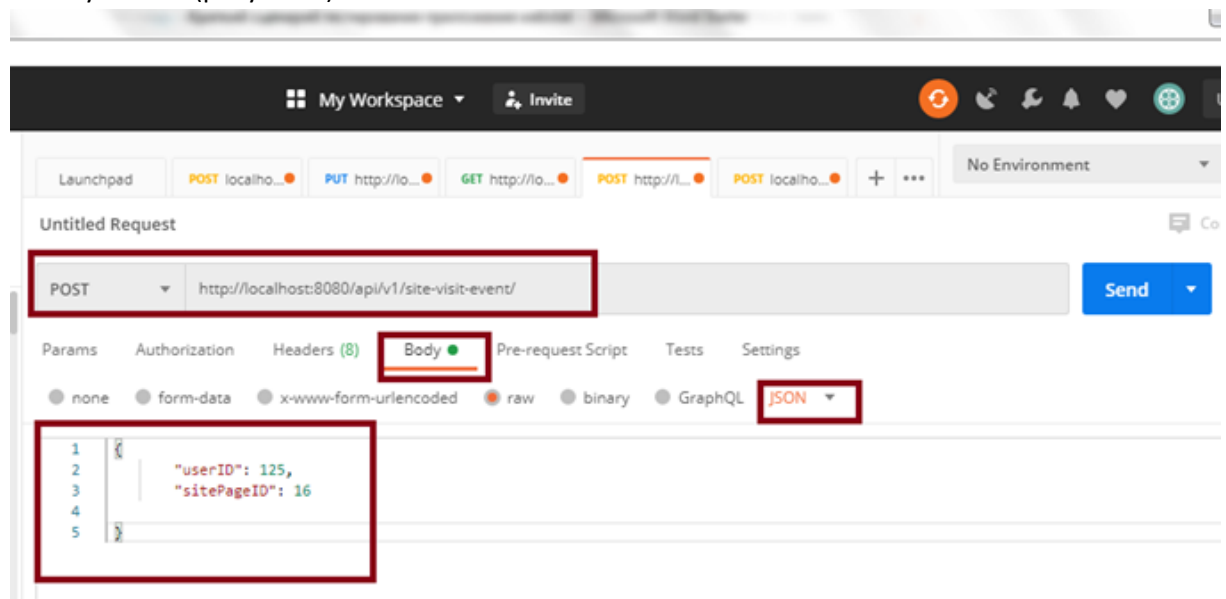


Рисунок 3 – Параметры для выполнения задания №1

Результат выполнения веб-сервиса можно увидеть в нижнем окне приложения Postman в формате JSON (рисунок 4), где:

- “visitsLast24Hrs” – это общее количество посещений сайта за текущие сутки;
- “uniqueVisitorsLast24Hrs” обозначает количество уникальных пользователей за текущие сутки.

