

BetterMe

QA Hiring Event

В цьому документі ви можете ознайомитися з ключовими кроками майбутніх завдань, а також підготувати усі необхідні вам інструменти. Все необхідне краще налаштувати і підготувати заздалегідь, щоб не втрачати час під час івенту.

Результати виконаних завдань потрібно буде завантажити у власний public repository на GitHub, тому його краще створити перед участю в івенті. Як це зробити можна подивитися за посиланням: [Getting started with your GitHub account](#)

Опис завдань:

Завдання 1: Учасники отримають мобільний застосунок з функцією підписок (покупка підписок через Google Play). Метою буде перевірка коректності обробки всіх етапів життєвого циклу підписки: активація, автоматичне продовження, скасування, відновлення після перерви, завершення терміну дії.

Ключові кроки:

- Перевірка процесу покупки та поновлення підписки.
- Перевірка процесу скасування підписки та перевірка реакції системи.
- Перевірка сценаріїв закінчення терміну підписки та поведінки користувача після його завершення.
- Виявлення потенційних багів, пов'язаних із невірною обробкою транзакцій.

Інструменти:

- Тобі знадобиться: емулятори в [Android studio](#) або реальні android девайси, [GitHub акаунт](#).
- Як створити репозиторій і залити туди рішення: [GitHub Docs](#)

Завдання 2: Надати учасникам доступ до Swagger-документації для API сервісу, що обробляє запити від мобільного застосунку. Мета — розробити тестові кейси для основних REST API ендпоінтів, включаючи позитивні та негативні сценарії.

Ключові кроки:

- Аналіз Swagger документації для розуміння методів API (GET, POST, PUT, DELETE).
- Написання тест кейсів для кожного з методів, враховуючи різні статуси відповідей (200 OK, 404 Not Found, 500 Internal Server Error).

- Симуляція некоректних запитів (наприклад, з неправильною авторизацією або відсутніми полями).
- Створити невелику Postman колекцію, з ланцюжком 3-4 запита, логічно пов'язаних один з одним, використовуючи Swagger документацію.
- Рішення залити на github у вигляді експортованої колекції.

Інструменти:

- Тобі знадобиться: [Postman](#), [GitHub акаунт](#).
- 1. У запитах не повинно бути (або мінімум) статичних даних. Необхідно використовувати на вибір: вбудовані функції postman для генерації даних або Pre-Request Script
- [Write pre-request scripts to add dynamic behavior in Postman | Postman Learning Center](#)
- [Reuse data with variables and environments in Postman | Postman Learning Center](#)
- [Use dynamic variables to return randomly generated data | Postman Learning Center](#)

Pre-Request Script example:

```
"var pet_id = pm.variables.replaceIn('{{$randomInt}}');"
"pm.environment.set("PET_ID", parseInt(pet_id));"
```

1. До кожного запиту мають бути тести. Розробка тестів та їх кількість (мінімум 1 тест) повністю на розсуд учасника.
- [Postman test script examples | Postman Learning Center](#)
1. Колекцію можна запустити всю повністю за допомогою Run collection в Postman без редагування.
- [Test your API using the Collection Runner | Postman Learning Center](#)

Завдання 3: За допомогою інструментів проксування (наприклад, Charles Proxy або Fiddler) учасники мають перехопити HTTP/HTTPS-запити. Метою буде перевірити, які саме дані передаються між клієнтом і сервером і як реагує додаток на різні статуси коду.

Ключові кроки:

- Налаштування проксі для перехоплення запитів від застосунку.
- Аналіз запитів на відповідність специфікаціям API та коректність даних.
- Виявлення помилок або зайвих запитів, які можуть вплинути на продуктивність.

Інструменти:

- Тобі знадобиться: Postman, Charles Proxy, Fiddler або Proxyman, [GitHub акаунт](#).
- Налаштуйте debug proxy інструмент: [Charles Web Debugging Proxy](#)

Як налаштувати Charles:

1. Після установки треба підключити девайс до комп'ютера
2. Відкрийте налаштування wifi на девайсі
3. Натисніть властивості (advanced)
4. У HTTP PROXY розділі оберіть Manual
5. Введіть ip адресу вашого комп'ютера (через термінал Win - ipconfig)
6. Введіть проху порт (8888 - за замовчуванням)
7. Дозвольте підключеному пристрою переглядати трафік через ваш проксі (на вашому комп'ютері).
8. Також необхідно встановити сертифікати на пристрої для розшифрування різних запитів HTTPS-трафіку:
 - Налаштуйте пристрій для використання Charles як його HTTP-проксі на <IP-адрес вашої робочої станції>
 - На девайсі відкрити <http://chls.pro/ssl>
 - Установити Certificate

Завдання 4: Учасники мають проаналізувати мобільний додаток і виходячи з результатів сформулювати список тест кейсів, який би включав перевірки існуючого функціоналу з інтеграцією нової фічі.

Ключові кроки:

- Exploratory testing.
- Аналіз вимог задачі.
- Написання тест кейсів.

Інструменти:

- [Мобільні емулятори \(Android Studio\)](#), [GitHub акаунт](#).