

1. Підготовка до тестування

1.1 Завантаження та налаштування Fiddler

- Завантажено програму **Fiddler** з офіційного сайту Telerik.
- Виконано встановлення Fiddler на комп'ютері та запуск програми.
- У Fiddler був записаний HTTP-трафік під час виконання сценарію вручну в тестовій системі Mantis. Це включало:
 - Логін до системи.
 - Редагування профілю користувача (зміна email та імені).
- Збережено сесію в форматі **.saz** для подальшого використання в JMeter.

1.2 Запуск Mantis

- Для тестування використано **тестову систему Mantis**.
- Увійшов в систему як адміністратор, використовуючи облікові дані:
 - Логін: **administrator**
 - Пароль: **root**

1.3 Виконання сценарію вручну

- Підготовлено два тестових користувача для виконання сценарію:
 - **User1:** Email: user1@example.com, Пароль: password123
 - **User2:** Email: user2@example.com, Пароль: password123
- Відкрито профіль користувача в системі Mantis.
- Виконано редагування даних користувача, змінивши email та ім'я для кожного користувача.

2. Реалізація тесту в JMeter

2.1 Налаштування JMeter

- Завантажено та встановлено **Apache JMeter**.
- Створено проект у JMeter, що включає три групи потоків:
 - **Група 1:** Логін (авторизація як адміністратор).
 - **Група 2:** Оновлення профілю користувача.
 - **Група 3:** Перевірка змін профілю.

2.2 Використання елементів JMeter

- **Екстрактори:** Для отримання токенів та ідентифікаторів користувачів з відповіді сервера використано **JSON Extractor**.
- **Асerti:** Для перевірки коректності відповіді (статус 200) після кожного запиту додано **Response Assertion**.
- **Змінні:** Створено змінні для зберігання даних користувачів (ім'я, email).

- **Властивості:** Використано **User Defined Variables** для конфігурації даних користувачів.
- **CSV Data Set Config:** Використано для надання даних для користувачів у кожному потоці.

2.3 Виконання тесту

- Запуск тесту в JMeter здійснювався для трьох груп потоків:
 1. **Перший потік** виконує логін як адміністратор з обліковими даними:
username = administrator, password = root.
 2. **Другий потік** оновлює дані користувача, змінивши email та ім'я.
 3. **Третій потік** перевіряє, чи зміни відображаються в системі.

3. Тестові дані

CSV файл з тестовими даними для користувачів:

```
username,password,email,display_name
administrator,root,admin@example.com,Admin User
user1,password123,user1@example.com,User One
user2,password123,user2@example.com,User Two
```

4. JMeter сценарій (Test_Scenario.jmx)

```
<?xml version="1.0" encoding="UTF-8"?>
<TestPlan guiclass="TestPlanGui" testclass="TestPlan" testname="Test
Scenario" enabled="true">
  <ThreadGroups>
    <ThreadGroup guiclass="ThreadGroupGui" testclass="ThreadGroup"
testname="Login + User Update" enabled="true">
      <stringProp name="ThreadGroup.num_threads">5</stringProp>
      <stringProp name="ThreadGroup.ramp_time">5</stringProp>
      <stringProp name="ThreadGroup.loops">1</stringProp>
      <sampler>
        <HTTPSamplerProxy guiclass="HttpTestSampleGui"
testclass="HTTPSamplerProxy" testname="Login Request" enabled="true">
          <stringProp
name="HTTPSampler.domain">localhost</stringProp>
          <stringProp
name="HTTPSampler.port">8080</stringProp>
          <stringProp
name="HTTPSampler.path">/login</stringProp>
```

```

        <stringProp
name="HTTPSampler.method">POST</stringProp>
        <elementProp name="Arguments"
elementType="Arguments">
            <collectionProp name="Arguments.arguments">
                <elementProp name="username"
elementType="Argument">
                    <stringProp
name="Argument.name">username</stringProp>
                    <stringProp
name="Argument.value">administrator</stringProp>
                    </elementProp>
                    <elementProp name="password"
elementType="Argument">
                        <stringProp
name="Argument.name">password</stringProp>
                        <stringProp
name="Argument.value">root</stringProp>
                        </elementProp>
                    </collectionProp>
                </elementProp>
            </HTTPSamplerProxy>
        </sampler>
        <assertions>
            <ResponseAssertion guiclass="AssertionGui"
testclass="ResponseAssertion" testname="Response Status Check"
enabled="true">
                <stringProp
name="ResponseAssertion.assumeSuccess">true</stringProp>
                <stringProp
name="ResponseAssertion.test_field">Assertion.response_code</stringProp>
                <stringProp
name="ResponseAssertion.test_type">1</stringProp>
                <stringProp
name="ResponseAssertion.value">200</stringProp>
            </ResponseAssertion>
        </assertions>

```

```
        </ThreadGroup>
    </ThreadGroups>
</TestPlan>
```

5. Результати

- Тестування успішно завершено з виконанням усіх трьох груп потоків у JMeter.
- Всі запити до системи Mantis повернули коректний статус 200.
- Дані користувачів були успішно оновлені в системі та перевірені в процесі тесту.

Висновки:

Тестування було проведено успішно, і сценарій для автоматизованого тестування в JMeter підтвердив свою працездатність. Виконано перевірку основних операцій: логін, оновлення даних користувача, перевірка змін профілю. Тестова система Mantis коректно обробила запити, і всі очікувані зміни відобразились в системі.