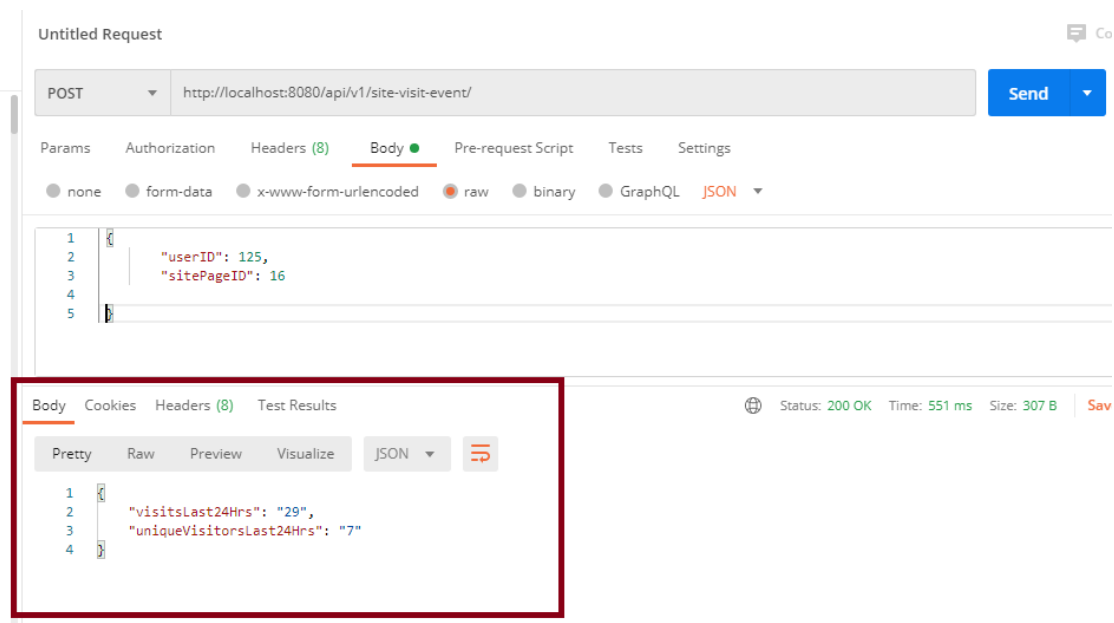


Рисунок 4 - Результат выполнения веб-сервиса по заданию №1

5. Протестировать в приложении Postman задание №2 “Получение статистики посещения за произвольный период”. Для этого необходимо указать следующие параметры и нажать кнопку “Send” (рисунок 5):

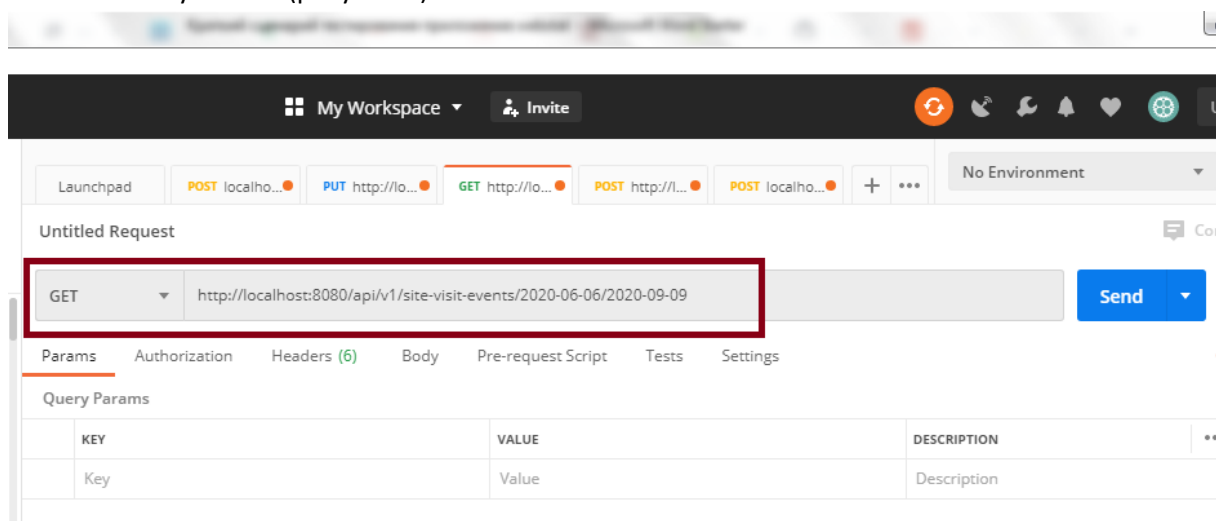


Рисунок 5 – Параметры для выполнения задания №2

Результат выполнения веб-сервиса можно увидеть в нижнем окне приложения Postman в формате JSON (рисунок 6), где:

- “visitsByPeriod” – это общее количество посещений сайта за период, заданный в строке запроса;
- “uniqueVisitorsByPeriod” обозначает количество уникальных пользователей за период, заданный в строке запроса;

- “regularVisitorsByPeriod” - количество постоянных пользователей за указанный период (пользователей, которые за период просмотрели не менее 10 различных страниц).

В данном примере данные представлены за период с 6 июня 2020 года по 9 сентября 2020.

The screenshot shows a REST client interface. At the top, a GET request is configured for the URL `http://localhost:8080/api/v1/site-visit-events/2020-06-06/2020-09-09`. Below the URL bar, there are tabs for Params, Authorization, Headers (6), Body, Pre-request Script, Tests, and Settings. The 'Query Params' section is currently active, showing a table with columns KEY, VALUE, and DESCRIPTION. Below this, the 'Body' tab is selected, displaying the response in JSON format. The JSON response is as follows:

```
{
  "visitsByPeriod": "29",
  "uniqueVisitorsByPeriod": "7",
  "regularVisitorsByPeriod": "1"
}
```

On the right side of the interface, the status is 200 OK, the time taken is 463 ms, and the size of the response is 335 B.

Рисунок 6 - Результат выполнения веб-сервиса по заданию №2