

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«Київський політехнічний інститут імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп'ютерних систем

ТЕХНІЧНЕ ЗАВДАННЯ НА КУРСОВУ РОБОТУ
з дисципліни "Основи програмування"
на тему:
“Розробка мікрофреймворка для web-додатків”

Виконав: Кравчук Аркадій Андрійович
Група: КП-71

Затверджено

2 семестр 2017/2018

Вступ

В час інноваційних технологій створюються безліч готових продуктів, що здатні спростити та автоматизувати певні процеси або їх комплекс задля підвищення ефективності виконуваної роботи. Причому дуже зручно, коли цей продукт (програму) можна використовувати з різних пристроїв, а також коли кожен екземпляр завжди матиме актуальну інформацію. Вищеописану концепцію можна реалізувати за допомогою web-додатків, які зазвичай є кросплатформеними, адже їх функціонал доступний у web-браузері, а основна логіка зосереджується на сервері. Розробивши web-мікрофреймворк, можна швидко та без особливих зусиль створювати такі програми із наявних модулів, шаблонів, каркасів, розширень (попередньо налаштувавши їх), що описані та реалізовані у самому фреймворці. В результаті матимемо готовий продукт, який буде розміщений на web-сервері та здатний: генерувати сторінки web-сайту, зберігати, опрацьовувати і читати інформацію з БД, оброблювати відповідні запити (з декстопної програми) і надсилати відповіді.

Web-сервер - це сервер (комп'ютер із відповідним ПЗ), який підключений до мережі Інтернет та оброблює HTTP-запити (HTTP - основний мережевий протокол) від клієнтів, зазвичай веб-браузерів, видає їм HTTP-відповіді, разом з HTML-сторінкою, зображенням, файлом, медіа-потокм або іншими даними, які були розміщені на сервері або динамічно згенеровані встановленим ПЗ.

Фреймворк - готовий до використання комплекс програмних рішень, включаючи дизайн, логіку та базову функціональність системи або підсистеми, що полегшує створення та поєднання різних компонентів великого програмного забезпечення, дає можливість швидкого створення готового і об'ємного програмного продукту на основі каркасу зі змінними модулями чи розширеннями.

Web-фреймворк - програмний каркас, призначений для створення динамічних веб-додатків, служб або ресурсів на основі налаштованих модулів та інструментів в фреймворці. Він спрощує розробку, частково за рахунок автоматизації, і позбавляє від необхідності написання рутинного коду.

На відміну від більшості інших web-фреймворків, які зазвичай розроблені за допомогою інтерпретованих мов програмування, фреймворк на компільованій мові програмування C++ має значну перевагу у швидкодії та ефективності, здатен витримувати досить високі навантаження без залучення додаткових обчислювальних ресурсів.

Функціональні вимоги

1. Фреймворк дозволяє створювати екземпляри веб-додатків (Application) та запускати їх на певній IP-адресі та порті як веб-сервери.
2. Можна налаштувати Application за допомогою обробника запитів (Handler) загального призначення, який оброблятиме будь-які HTTP-запити до додатку. Також можна задати декілька однорангових обробників, тоді вони сформують ланцюжок обробників запиту.
3. Кожен Handler викликається із посиланням на додаток, у кому він зареєстрований (App), із контекстом запиту (Context) та посиланням на наступний Handler (Next) із ланцюжка налаштованих обробників.
4. Context у собі містить розібрані дані із HTTP-запиту (дані першого рядка, аргументи URL, словник заголовочних полів та тіло запиту) у вигляді екземплярів відповідних класів та методи для встановлення даних HTTP-відповіді (дані першого рядка, словник заголовочних полів та тіло відповіді).
5. Можна налаштувати обробники для визначених URL-шляхів (Route) та визначених HTTP-методів із внесенням параметрів у ці шляхи за допомогою простого формату, та із автоматичним парсингом всіх таких параметрів у контекст запиту у вигляді словника (Params).
6. Context дозволяє виконати перенаправлення (redirect) на іншу веб-сторінку.
7. Context містить словник розширення, що дозволяє обробникам додавати у нього власні дані та функції. Це дозволить створювати модулі-посередники (Middleware), що розширюватиме базовий функціонал Application.
8. Створити Middleware, що автоматично парсить JSON у тілі запиту, якщо встановлене відповідне значення у заголовочному полі Content-Type. Також цей модуль дозволяє додати у відповідь сервера функцію, що автоматично із JSON-об'єкта формує тіло відповіді у форматі JSON.

