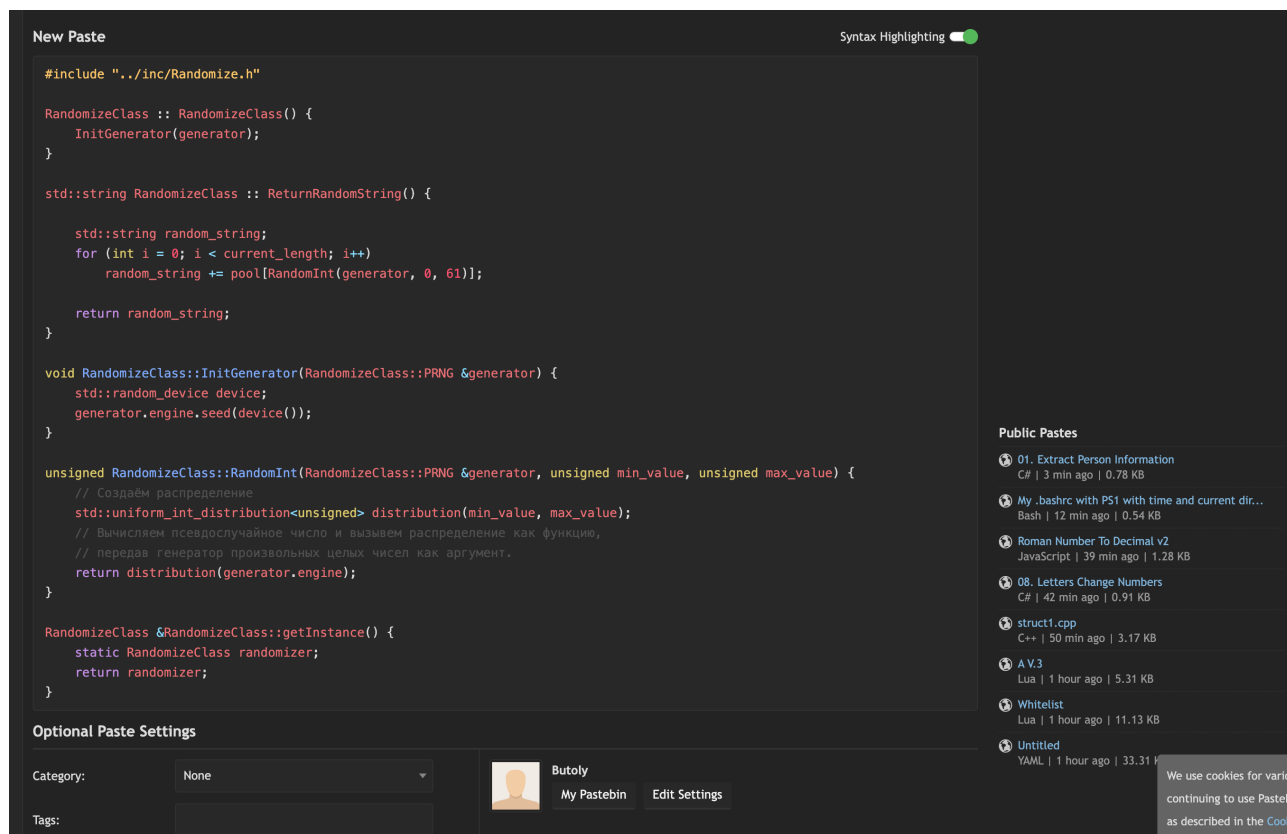


Введение

В рамках проекта курсовой работы рассматривается сервис PasteSlash реализованный в рамках обучения в НОЦ «Технопарк» при МГТУ им.Н.Э.Баумана. Сервис PasteSlash является аналогом сервиса PasteBin - <https://pastebin.com/>

PasteBin позволяет разработчикам делиться исходным кодом путём размещения кода в веб-сервисе. Веб-сервис генерирует URL, по которому разработчики могут увидеть исходный фрагмент кода.



При создании «Пасты» (так называется исходный код, которым разработчик хочется поделится с другими) веб-сервис генерирует уникальный URL, который разработчик может передать другим. Пример - <https://pastebin.com/MyEaZdEN>

О проекте

В данной курсовой работе предлагается рассмотреть серверную составляющую проекта PasteSlash.

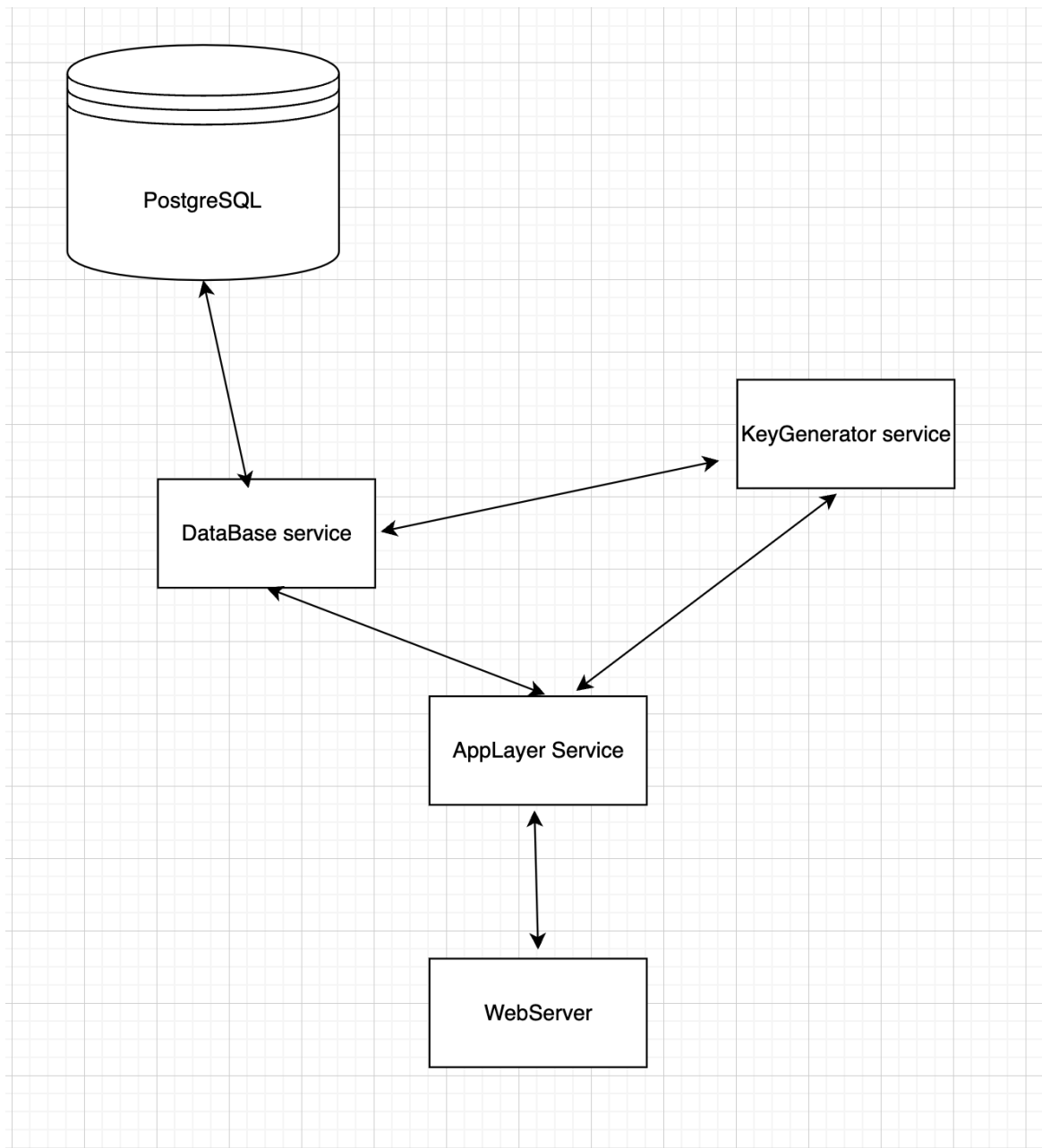
На рисунке представлена схема реализации микросервисной архитектуры.

WebServer - отвечает за http взаимодействие

DataBase service - отвечает за работу с Базой Данных

KeyGenerator - отвечает за генерацию уникальных ссылок, которые мапятся с пастами пользователей

AppLayer service - отвечает за бизнес логику сервиса. Например, авторизация пользователя, создание паст и т.д.



KeyGenerator service

Для написания тестов был выбран сервис KeyGenerator

Функционал микросервиса:

- Открытие вебсокетов для общения с AppLayer service и DataBase service
- Генерация уникальных урлов для паст методом Вихря Мерсенна
- Многопоточная обработка очередей генерации ключей

Микросервис реализован на языке C++. Для работы с сетью использовалась библиотека Boost.asio. Модульное тестирование реализовано посредством фреймворка GTests разработанный компанией Google. Для сборки проекта CMAKE/MAKE.

