

<b>Перевірка</b>
<b>1.Post /user</b>
<b>1.1Валідність даних</b>
1.1.1Переконатися, що запит з усіма необхідними полями успішно створює користувача.
1.1.2Перевірити, що при відсутності будь-якого обов'язкового поля запит повертає помилку.
<b>1.2 Статус-коди</b>
1.2.1Перевірити, що успішний запит повертає статус-код 200.
1.2.2Перевірити, що невалідний запит повертає статус-код помилки
<b>1.3. Відповідь API</b>
1.3.1Перевірити, що тіло відповіді містить підтвердження створення користувача
1.3.2Перевірити, що тіло відповіді містить відповідні поля
-code
-type
-message
1.3.3Перевірити, що тіло відповіді має формат JSON.
<b>1.4.Перевірка полів запиту</b>
1.4.1 поле "id"
-Перевірити, наявність поля "id"
-Перевірити, що поле "id" заповнене валідними даними
1.4.2 поле "username"
-Перевірити, наявність поля "username"
-Перевірити, що поле "username" заповнене валідними даними
1.4.3 поле "firstName"
-Перевірити наявність поля "firstName"
-Перевірити, що поле "firstName" заповнене валідними даними
1.4.4 поле "lastName"
-Перевірити наявність поля "lastName"
-Перевірити, що поле "lastName" заповнене валідними даними

1.4.4 поле "email"
-Перевірити наявність поля "email"
-Перевірити, що поле "email" заповнене валідними даними
1.4.5 поле "password"
-Перевірити наявність поля "password"
-Перевірити, що поле "password" заповнене валідними даними
1.4.6 поле "phone"
-Перевірити наявність поля "phone"
-Перевірити, що поле "phone" заповнене валідними даними
<b>1.5.Перевірка коректності URL адреси</b>
1.5.1 URL починається з правильного протоколу -https
1.5.2 URL містить відповідні шляхи
<b>1.6.Перевірка коректності типу запиту (метод HTTP)</b>
1.6.1 Використовується правильний HTTP метод- POST
<b>2.Get /user/{username}</b>
<b>2.1 Перевірка коректності URL адреси</b>
2.1.1 URL починається з правильного протоколу -https
2.1.2 URL містить відповідні шляхи
<b>2.2 Перевірка коректності типу запиту (метод HTTP)</b>
2.2.1 Використовується правильний HTTP метод- GET
<b>2.2 Валідне ім'я користувача</b>
2.2.1 Переконайтеся, що відповідь на запит з існуючим іменем користувача повертає статус-код 200.
2.2.2 Перевірити, що інформація про користувача відображається коректно.
<b>2.3 Невалідне ім'я користувача</b>
2.3.1 Переконайтеся, що запит з невалідним іменем користувача повертає статус-код 400.
2.3.2 Перевірити, що тіло відповіді містить повідомлення про те, що користувача не знайдено.
<b>2.4 Відповідь API</b>
2.4.1 Перевірити, що тіло відповіді має формат JSON.

2.4.2 Перевірити, що відповідь містить усі поля, що вказані у документації
<b>3.Get /user/login</b>
<b>3.1 Перевірка коректності URL адреси</b>
3.1.1 URL починається з правильного протоколу -https
3.1.2 URL містить відповідні шляхи
<b>3.2 Перевірка коректності типу запиту (метод HTTP)</b>
3.2.1 Використовується правильний HTTP метод- GET
<b>3.3 Статус-коди</b>
3.3.1 Перевірити, що успішний запит повертає статус-код 200.
3.3.2 Перевірити, що невалідний запит повертає статус-код помилки
<b>3.4.Перевірка полів запиту</b>
3.4.1 поле "id"
-Перевірити, наявність поля "id"
-Перевірити, що поле "id" заповнене валідними даними
3.4.2 поле "username"
-Перевірити, наявність поля "username"
-Перевірити, що поле "username" заповнене валідними даними
3.4.3 поле "firstName"
-Перевірити наявність поля "firstName"
-Перевірити, що поле "firstName" заповнене валідними даними
3.4.4 поле "lastName"
-Перевірити наявність поля "lastName"
-Перевірити, що поле "lastName" заповнене валідними даними
3.4.4 поле "email"
-Перевірити наявність поля "email"
-Перевірити, що поле "email" заповнене валідними даними
3.4.5 поле "password"
-Перевірити наявність поля "password"

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]