 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №6**

з дисципліни «ВЕБ-технології та ВЕБ-дизайн»

# на тему: «Тестування веб додатків»

**Виконав:**

студент гр. БС-83

Погребенко В.О.

**Перевірили:**

ас. Давидько О.Б.

ас. Матвійчук О.В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2020

**Мета роботи:**

Ознайомитися із створенням тестів.

**Завдання:**

1. Написати тестовий сценарій використання програмного продукту (приклад <https://victorz.ru/202001101079>)

2. Написати 2-3 unit теста для додатку

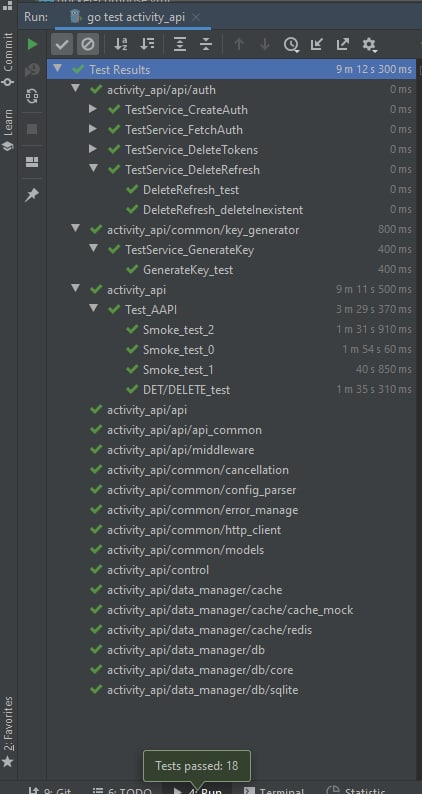
|  |  |
| --- | --- |
| **Заголовок** | Перевірка основної функціональності Activity Api. |
| **Передумова** | Працюючий сервіс з доступом до його апі |
| **Крок** | **Очікуємий результат** |
| Зареєструватись у сервісі надіслав довільне ім’я та хеш свого довільного паролю на хендлер /register | У відповіть повернено http повідомлення зі статусом успіху(201) |
| Зробити логін у сервіс надіслав ім’я та хеш паролю на хендлер /login | У відповіть повернено http повідомлення зі статусом успіху(200)та токен доступу |
| Додати довільну кількість коректних департаментів надіслав їх у форматі JSON на хендлер /departments | У відповіть на кожне додавання повернено http повідомлення з id доданого об’єкту у форматі JSON |
| Додати довільну кількість коректних користувачів до доданих департаментів надіслав їх у форматі JSON на відповідний хендлер | У відповіть на кожне додавання повернено http повідомлення з id доданого об’єкту у форматі JSON |
| Додати довільну кількість коректних активностей до доданих користувачів надіслав їх у форматі JSON на відповідний хендлер | У відповіть на кожне додавання повернено http повідомлення з id доданого об’єкту у форматі JSON |
| Перевірити існуючі об’єкти у БД | Таблиці з департаментами, користувачами та активностями повинні містити усі додані об’єкти з попередніх кроків |
| Зпросити департамент(и)\ юзера(ів)\ активність(ності)\ з відповідних хендлерів апі | У відповіть повернено http повідомлення з записами про департамент(и)\ юзера(ів)\ активність(ності), що збігаються з даними, що були завантажені до сервісу на попередніх кроках |
| Запросити дані про час активності довільного користувача від сервісу за відповідним хендлером, та порівняти його з очікуванними даними підрахованими вручну | У відповіть повернено http повідомлення з даними про активність користувачів (у форматі JSON), що збігаються з даними підрахованими «вручну» |
| Запросити дані про час активності довільного департаменту від сервісу сервісу за відповідним хендлером, та порівняти його з очікуванними даними підрахованими вручну | У відповіть повернено http повідомлення з даними про активність департаментів (у форматі JSON) що збігаються з даними підрахованими «вручну» |
| Видалити усі додані об’єкти за допомогою їх ID використовуючи відповідні хендлери Api | У відповіть на кожне видалення повернено http повідомлення зі статусом успіху та кількістью змінених об’єктів у форматі JSON. Кільвість змінених об’єктів повинна бути < 2 |
| Анреєструватися у сервісі | У відповіть на кожне видалення повернено http повідомлення зі статусом успіху, та виконався анлогін у сервісі |
| Перевірити існуючі об’єкти у БД | Таблиці БД з департаментами, юзерами та активностями повинні бути пустими |

Табл.1 - Тестовий сценарій використння програмного продукту

**Код за посиланням:**

<https://github.com/PogrebenkoBS81/Web_course3/tree/master/lab4-5/activity_api>

Результат роботи:



***Контрольні питання:***

1. ***Що таке End-to-end тестування***

***End-to-end тестування*** - це метод тестування програмного забезпечення, який перевіряє все програмне забезпечення від початку до кінця, а також його інтеграцію із зовнішніми інтерфейсами. Метою такого тестування є тестування всього програмного забезпечення на залежності, цілісність даних та зв'язок з іншими системами, інтерфейсами та базами даних для здійснення повного виробничого сценарію.

1. ***Який функціонал додатку покривається Smoke тестами***

***Smoke*** тести покривають лише найголовніший, основний функціонал, і перевіряють його на базові та найбільш серйозні помилки. Результат цього тестування використовується для того, щоб вирішити, чи достатньо стабільний білд для подальшого тестування.

1. ***Сценарії використання unit тестів***
2. ***Які існують рівні тестування***

***Unit Testing:*** перевіряє, чи компоненти програмного забезпечення відповідають функціональним можливостям чи ні. Мета - протестувати кожну частину програмного забезпечення, відокремивши її. Він перевіряє, чи компонент відповідає функціональним можливостям чи ні. Таке тестування проводять розробники. Цей вид тестування допомагає протестувати кожен модуль окремо.

***Integration Testing:*** перевіряє потік даних від одного модуля до інших модулів. На цьому етапі тестування різні програмні модулі поєднуються та тестуються як група, щоб переконатися, що інтегрована система готова до тестування системи.

***System Testing:*** оцінює як функціональні, так і нефункціональні потреби в тестуванні. Проводиться на повній, інтегрованій системі. Це дозволяє перевірити відповідність системи відповідно до вимог. Він перевіряє загальну взаємодію компонентів. Він передбачає тестування навантаження, продуктивності, надійності та безпеки.

***Acceptance Testing:*** перевіряє, чи відповідають вимогам специфікації або контракту відповідно до її поставки. Таке тестування в основному проводиться користувачем або замовником. Однак до цього процесу можуть бути залучені й інші акціонери.