Audit
1. Post /user
1.1 Data validity
1.1.1 Verify that the request with all required fields successfully creates a user.
1.1.2 Check that the request returns an error in the absence of any mandatory field.
1.2 Status codes
1.2.1 Verify that a successful request returns a status code of 200.
1.2.2 Verify that an invalid request returns an error status code
1.3. API response
1.3.1 Verify that the response body contains confirmation of user creation
1.3.2 Check that the response body contains the appropriate fields
- code
- type
-message
1.3.3 Verify that the response body is in JSON format.
1.4. Checking the request fields
1.4.1 "id" field
-Check the presence of the "id" token
- Check that the "id" field is filled with valid data
1.4.2 "username" field
- Check the presence of the username "username"
- Check that the "username" field is filled with valid data
1.4.3 "first name" field
-Check for "firstName" pony
-Check that the "first name" field is filled with valid data
1.4.4 "lastName" field
-Check for "lastName" pony
- Check that the "lastName" field is filled with valid data

1.4.4 "email" field
- Check the presence of the "email" password
-Check that the "email" field is filled with valid data
1.4.5 "password" field
- Check the presence of the password "password"
- Check that the "password" field is filled with valid data
1.4.6 "phone" field
- Check for "phone" pony
- Check that the "phone" field is filled with valid data
1.5. Checking the correctness of the URL address
1.5.1URL starts with the correct protocol -HTTPS
1.5.2 The URL contains relevant paths
1.6. Checking the correctness of the request type (HTTP method)
1.6.1 The correct HTTP method is used - POST
2. Get /user/{username}
2.1 Checking the correctness of the URL address
2.1.1URL starts with the correct protocol -HTTPS
2.1.2 The URL contains relevant paths
2.2 Checking the correctness of the request type (HTTP method)
2.2.1 The correct HTTP method - GET - is used
2.2 Valid Username
2.2.1 Ensure that a response to a request with an existing username returns a status code 200.
2.2.2 Check that user information is displayed correctly.
2.3 Invalid username
2.3.1 Verify that a request with an invalid username returns a 400 status code.
2.3.2 Verify that the response body contains a message that the user was not found.
2.4 API response
2.4.1 Verify that the response body is in JSON format.

2.4.2 Verify that the response contains all the fields specified in the documentation
3. Get /user/login
3.1 Checking the correctness of the URL address
3.1.1URL starts with the correct protocol -HTTPS
3.1.2 The URL contains relevant paths
3.2 Checking the correctness of the request type (HTTP method)
3.2.1 The correct HTTP method - GET - is used
3.3 Status codes
3.3.1 Verify that a successful request returns a status code of 200.
3.3.2 Verify that an invalid request returns an error status code
3.4. Verification of request fields
3.4.1 "id" field
-Check the presence of the "id" token
- Check that the "id" field is filled with valid data
3.4.2 "username" field
- Check the presence of the username "username"
- Check that the "username" field is filled with valid data
3.4.3 "first name" field
-Check for "firstName" pony
-Check that the "first name" field is filled with valid data
3.4.4 "lastName" field
-Check for "lastName" pony
- Check that the "lastName" field is filled with valid data
3.4.4 "email" field
- Check the presence of the "email" password
-Check that the "email" field is filled with valid data
3.4.5 "password" field
- Check the presence of the password "password"

- Check that the "password" field is filled with valid data
3.4.6 "phone" field
- Check for "phone" pony
- Check that the "phone" field is filled with valid data
3.5. API response
3.5.1 Verify that the response body contains confirmation of user creation
3.5.2 Verify that the response body contains the appropriate fields
- code
- type
-message
3.5.3 Verify that the response body is in JSON format.