Структура проекта:

/bin - папка в которую сохраняется скомпилированная программа

/CMakeFiles - папка в которой хранятся файлы генерированные утилитой CMAKE для сборки проекта

/inc - заголовочные файлы проекта

/src - реализация методов классов

/tests - Модульное тестирование

Сборка проекта:

В корне проекта keygenerator запускаем следующие утилиты

CMake

```
[butol@192 in ~/Documents/PasteSlash/keygenerator (url-gen) $ cmake .  
-- Configuring done  
-- Generating done  
-- Build files have been written to: /Users/butol/Documents/PasteSlash/keygenerator
```

Make

```
[butol@192 in ~/Documents/PasteSlash/keygenerator (url-gen) $ make  
[100%] Built target url_gen
```

Запуска бинарного файла из папки /bin

```
[butol@192 in ~/Documents/PasteSlash/keygenerator/bin (url-gen) $ ./url_gen
```

Сервис поднимается на локалхосте с портом 3002

Откроем вебсокет с помощью утилиты Curl

```
[butol@192 in ~/Documents/PasteSlash/keygenerator/bin (url-gen) $ curl 127.0.0.1:3002  
7iGCPI3Xbutol@192 in ~/Documents/PasteSlash/keygenerator/bin (url-gen) $
```

Сервис вернет нам уникальный ключ для будущей пасты которая будет создана в AppLayerService.

На стороне сервиса видим соответствующие логи:

```
butol@192 in ~/Documents/PasteSlash/keygenerator/bin (url-gen) $ ./url_gen  
New connection  
GET  
Key: 7iGCPI3X  
Connection close
```

- Сервер открыл новое соединение по запросу
- Залакировал сгенерированный ключ
- Закрыв соединение

Тестирование

В рамках проекта были реализованы два вида тестирования:

- UnitTest
- EndpointTests

Компиляция и запуск модульного тестирования:

Из папки /tests запускаем утилиту cmake

```
[butol@192 in ~/Documents/PasteSlash/keygenerator/tests (url-gen) $ cmake .
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/butol/Documents/PasteSlash/keygenerator/tests
```

Запускаем утилиту make

```
[butol@192 in ~/Documents/PasteSlash/keygenerator/tests (url-gen) $ make
Scanning dependencies of target tests_run
[ 14%] Building CXX object CMakeFiles/tests_run.dir/tests.cpp.o
[ 28%] Linking CXX executable tests_run
[100%] Built target tests_run
butol@192 in ~/Documents/PasteSlash/keygenerator/tests (url-gen) $
```

Запускаем скомпилированный файл с модульными тестами:

```
[butol@192 in ~/Documents/PasteSlash/keygenerator/tests (url-gen) $ ./tests_run
Running main() from /Users/butol/Documents/googletest/googletest/src/gtest_main.cc
[=====] Running 6 tests from 3 test suites.
[-----] Global test environment set-up.
[-----] 1 test from GeneratorKeyTest
[ RUN      ] GeneratorKeyTest.NotEmptyReturnKey
[       OK ] GeneratorKeyTest.NotEmptyReturnKey (0 ms)
[-----] 1 test from GeneratorKeyTest (0 ms total)

[-----] 3 tests from ValidatorKeyTest
[ RUN      ] ValidatorKeyTest.ValidKey
[       OK ] ValidatorKeyTest.ValidKey (0 ms)
[ RUN      ] ValidatorKeyTest.CheckNonNullSingleton
[       OK ] ValidatorKeyTest.CheckNonNullSingleton (0 ms)
[ RUN      ] ValidatorKeyTest.CheckSingleton
[       OK ] ValidatorKeyTest.CheckSingleton (0 ms)
[-----] 3 tests from ValidatorKeyTest (0 ms total)

[-----] 2 tests from RandomizerTests
[ RUN      ] RandomizerTests.CheckSingleton
[       OK ] RandomizerTests.CheckSingleton (0 ms)
[ RUN      ] RandomizerTests.CheckGetRandomString
[       OK ] RandomizerTests.CheckGetRandomString (0 ms)
[-----] 2 tests from RandomizerTests (0 ms total)

[-----] Global test environment tear-down
[=====] 6 tests from 3 test suites ran. (0 ms total)
[ PASSED  ] 6 tests.
```

Для реализации Эндплинтестов был реализован bash-скрипт - ./endpoint_test.sh

```
[butol@192 in ~/Documents/PasteSlash/keygenerator/tests (url-gen) $ ./endpoint_test.sh
% Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
             %                   Dload  Upload    Total   Spent    Left   Speed
100      8    0       8    0    0   8000      0 --:--:-- --:--:-- --:--:--   8000
10T0GkfI
$key is NOT empty
[butol@192 in ~/Documents/PasteSlash/keygenerator/tests (url-gen) $ ]
```

Тест проверяет смог ли сервис вернуть ключ и выдает результат теста