

МIНIСТЕРСТВО  ОСВIТИ І НАУКИ  УКРАЇНИ

НАЦІОНАЛЬНИЙ   ТЕХНІЧНИЙ УНІВЕРСИТЕТ   УКРАЇНИ

“КИЇВСЬКИЙ  ПОЛІТЕХНІЧНИЙ  ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Practice 5**

з дисципліни “Якість та тестування програмного забезпечення ”

на тему «**Apache JMeter**»

|  |  |
| --- | --- |
| Виконав  студент II курсу  групи КП-92  Остапенко Іван Петрович  (*прізвище, ім’я, по батькові*) |  |

Київ 2021

**Завдання**

Script the next flow of following site:

1.Open homepage http://demo.nopcommerce.com/

2.Search for random product

3.Open random product page from search results

4.Add to cart

5.Go to shopping cart

Add specific assertions for each request (e.g. response codes, response body etc.)

Place results on public GitHub (or git.epam.com)

Send us the link with the results

**Особливості реалізації тест-кейсу**

1.Open homepage <http://demo.nopcommerce.com/>

За допомогою HTTP request “open site”

2.Search for random product

Особливості реалізації сайту: пошуковий рядок має складатися як мінімум з 3 символів. Якщо генерувати рандомний рядок, наприклад, за допомогою вбудовагої функції JMeter - RandomString, то в більшості випадків результат пошуку буде 0 продуктів. Тому реалізовано пошук з масиву "app", "nike", "htc","adi", "empty" (див. HTTP request “search ${SEARCH}”)

Після запиту виконується XPath Extractor “link random product”, що зберігає посилання на товар з результату пошуку випадково.

Після виконується перевірка, чи було знайдено товар з результатів пошуку.

3.Open random product page from search results

За допомогою HTTP request “go to ${PRODUCT\_LINK}”

Після запиту виконується XPath Extractor “product id”, що зберігає id товару.

Після запиту виконується XPath Extractor “default quantity”, що зберігає кількість товару за замовчуванням. (для деяких товарів – це не 1)

4.Add to cart

Особливості реалізації сайту: добавлення до кошику відповується за допомогою POST-запиту:

Запит: /addproducttocart/details/${PRODUCT\_ID}/1

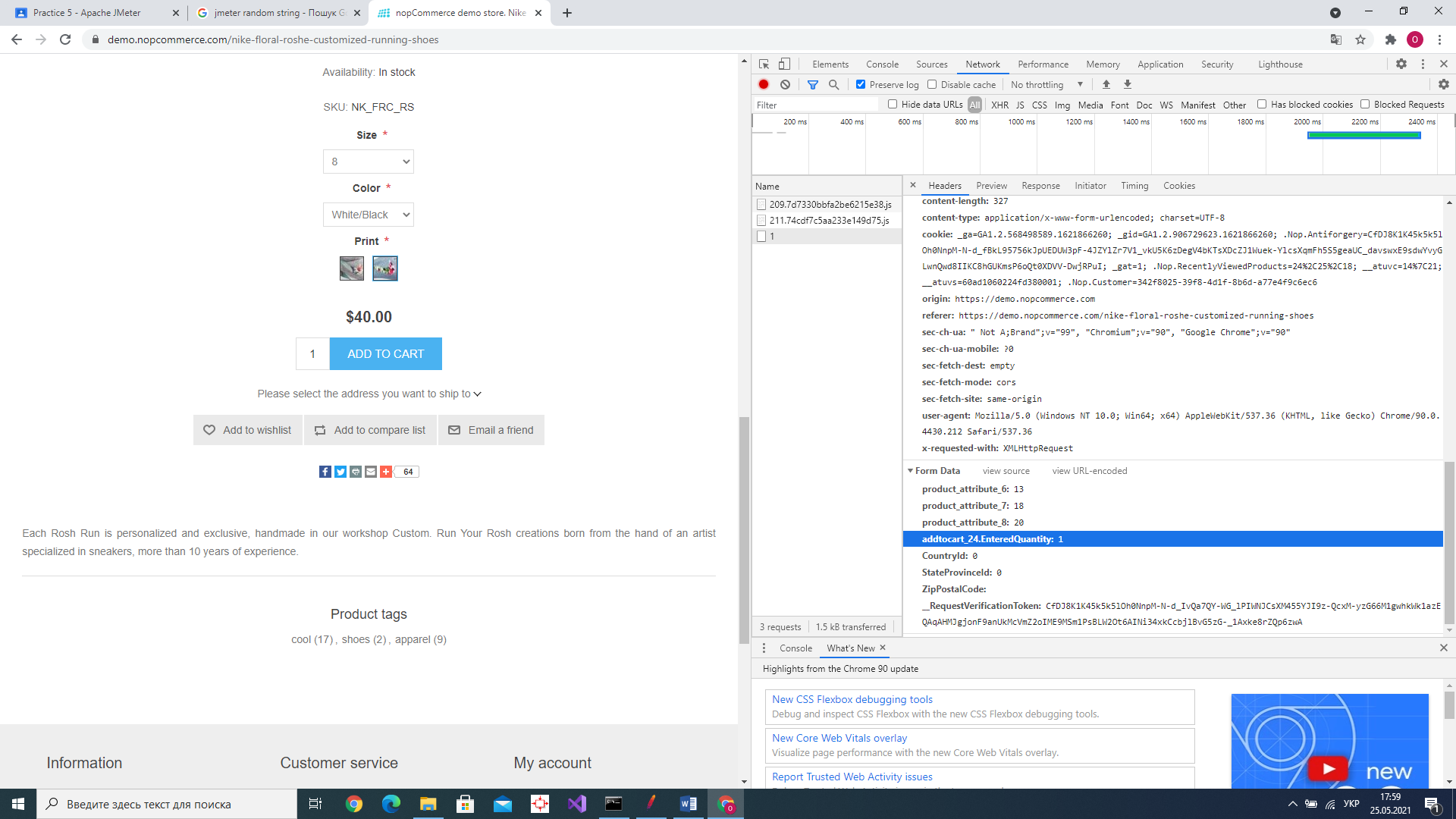
Тіло запиту: addtocart\_${PRODUCT\_ID}.EnteredQuantity – кількість товарів для замовлення

Особиста думка щодо реалізації сайту:

планувалося оброблювати запити у вигляді:

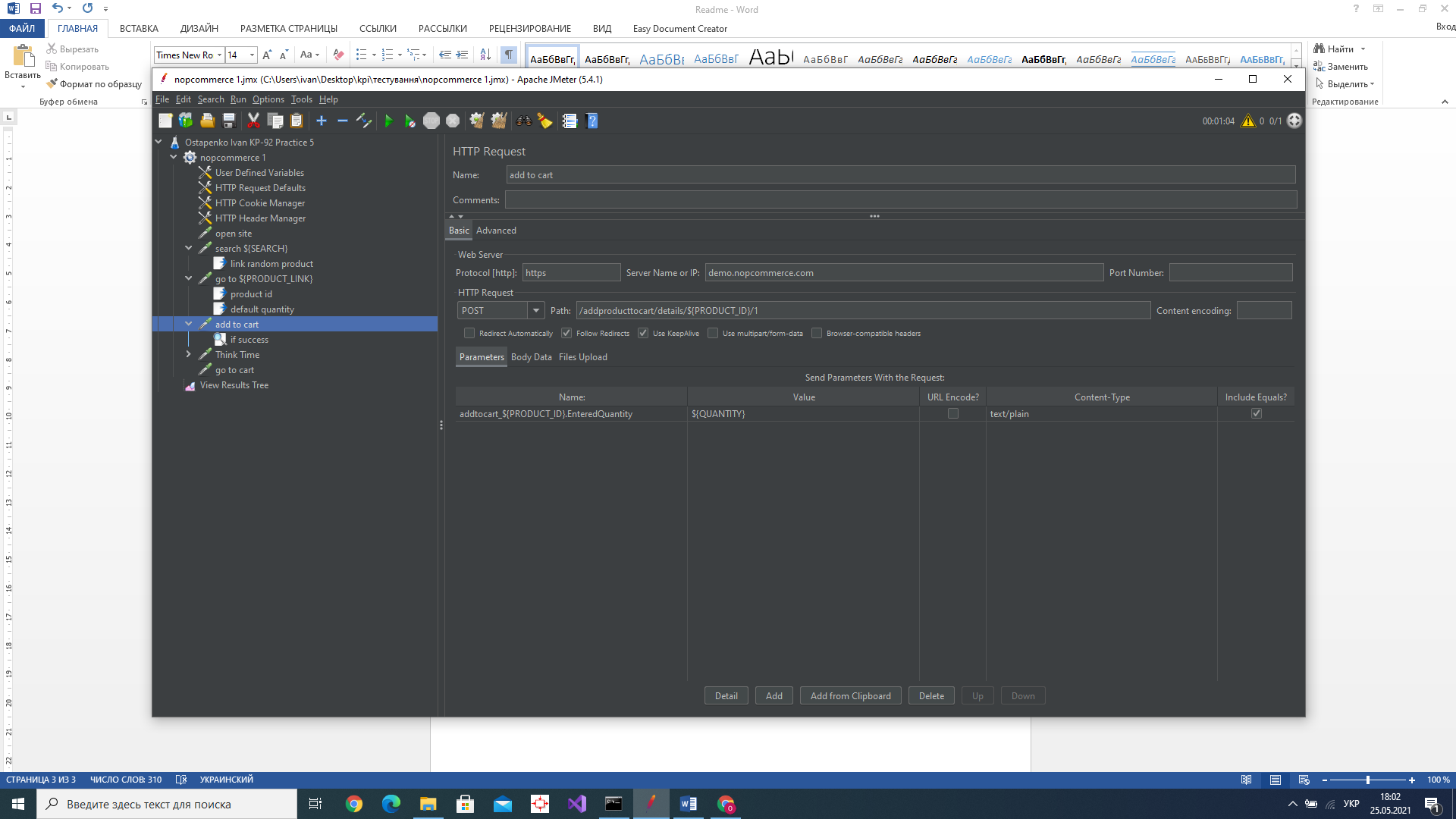
/addproducttocart/details/${PRODUCT\_ID}/${QUANTITY}

Однак під час добавлення до множини, наприклад, взуття, з’явилися додаткові параметри такі як розмір, колір, тощо. Тому було прийнято рішення передавати дані у тілі запиту. Чому передаються ці дані у такому вигляді



де product\_attribute\_6 – size, product\_attribute\_7 – color, product\_attribute\_8 – print, addtocart\_24.EnteredQuantity – кількість.

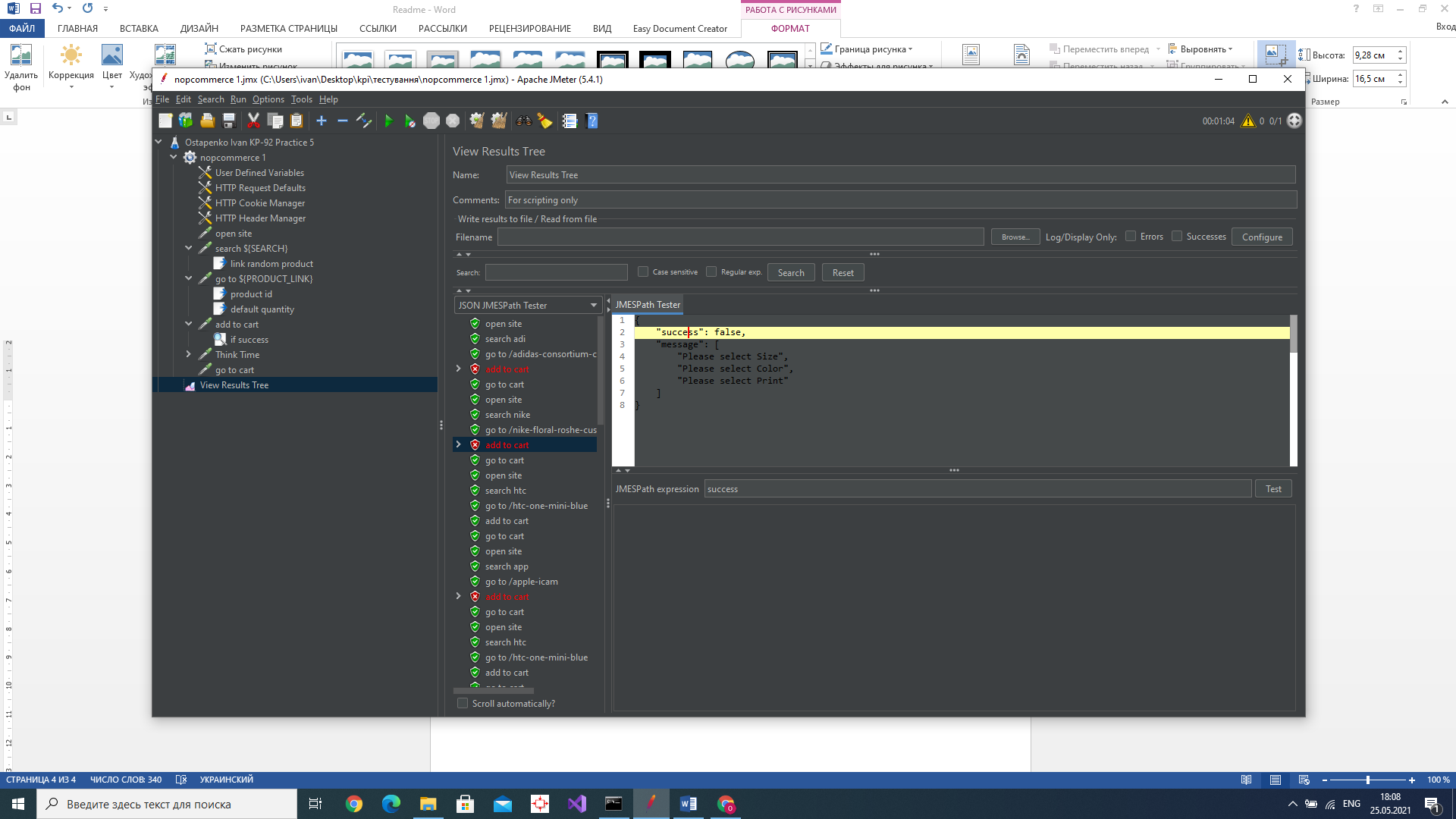
Така назва параметрів запиту ускладнює тестування та підтримку сайта (порушені правила іменування, наприклад, за книгою R. Martin Clean Code. Через що, запит на добавлення товару, також виглядає менш читабельно ніж могло бути.