9. Створити Middleware для HTTP Cookie, що парсить із відповідного HTTP заголовку словник Cookie та дозволяє встановити нові Cookie у відповідь сервера.
10. Створити Middleware для запиту від форм, що містять тіло у форматі application/x-www-form-urlencoded. На основі даних такого тіла формується словник даних, що були внесені користувачем у формі веб-сторінки.
11. Створити Middleware, що дозволить формувати веб-сторінки відповіді сервера використовуючи простий logic-less шаблонізатор.
12. Необхідно створити демонстраційний графічний клієнт (Рис.1 ,2), що дозволить виконувати HTTP запити до веб-сервера та відображати відповіді.
13. Необхідно створити демонстраційний веб-сервер (Рис.3) для логістичної компанії, що використовує всі розроблені можливості і дозволяє обробляти клієнтські запити, підключатись до простої бази даних для отримання відповідних даних та генерувати клієнту за допомогою шаблонізатора веб-сторінки із результатами.
14. Розробити наступний функціонал для демонстраційного веб-серверу:
 - a. відстеження відправленого вантажу користувачем за номером квитанції. Для кожної квитанції зберігатиметься інформація про відправника, одержувача, дату прибуття та відправлення, поточне місцезнаходження та вартість доставки у БД. Ці дані будуть надіслані та відображені на веб-сторінці.
 - b. розрахунок приблизної вартості доставки (Рис.4) вантажу із зазначеними даними: габаритами, вагою, містом відправника та одержувача. На сервер буде здійснено відповідний запит, внаслідок чого користувач отримає бажану інформацію.
 - c. розрахунок приблизного терміну доставки на основі зазначеної дати відправлення, міста відправника та одержувача.
 - d. отримання інформації (графіку роботи, адреси) про відділення логістичної компанії у зазначеній(-ому) області(місті).
 - e. можливість виклику кур'єра з сайту, заповнивши форму (вказавши ПІБ, адресу), для відправлення/отримання вантажу.

- f. Відображення загальної інформації про компанію, новини у відповідних розділах, а також можливість відправити відгук (т.з. feedback)

Ескізи графічного інтерфейсу користувача

The interface shows a REST client with the following components:

- Method:** GET
- URL:** `http://training-american-ws.cloudhub.io/api/flights...`
- Query Parameters:** A table with one entry:

Key	Value
destination	CLE
- Tabs:** Authorization, Headers, Body, Pre-request Script, Tests. The 'Authorization' tab is selected.
- Auth Type:** No Auth
- Body Tab:** Contains sub-tabs: Body, Cookies, Headers (4), Tests. The 'Body' sub-tab is selected.
- Body View:** Pretty, Raw, Preview. The 'Text' view is selected.
- Body Content:** `1 No listener for endpoint: /`

Рис 1. Графічний клієнт для тестування роботи сервера

The interface shows a REST client with the following components:

- Method:** DELETE
- URL:** `https://postman-echo.com/delete`
- Tabs:** Authorization, Headers (1), Body, Pre-request Script, Tests. The 'Body' tab is selected.
- Body Type:** form-data, x-www-form-urlencoded, raw, binary. The 'x-www-form-urlencoded' type is selected.
- Body View:** Pretty, Raw, Preview. The 'JSON' view is selected.
- Body Content:**

```
1 {
2   "args": {},
3   "data": "Donec fermentum, nisi sed cursus eleifend, nulla tortor ultricies tellus,
4     condimentum vestibulum sapien. Etiam dignissim diam quis eros lobortis gravida
5   "files": {},
6   "form": {},
7   "headers": {
8     "host": "postman-echo.com",
9     "content-length": "256",
10    "accept": "**/*",
11    "accept-encoding": "gzip, deflate",
12    "cache-control": "no-cache",
13    "content-type": "text/plain"
```

Рис 2. Графічний клієнт для тестування роботи сервера

[Про компанію](#) [Новини](#) [Правова інформація](#) [Feedback](#)

Відстежити вантаж

Вартість доставки

Терміни доставки

Найближче відділення

Графік роботи відділень

Виклик кур'єра

Banner

Рис 3. Демонстраційний веб-сервер (початкова веб-сторінка)

Вартість доставки

Даний сервіс допоможе Вам дізнатися орієнтовну вартість доставки Вашого відправлення

Маршрут

Місто-відправник

Місто-одержувач

Вид відправлення

Вантажі

Фактична вага

кг

Габарити відправлення

Довжина

Ширина

Висота

см

Розрахувати вартість

[Очистити](#)

Рис 4. Демонстраційний веб-сервер (розрахунок приблизної вартості доставки)