

## WEB-модуль прискореного обміну інформацією з біржовою системою

Вінницький національний технічний університет

### **Анотація**

*В роботі проведено аналіз існуючих WEB-модулів для прискорення обміну інформацією із сервером. Проведено аналіз особливостей існуючих методів та технологій скорочення тривалості обробки інформації, та приведено порівняльну характеристику. Розглянуті питання, що пов'язані з попередньою обробкою інформації на стороні клієнта, завантаженням скриптів. Запропоновано варіант реалізації WEB-модулю, який зменшує час обміну інформацією з біржовою системою у порівнянні із аналогами.*

**Ключові слова:** WEB-модуль, скрипт, клієнт, обмін інформацією, біржова система, трейдинг.

### **Abstract**

*It was done the aliases of existing WEB-modules for acceleration of data exchange between WEB-client and server. Also it was done the analysis of characteristics of existing methods and technologies for reducing the length of processing information. It was analyzed a lot of number of important issues related to the handling and scripts transferring. It was designed the WEB-module that reduces the volume of exchanging information with Trading system.*

**Keywords:** WEB-module, volume, information, processing, information, WEB-server, exchange system trading.

### **Вступ**

Однією з тенденцій інформаційних технологій є клієнт-серверна архітектура в “тонким” клієнтом. Її реалізація на WEB-технологіях дає можливість створення мультиплатформених програмних додатків з можливістю централізованого оновлення програмного забезпечення. Але класична схема їх роботи для динамічних структур вимагає обміну суттєвими об'ємами інформації між сервером і клієнтом. Варіантом зменшення мережевого трафіку є попередня обробка інформації і перенесення частини серверних функцій на клієнтську частину. Це дозволяє перерозподілити навантаження і збільшити реактивність системи..

### **Результати**

В цій розробці була поставлена задача створення програмного забезпечення, яке б перенесло частину обчислювальних функцій, а також перевірку даних на сторону клієнта. Основною концепцією даного підходу було використання серверу в якості RESTfull API. Даний підхід надає ряд переваг, основною з яких є те, що додаток в першу чергу відповідає бізнес завданням і відповідно до цього повинен обиратись функціонал. Проте, є певний ряд труднощів, які містять в собі даний підхід [1].

Одна з проблем є проблема індексації пошуковими системами. Лише декілька пошукових систем можуть проаналізувати сторінки, контент яких стає доступним після виконання JavaScript-додатків. Другою проблемою такого рішення є те, що контент не відворюється до повного завантаження WEB-додатку і запуску браузером його скриптів [2].

Одним з варіантів вирішення даної проблеми є зменшення кількості необхідних файлів для початку роботи додатку шляхом їхньої конкатенації та мініфікації, проте, в результаті не відбудеться значного приросту продуктивності додатку.

Інший спосіб полягає у асинхронному завантаженні компонентів додатку. Даний спосіб вимагає підтримки на рівні MVC-фреймворків.

Виходячи з постановлених задач, було розроблено WEB-модуль, який поєднує в собі вищенаведені методи. Під час запуску у ньому проводиться аналіз js-файлів та на основі дерева залежностей генерується оптимізований вихідний файл. Під час запиту необхідної сторінки

користувачем, відбувається виконання всіх js-скриптів на стороні сервера і згенерована WEB-сторінка надсилається користувачеві. На рис. 1, 2 наведені результати порівняльного дослідження швидкодії WEB-сторінки без і з розробленим модулем.

Range: 429 ms – 7.26 s

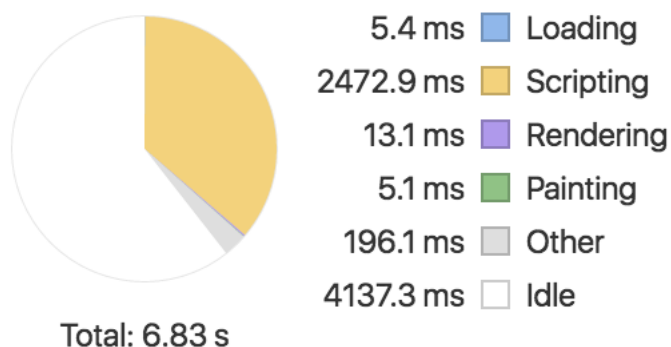


Рис. 1. Витрати часу на обробку запиту без використання розробленого модулю

Range: 716 ms – 6.98 s

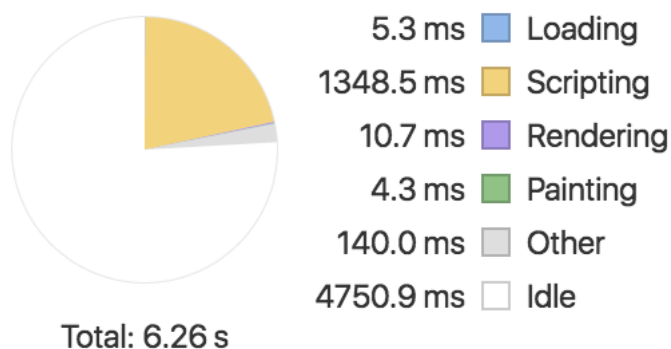


Рис. 2. Витрати часу на обробку запиту з використанням розробленого модулю

Як видно з рис. 1 та 2 час, необхідний WEB-переглядачеві на обробку js-скриптів зменшується при застосуванні модуля на 45% [3].

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Trustworthy Compilers, Vladimir O. Safonov, Chapter 8.6 «AOT Compilation»
2. Angular Universal [Електронний ресурс] /«Google» – Server side rendering–, 2016. – Режим доступу: <https://universal.angular.io>, вільний.
3. Aot компіляція [Електронний ресурс] /«Google» – *Ahead-of-Time Compilation*–, 2016. – Режим доступу: <https://angular.io/docs/ts/latest/cookbook/aot-compiler.html>, вільний.

**Ковальчук Віталій Валерійович** — студент групи 1AKIT-16сп, факультет комп'ютерних систем та автоматики, Вінницький національний технічний університет, Вінниця, e-mail: [v.kovalchuk.vitaliy@gmail.com](mailto:v.kovalchuk.vitaliy@gmail.com);

**Науковий керівник:** Паламарчук Євген Анатолійович — к.т.н., доцент, Вінницький національний технічний університет, м. Вінниця, email : [p@vntu.edu.ua](mailto:p@vntu.edu.ua).

**Kovalchuk Vitaliy V.** — Department Computer Systems and Automation, Vinnytsya National Technical University, Vinnytsya, email : [v.kovalchuk.vitaliy@gmail.com](mailto:v.kovalchuk.vitaliy@gmail.com);

**Supervisor:** Palamarchuk Yevhen A.— Ph.D., Vinnytsya National Technical University, Vinnytsya, email : [p@vntu.edu.ua](mailto:p@vntu.edu.ua).