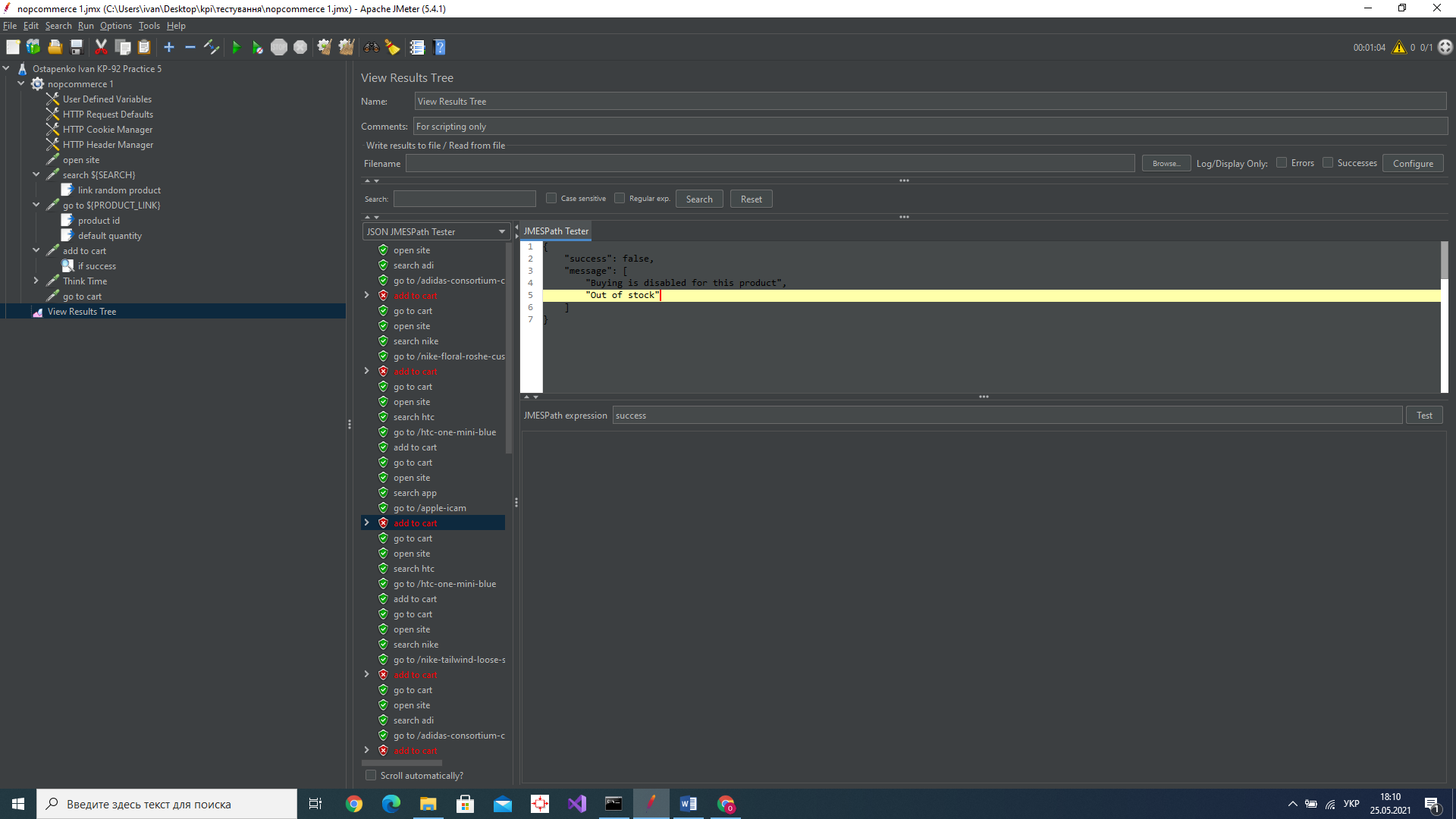


Якщо добавлення неможливе, то сервер відповідає:

Status: 200

Body:



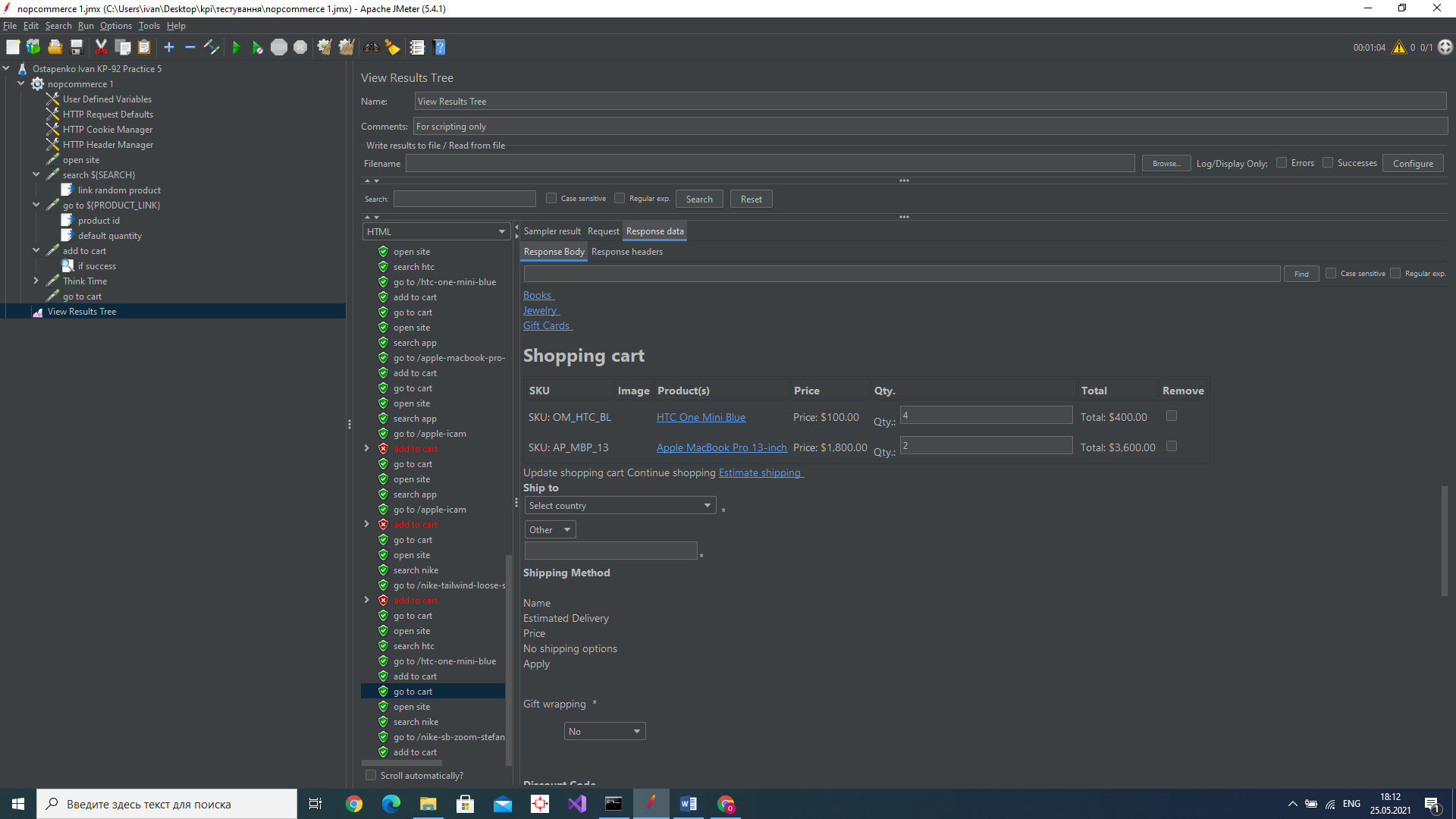


Тому після запиту за допомогою перевірки JSON Assertion перевіряється чи сапит пройшов успішно

5.Go to shopping cart

За допомогою HTTP request “go to cart”

Результат:



Для зручності демонстрації встановлено один потік для виконання та починання нового циклу після невдалої перевірки.

**Висновок**

Під час виконання лабораторної роботи отримано досвід роботи з JMeter (HTTP Request, HTTP Test Script Recorder), XPath.