Programming Assignment: Статистика web-сервера

✔ Passed · 1/1 points

Deadline The assignment was due on May 12, 11:59 PM PDT
You can still pass this assignment before the course ends.

Instructions My submission Discussions

Условие

Представим, что у нас есть web-сервер, который обслуживает запросы к интернет-магазину. Он поддерживает следующий набор запросов по протоколу HTTP:

- GET / HTTP/1.1 — получить главную страницу магазина
- POST /order HTTP/1.1 — разместить новый заказ
- POST /product HTTP/1.1 — добавить новый товар в магазин (команда админки)
- GET /order HTTP/1.1 — получить детали заказа
- PUT /product HTTP/1.1 — то же самое, что и POST /order HTTP/1.1
- GET /basket HTTP/1.1 — получить состав текущей корзины клиента
- DELETE /product HTTP/1.1 — удалить товар из интернет-магазина (команда админки)
- GET /help HTTP/1.1 — получить страницу о том, как пользоваться интернет-магазином

С точки зрения протокола HTTP, первые части приведённых выше запросов («GET», «POST», «PUT», «DELETE») называются *методами*. Вторые части называются *URI (Uniform Resource Identifier)*. Третья часть — это версия протокола. Таким образом, наш web-сервер поддерживает 4 метода: GET, POST, PUT, DELETE и 5 URI: «/», «/order», «/product», «/basket», «/help».

Главный системный администратор нашего сервера озаботился его масштабированием и для начала он решил изучить статистику использования. Он хочет для каждого метода и каждого URI посчитать, сколько раз он встречался в запросах к серверу за последний месяц. Он попросил вас помочь с этим.

У вас уже есть какая-то кодовая база для изучения запросов к серверу, и вы хотите воспользоваться ею, чтобы сэкономить время. У вас есть заголовочный файл http_request.h, содержащий структуру HttpRequest:

```
#pragma once

#include <string_view>

using namespace std;

struct HttpRequest {
  string_view method, uri, protocol;
};
```

Кроме того, есть заголовочный файл stats.h, содержащий объявления класса Stats и функции ParseRequest:

```
#pragma once

#include "http_request.h"

#include <string_view>
#include <map>

using namespace std;

class Stats {
public:
  void AddMethod(string_view method);
  void AddUri(string_view uri);
  const map<string_view, int>& GetMethodStats() const;
  const map<string_view, int>& GetUriStats() const;
};

HttpRequest ParseRequest(string_view line);
```

Наконец, у вас есть готовая функция ServeRequests:

```
Stats ServeRequests(istream& input) {
  Stats result;
  for (string line; getline(input, line); ) {
    const HttpRequest req = ParseRequest(line);
    result.AddUri(req.uri);
    result.AddMethod(req.method);
  }
  return result;
}
```

Вам нужно, основываясь на реализации функции ServeRequests, реализовать класс Stats и функцию ParseRequest. Дополнительные требования к классу Stats:

- метод GetMethodStats возвращает словарь, в котором для каждого метода хранится, сколько раз он встретился в качестве аргумента метода AddMethod;
- метод GetUriStats работает аналогично для URI;
- если метод, переданный в метод AddMethod, не поддерживается нашим сервером (список поддерживаемых методов приведён выше), то нужно на единицу увеличить счётчик для метода «UNKNOWN» (подробнее см. юнит-тесты в заготовке решения);
- если URI, переданный в метод AddUri, не поддерживается нашим сервером, то нужно на единицу увеличить счётчик для URI «unknown».

На проверку пришлите архив, состоящий из файлов stats.h и stats.cpp (а также любых других файлов, которые вы посчитаете нужным добавить в свой проект). При этом ваши файлы не должны содержать реализацию функции ServeRequests (если ваша посылка будет содержать функцию ServeRequests, вы получите ошибку компиляции).

Заготовка решения

http_request.h

stats.h

server_stats.cpp

Как будет тестироваться ваша посылка

К проекту из вашего архива будет добавлен cpp-файл, который:

- подключает заголовочный файл stats.h;
- содержит точно такую же реализацию функции ServeRequests, какая приведена в условии;
- содержит функцию main с набором юнит-тестов для функции ServeRequests

Ваш проект будет собран и запущен.

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab