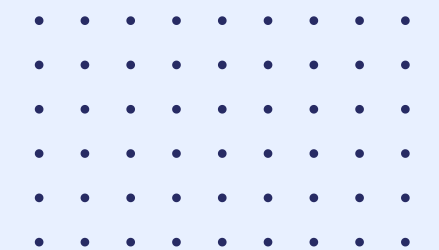


ТЕСТУВАННЯ SOAP I REST ЗАПИТІВ

МАЦЬКЕВИЧ МАР'ЯН



ЗАВДАННЯ

Попрактикуватися Soap та Rest запитів.

Після вироблення навичок створення коректних запитів ви повинні опанувати мистецтво формування некоректних запитів, що викликають появу помилок.

SOAP

Для виконання завдання було використана програма SoapUI. Реалізовано стартовий навчальний Soap проект.

ServiceSoapBinding

OverviewService EndpointsWSDL ContentWS-I Compliance

WSDL Definition

WSDL URL

Namespace

Binding

SOAP Version

Style

WS-A version

WSDL-WADL/sample-service.wsdl

http://www.soapui.org/sample/

ServiceSoapBinding

SOAP 1.1

RPC

NONE

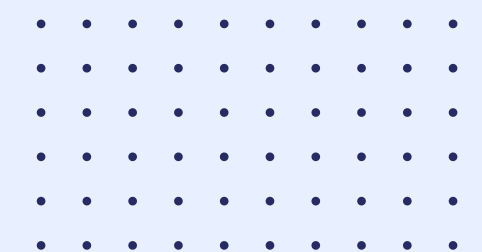
Definition Parts

sample-service.wsdl

WSDL-WADL/sample-service.wsdl

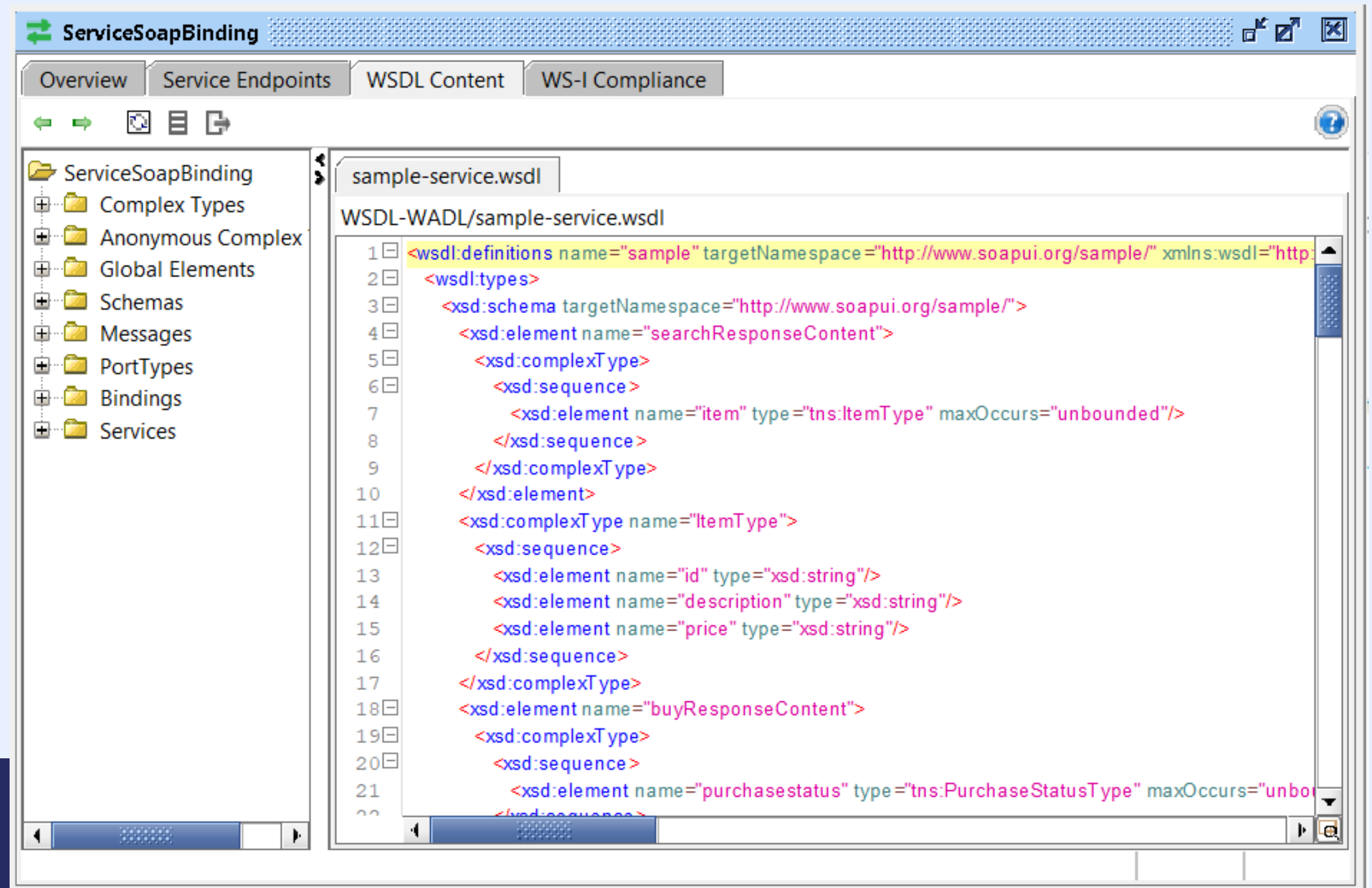
Operations

| Name | Use | One-Way | Action |
|--------|---------|---------|-------------------------------------|
| login | Literal | false | http://www.soapui.org/sample/login |
| logout | Literal | false | http://www.soapui.org/sample/logout |
| search | Literal | false | http://www.soapui.org/sample/search |
| buy | Literal | false | http://www.soapui.org/sample/buy |

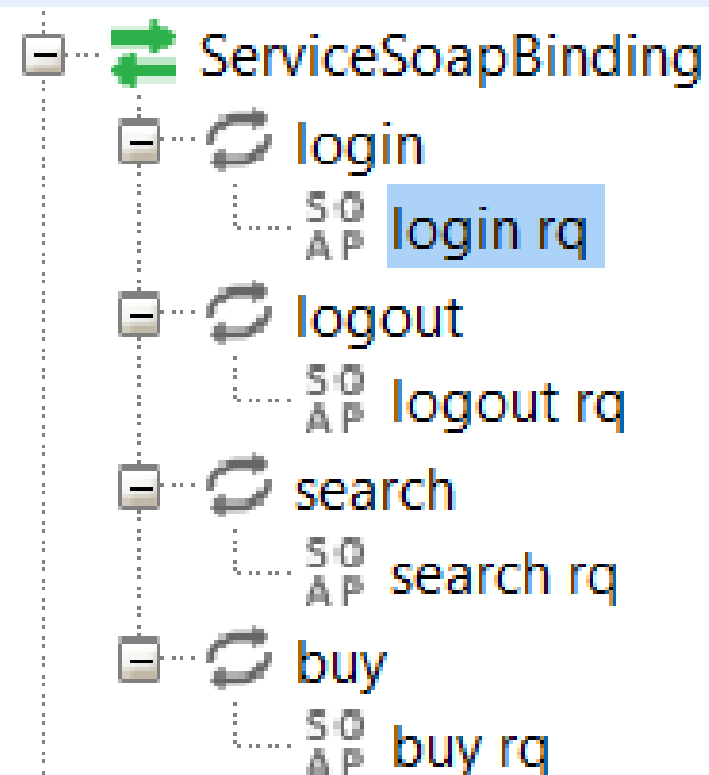


WSDL

Для виконання завдання було використана програма SoapUI. Реалізовано стартовий навчальний Soap проєкт.



ЗАПИТИ



Three screenshots of SOAP request XMLs from a testing tool:

logout rq (URL: http://127.0.0.1:8088/mockService)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <sam:logout>
      <sessionid>4833623015354377</sessionid>
    </sam:logout>
  </soapenv:Body>
</soapenv:Envelope>
```

login rq (URL: http://127.0.0.1:8088/mockSen)

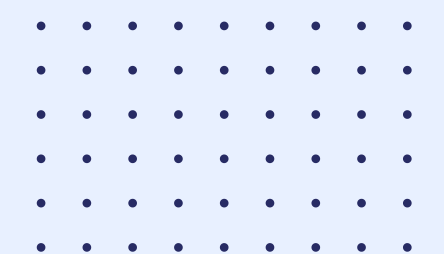
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <sam:login>
      <username>Login123</username>
      <password>Login123</password>
    </sam:login>
  </soapenv:Body>
</soapenv:Envelope>
```

search rq (URL: http://127.0.0.1:8088/mockService)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <sam:search>
      <sessionid>0154144954954053</sessionid>
      <searchstring>Login123</searchstring>
    </sam:search>
  </soapenv:Body>
</soapenv:Envelope>
```

buy rq (URL: http://127.0.0.1:8088/mockService)

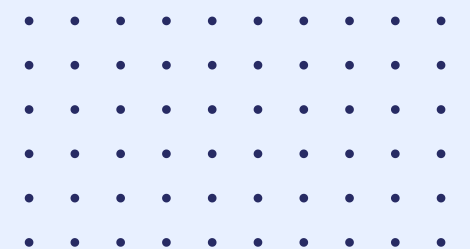
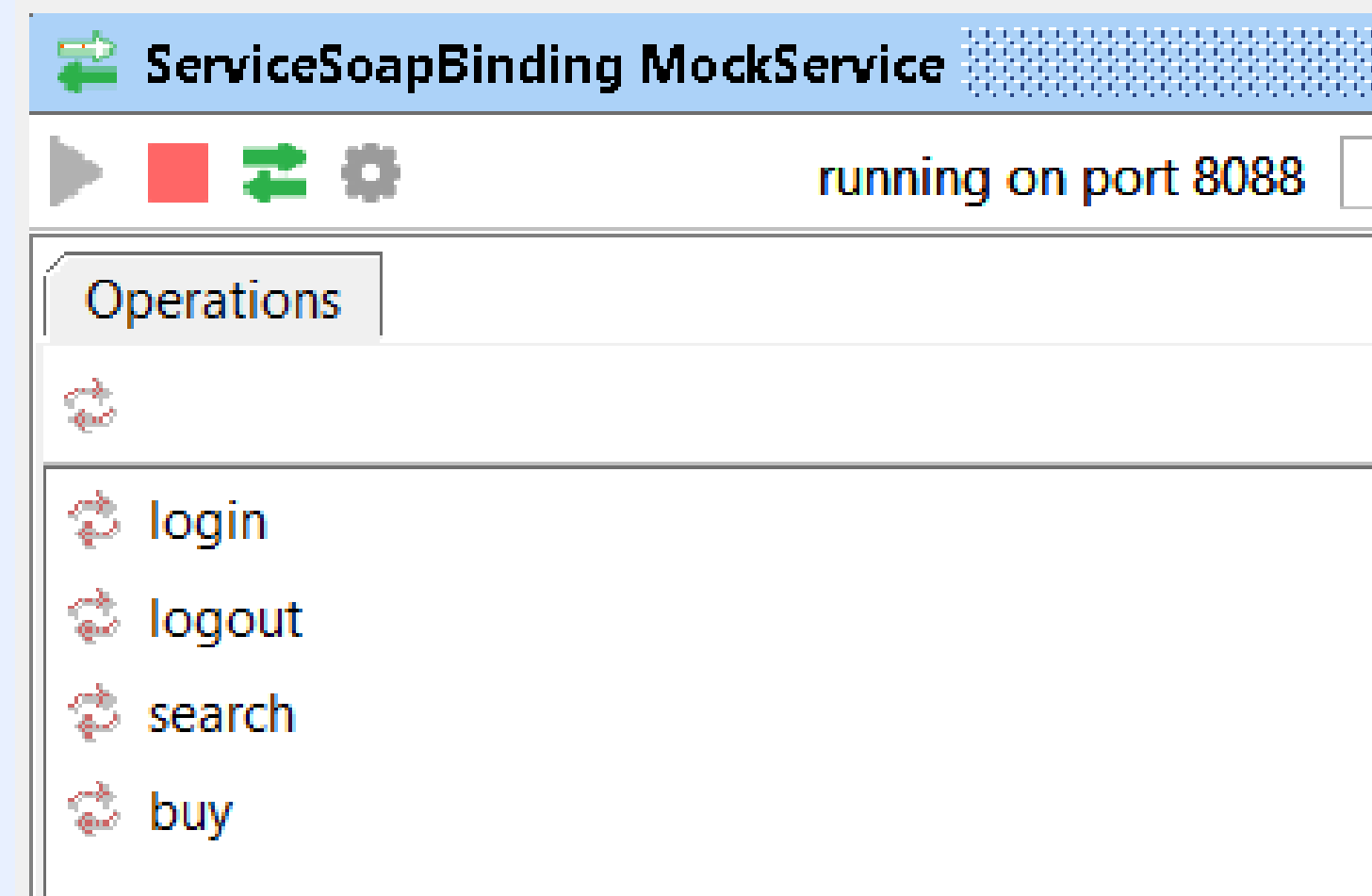
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <sam:buy>
      <sessionid>?</sessionid>
      <buystring>?</buystring>
    </sam:buy>
  </soapenv:Body>
</soapenv:Envelope>
```



WEB SERVICE MOCKING

Мокінг веб-сервісу дуже корисний у проектах, де впровадження веб-сервісу ще не розпочато або ще не завершено, або де ви з якоїсь іншої причини не можете отримати доступ до веб-сервісу.

Кінцеві точки зразків запитів посилаються на макет служби.



LOGIN MOCKING

The screenshot shows the SoapUI MockService interface for the 'login' operation. The left sidebar lists the service 'ServiceSoapBinding MockService' and its operations: 'login' (with 'Ok Response', 'Invalid Login Fault', and 'Already Logged In Fault'), 'logout' (with 'Ok Response' and 'Invalid Session Id Fault'), 'search', and 'buy'. The main panel is for the 'login' operation, showing 'Interface: ServiceSoapBinding' and 'Operation: login'. Under 'MockResponses', three responses are listed: 'Ok Response', 'Invalid Login Fault', and 'Already Logged In Fault'. The 'Dispatch' dropdown is set to 'SCRIPT' and the 'Default Response' is 'Ok Response'. A script is provided for the mock operation:

```
1 def groovyUtils = new com.eviware.soapui.support.GroovyUtils( context )
2 def holder = groovyUtils.getXmlHolder( mockRequest.requestContent )
3
4 def username = holder.getNodeValue( "//username" )
5 def password = holder.getNodeValue( "//password" )
6 def sum = username + password
7
8 // validate
9 if( sum.length() > 25 || sum.length() < 10 || !username.startsWith("Login") || !password.endsWith( "123" ) )
10     return "Invalid Login Fault"
11
12 if( context.hasProperty( sum ) )
13     return "Already Logged In Fault"
14
15 // seems ok
16 def sessionid = String.valueOf( Math.random() ).substring( 2 )
17
18 context.setProperty( sum, sessionid )
19 context.setProperty( sessionid, sum )
20
21 requestContext.sessionid = sessionid
22 return "Ok Response"
23
```

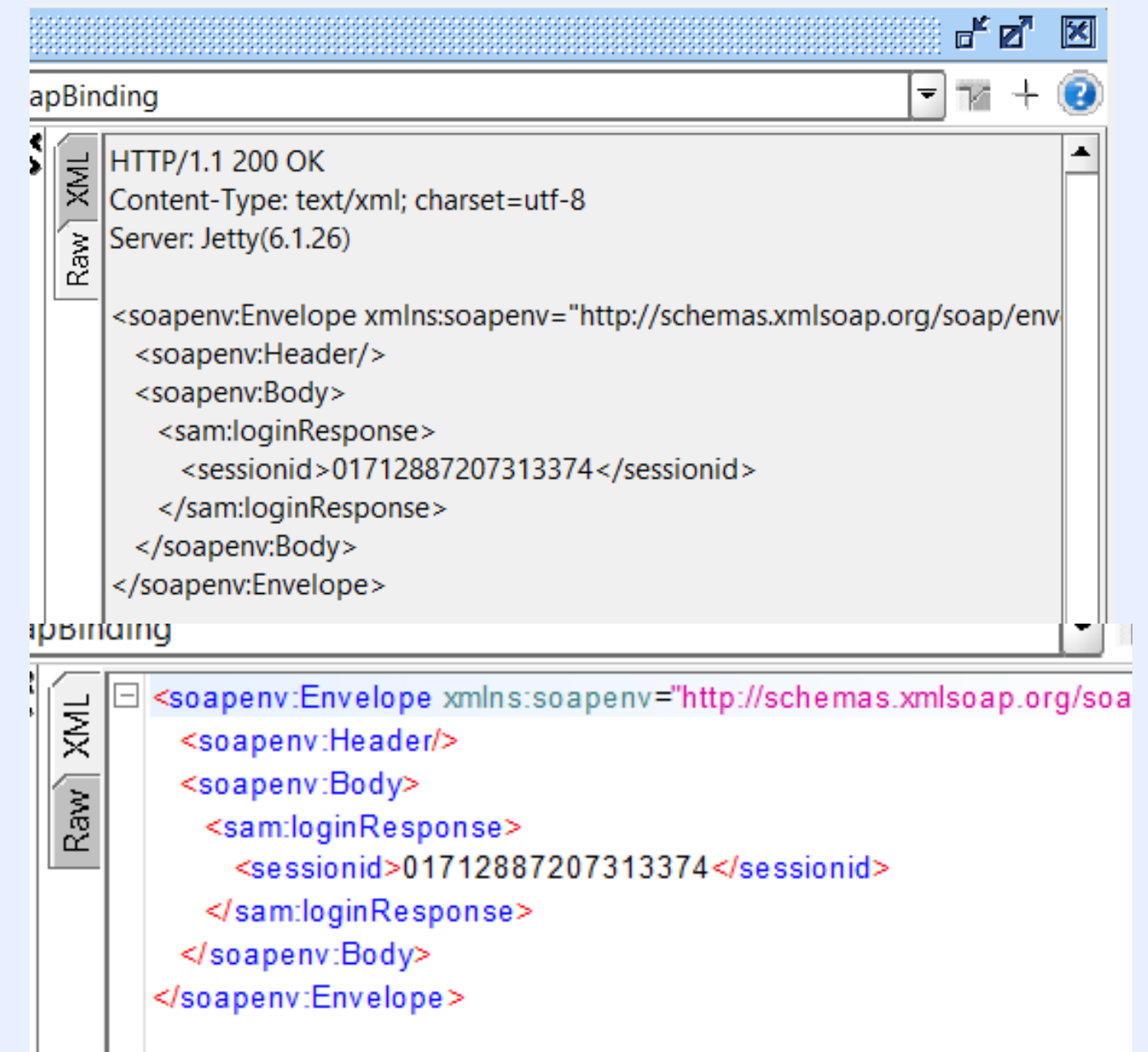
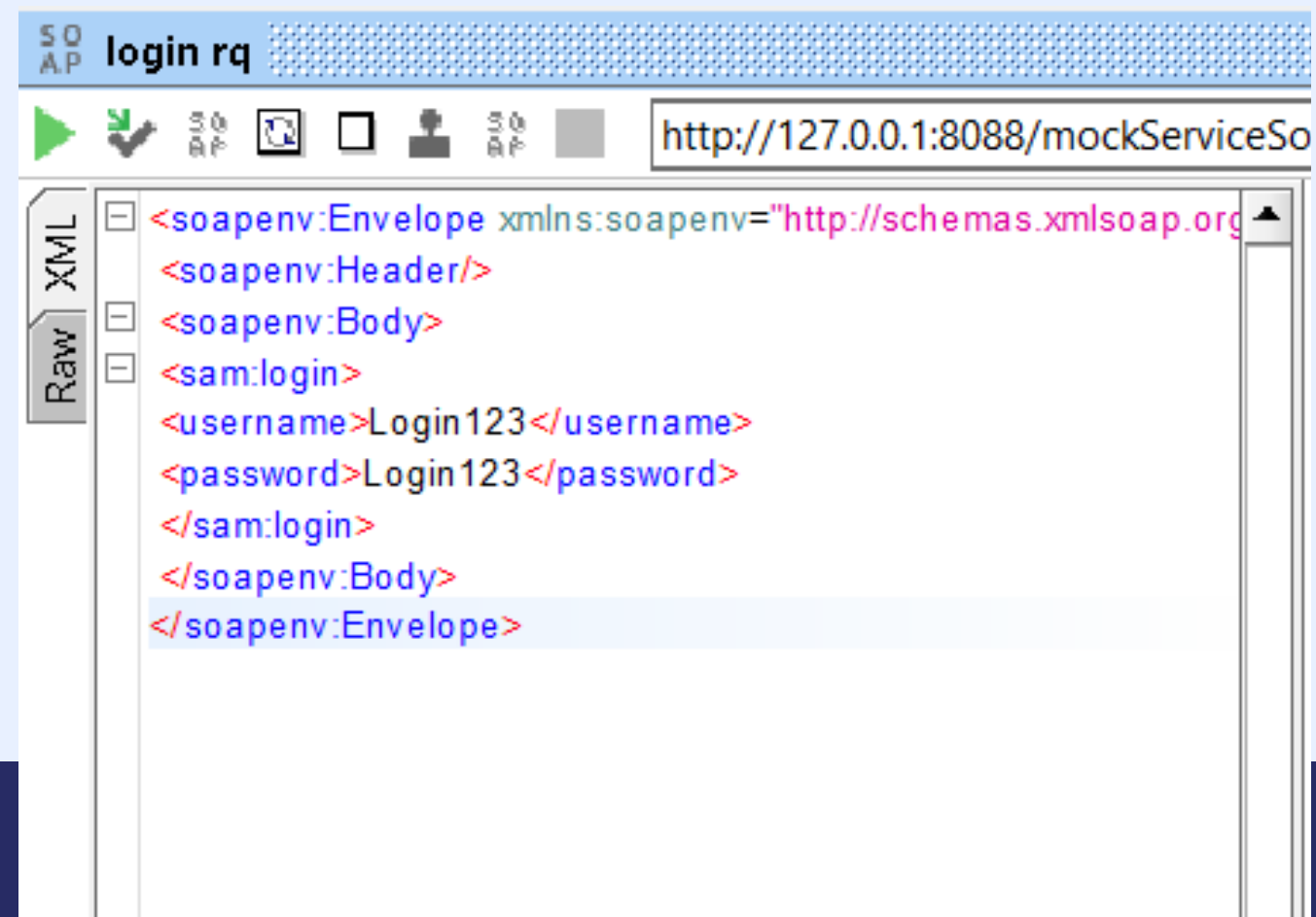
At the bottom, there is a 'Dispatch (SCRIPT)' button.

The right side of the image displays three screenshots of the XML structure for different mock responses in SoapUI:

- Ok Response:** Shows a SOAP envelope with a header and a body containing a `<sam:loginResponse>` element with a `<sessionid>${sessionid}</sessionid>` child.
- Invalid Login Fault:** Shows a SOAP envelope with a header and a body containing a `<soapenv:Fault>` element. The fault details include `<faultcode>Client</faultcode>`, `<faultstring>Invalid Login</faultstring>`, and a `<detail>` element containing `<sam:loginFault>The login credentials are invalid</sam:loginFault>`.
- Already Logged In Fault:** Shows a SOAP envelope with a header and a body containing a `<soapenv:Fault>` element. The fault details include `<faultcode>Client</faultcode>`, `<faultstring>Already Login</faultstring>`, and a `<detail>` element containing `<sam:loginFault>The specified credentials are already logged in</sam:loginFault>`.

ЗАПУСК ЗАПИТУ (LOGIN)

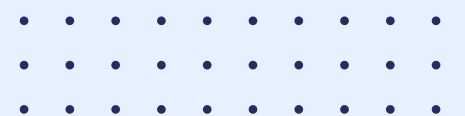
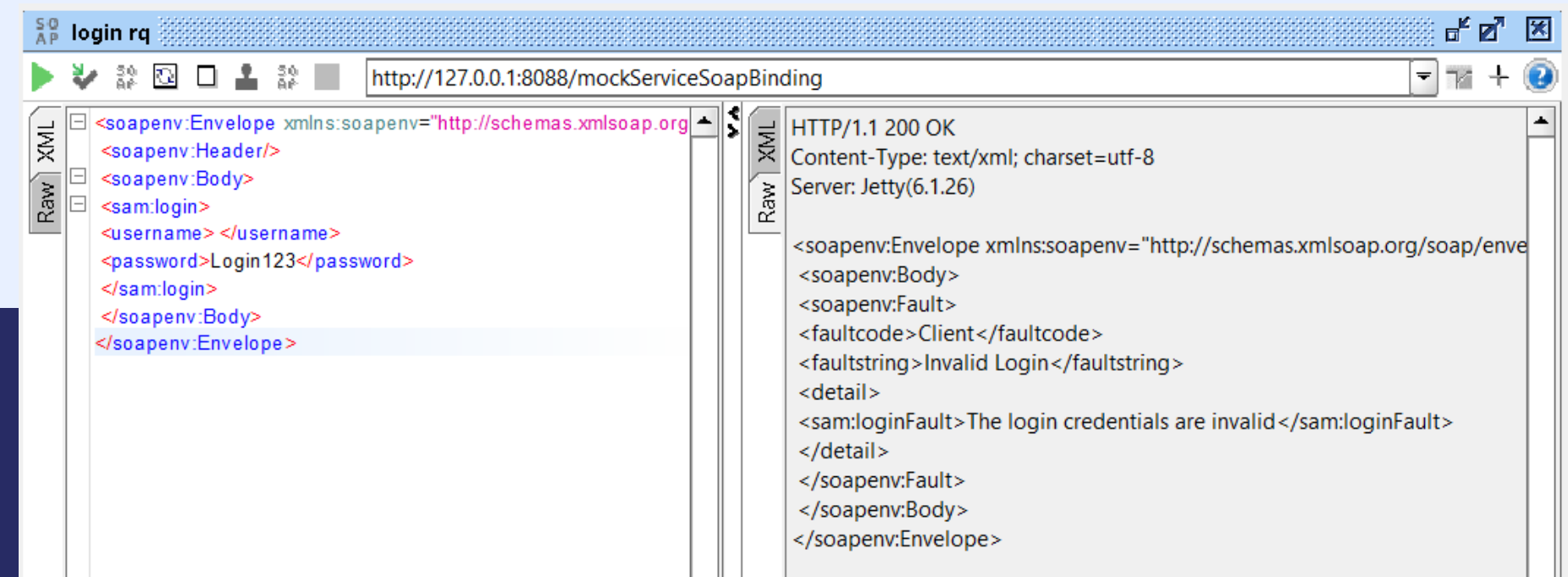
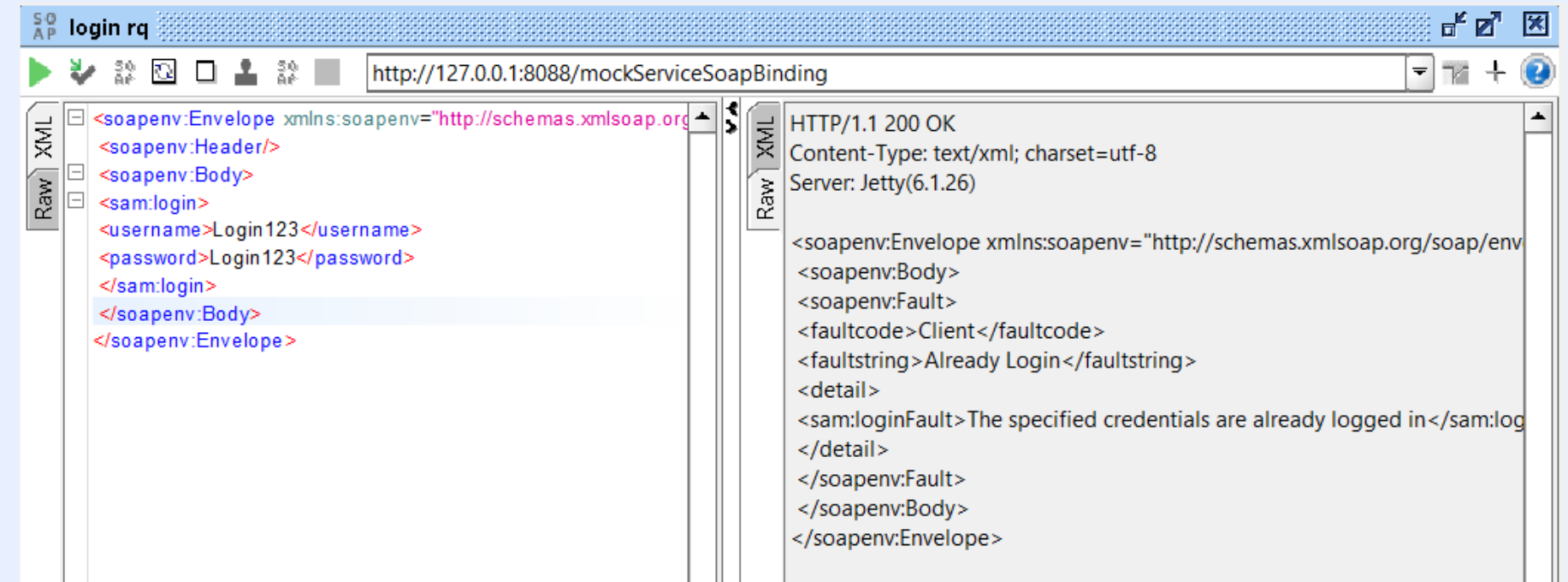
Спроба залогінитися. Якщо правильно введені дані, то формується id (токен) користувача.



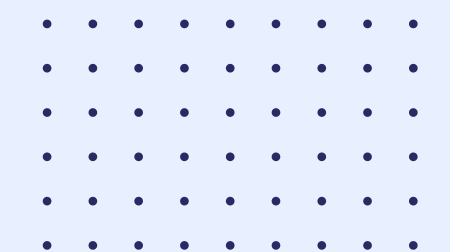
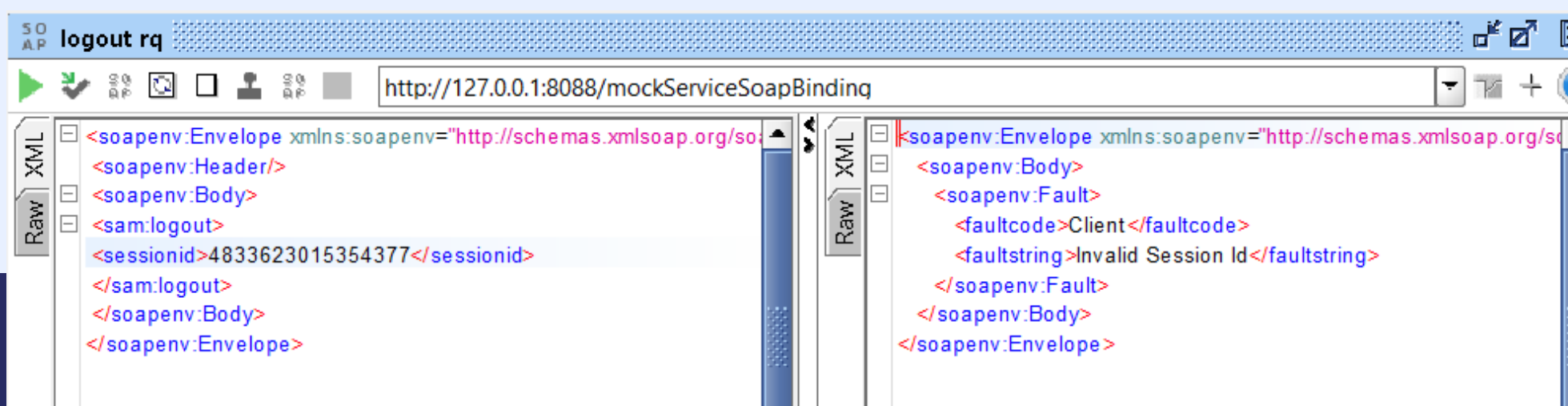
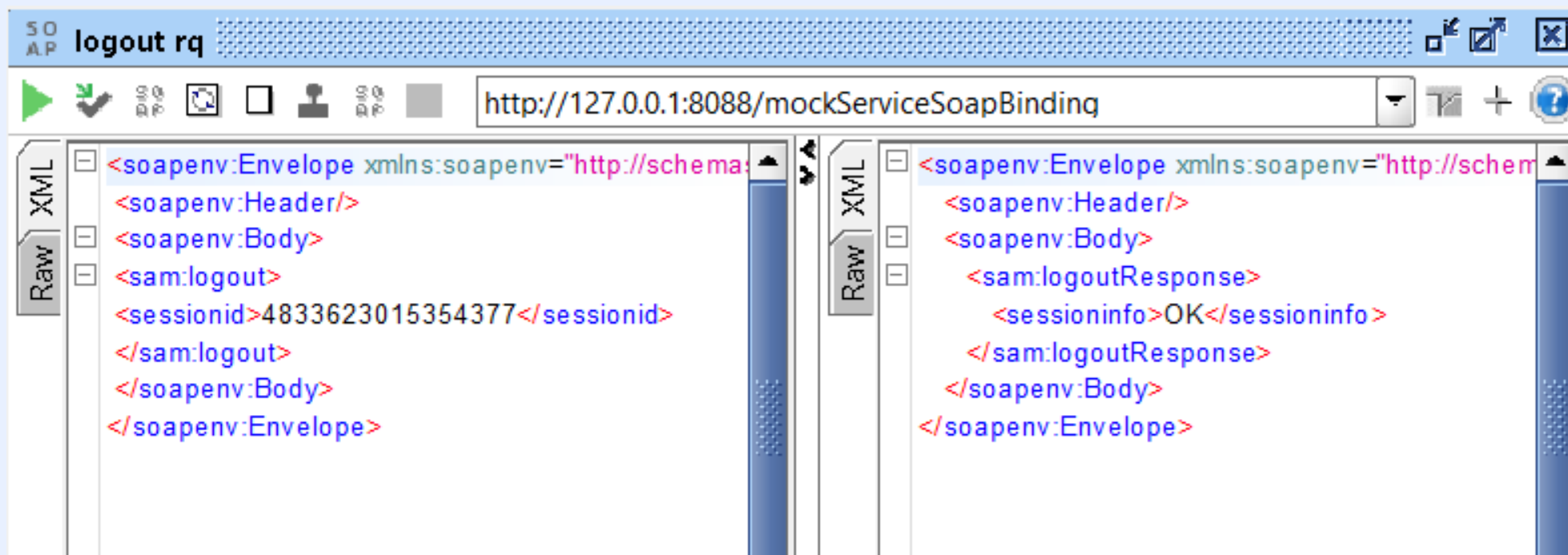
ЗАПУСК ЗАПИТУ (LOGIN)

Спроба викликати помилки:

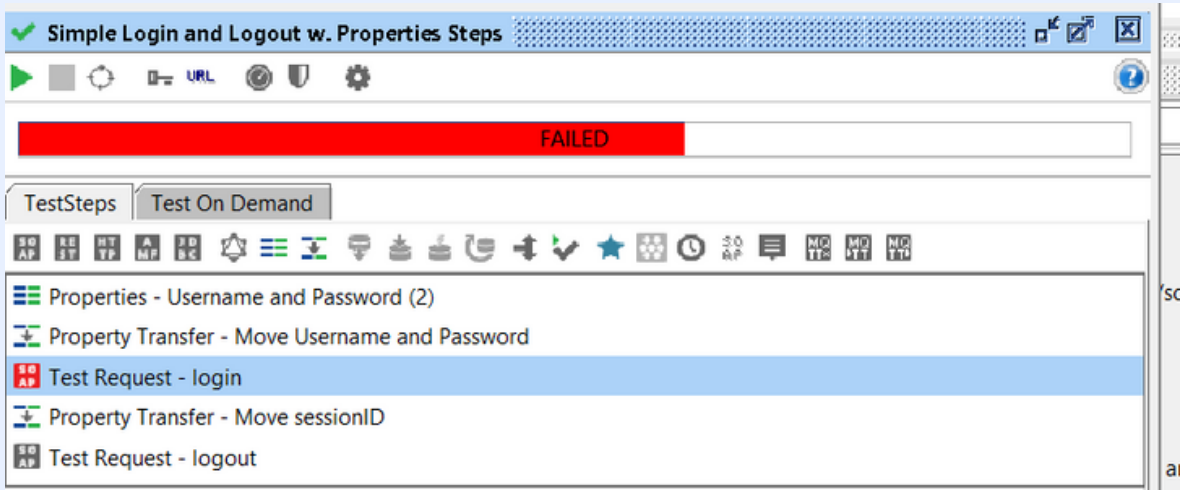
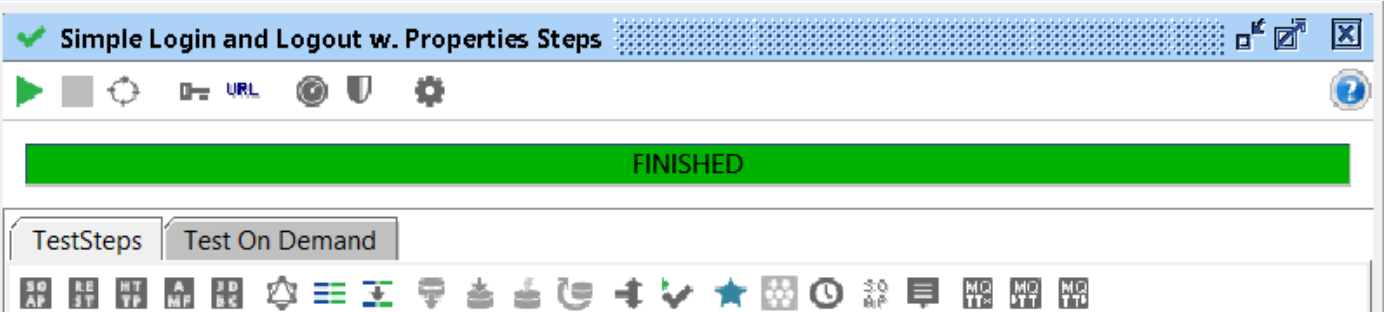
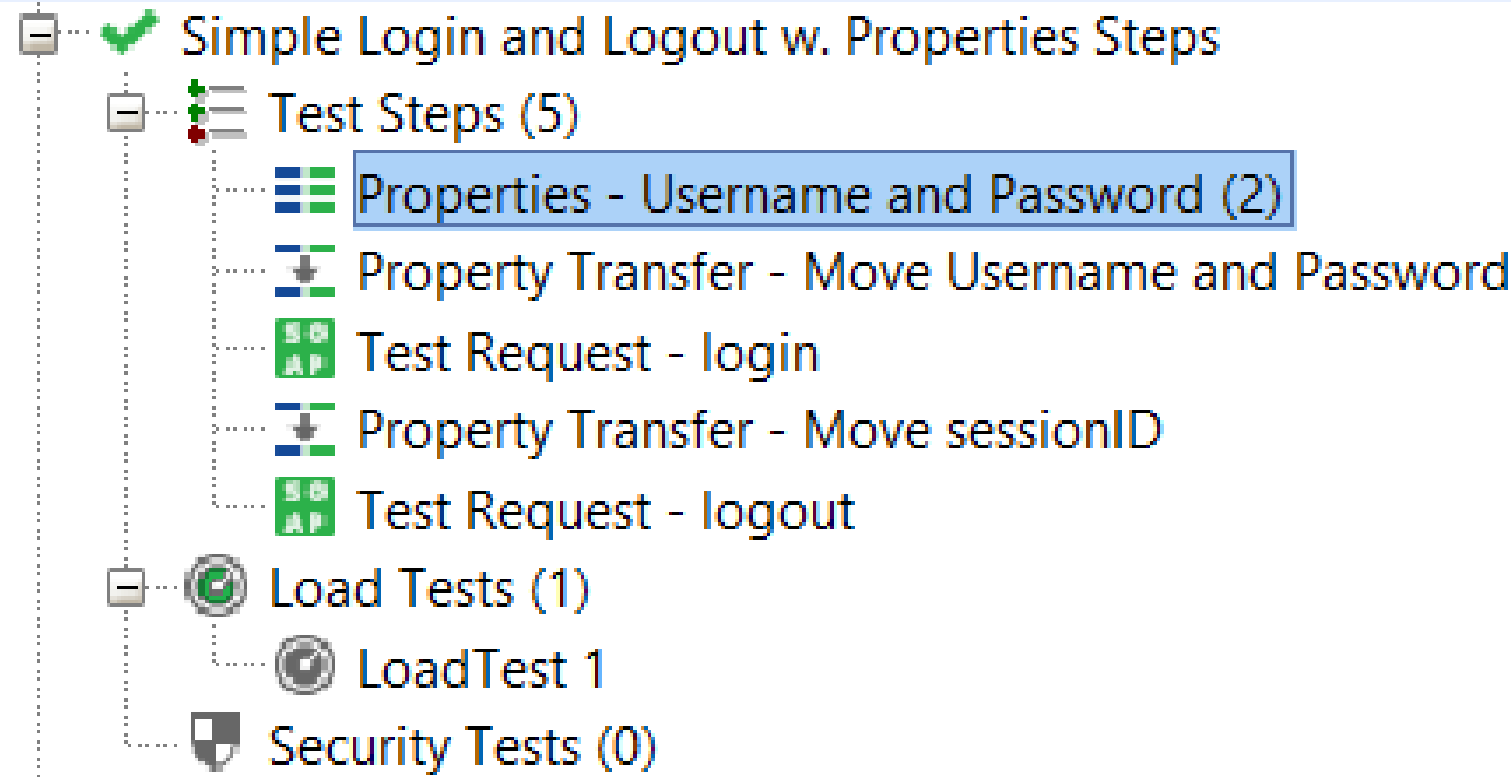
- 1) Спроба увійти вже залогіненого користувача
- 2) Спроба ввести некоректні дані



ЗАПУСК ЗАПИТУ (LOGOUT)



СТВОРЕННЯ ТЕСТІВ

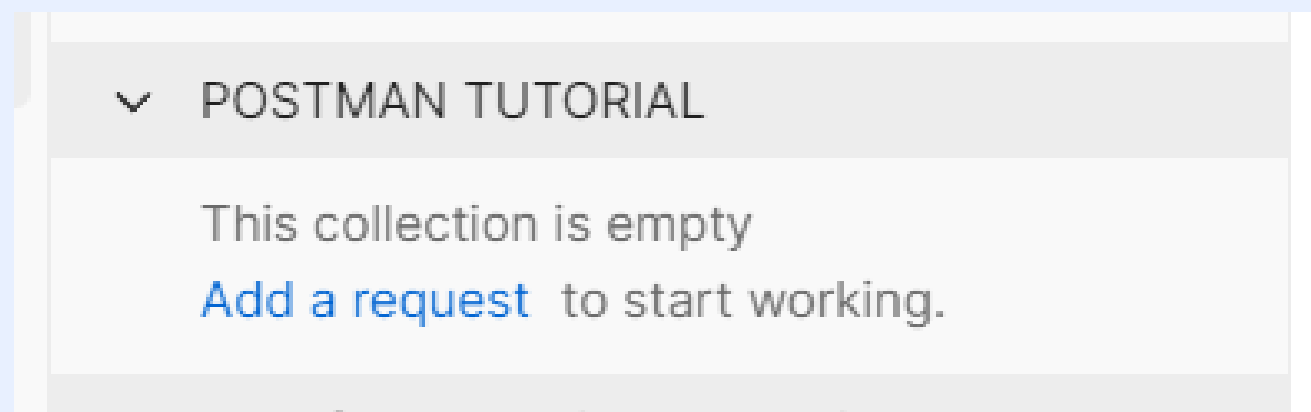


| Properties - Username and Password | |
|------------------------------------|---------------------------------|
| Load from: | |
| Name | Value |
| Username | Loginn\${=String.valueOf(Math.. |
| Password | ***** |

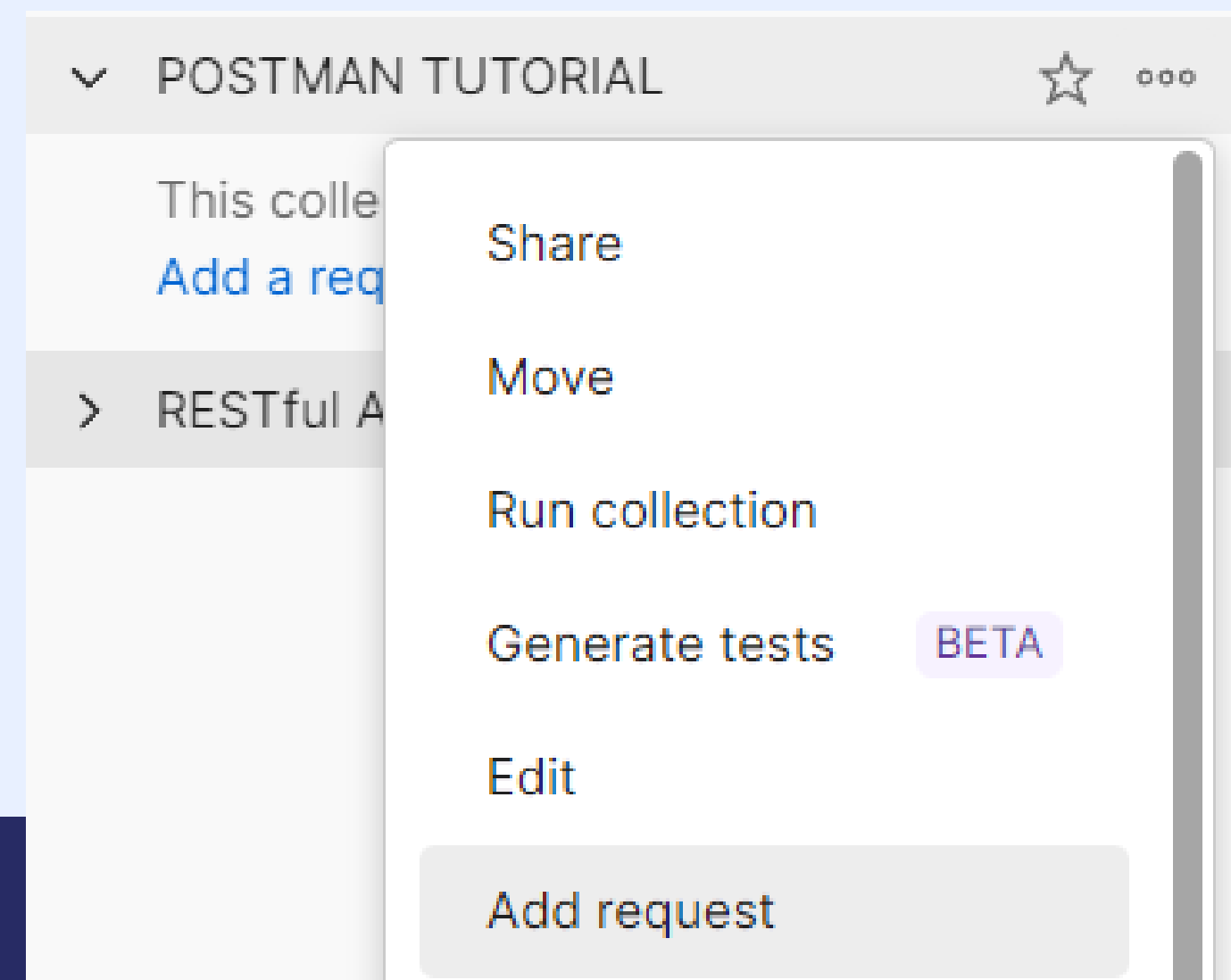
| LoadTest 1 | | | | | | | | | | | |
|-----------------------------------|-----|-----|-------|----------|------|---------|---------|------------|-----|------|--|
| Limit: | | 60 | | Seconds | | 100% | | | | | |
| Threads: | | 5 | | Strategy | | Simple | | Test Delay | | 1000 | |
| | | | | | | | | Random | | 0.5 | |
| Test Step | min | max | avg | last | cnt | tps | bytes | bps | err | rat | |
| Properties - Username and Pas.. | 0 | 0 | 0 | 0 | 387 | 0 | 0 | 0 | 0 | 0 | |
| Property Transfer - Move User... | 1 | 9 | 2,36 | 2 | 3872 | 117,... | 0 | 0 | 0 | 0 | |
| Test Request - login | 4 | 38 | 7,25 | 4 | 387 | 689,59 | 1183... | 2103... | 2 | 0 | |
| Property Transfer - Move sessi... | 1 | 6 | 2,19 | 4 | 3852 | 280,... | 0 | 0 | 0 | 0 | |
| Test Request - logout | 4 | 24 | 7,05 | 6 | 385 | 708,24 | 1024... | 1883... | 0 | 0 | |
| TestCase: | 10 | 77 | 18,81 | 16 | 387 | 265,81 | 2207... | 1515... | 2 | 0 | |

REST

Створення нової колекції

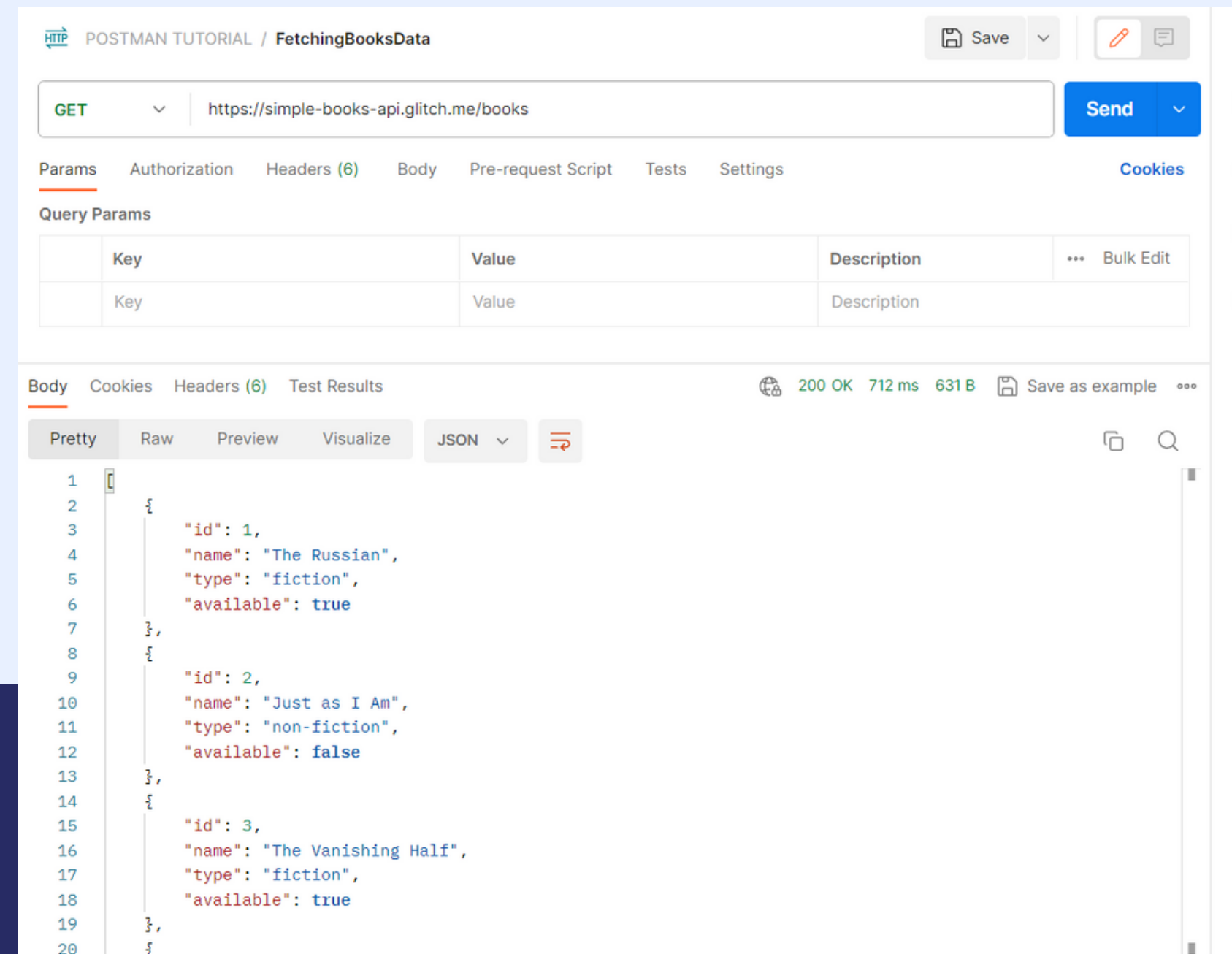


Додавання запиту

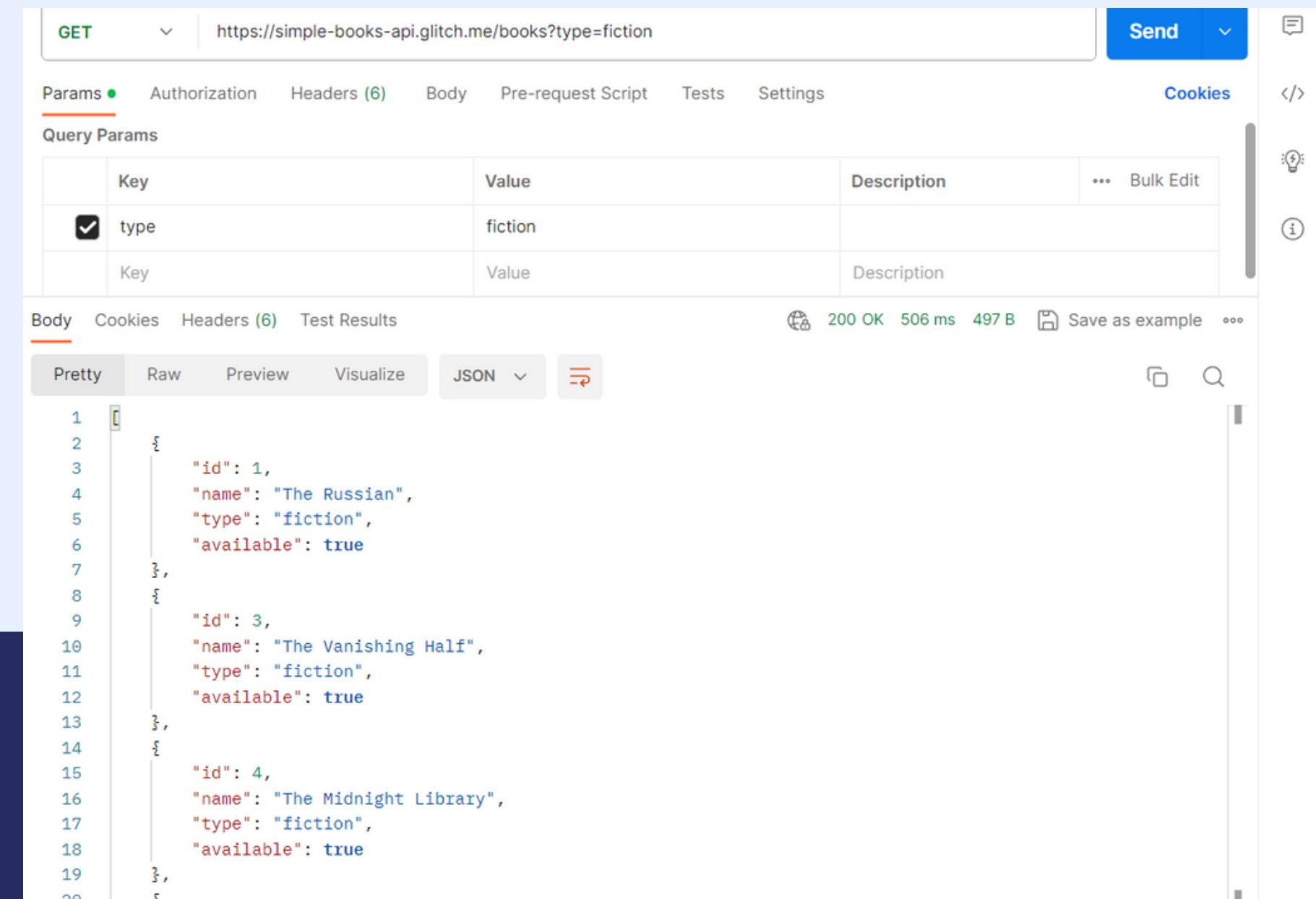


GET ЗАПИТИ

GET - <https://simple-books-api.glitch.me/books>

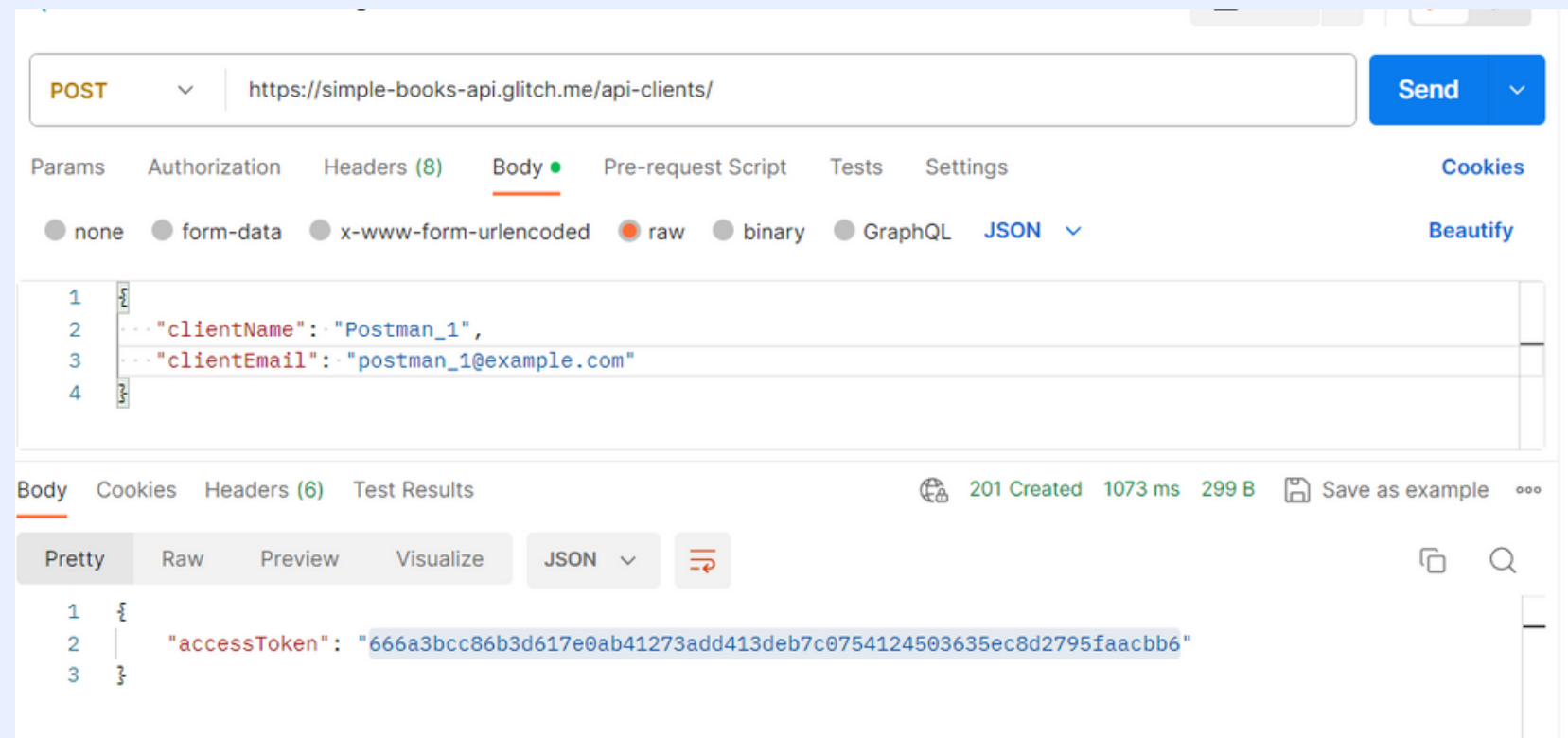


GET - <https://simple-books-api.glitch.me/books?type=fiction>

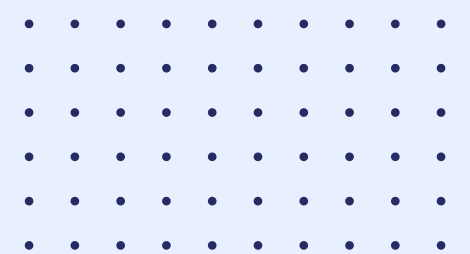
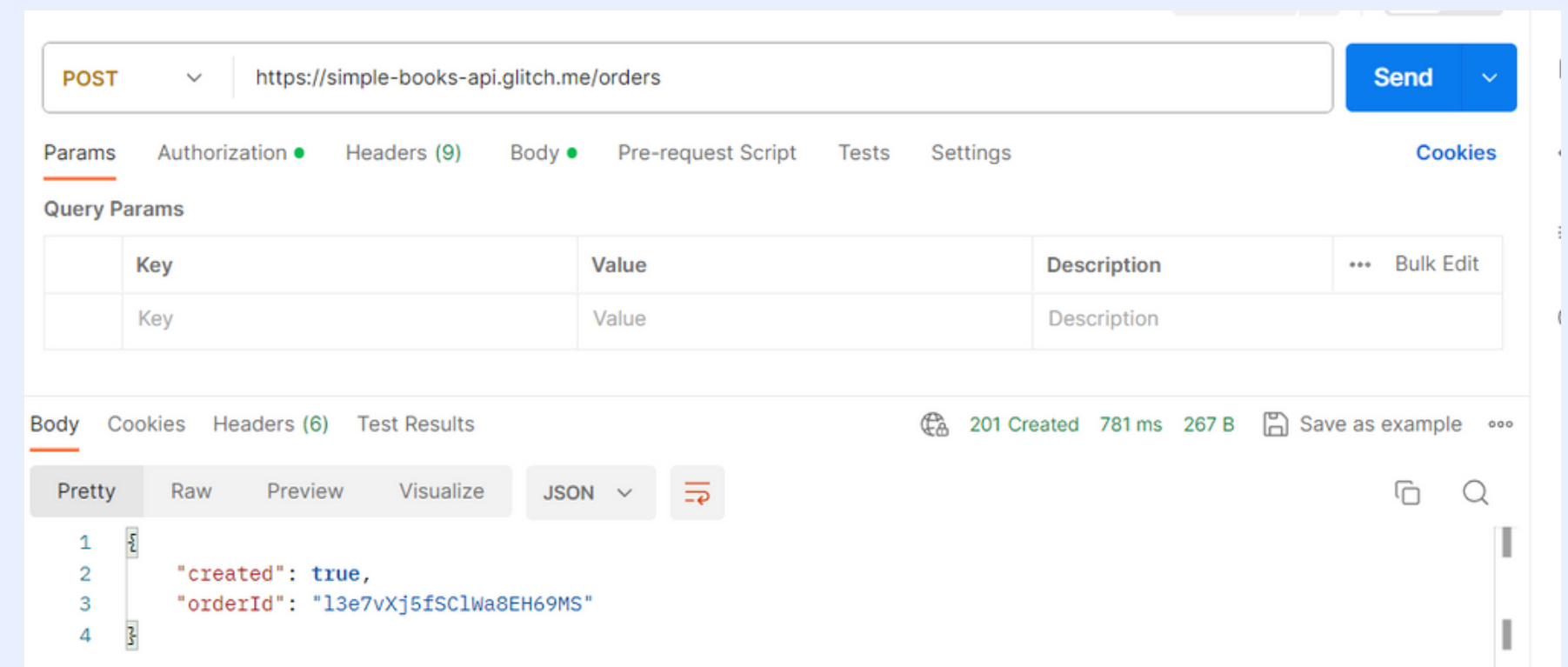


POST ЗАПИТИ

POST- <https://simple-books-api.glitch.me/api-clients/>

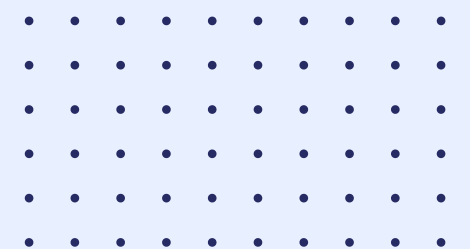
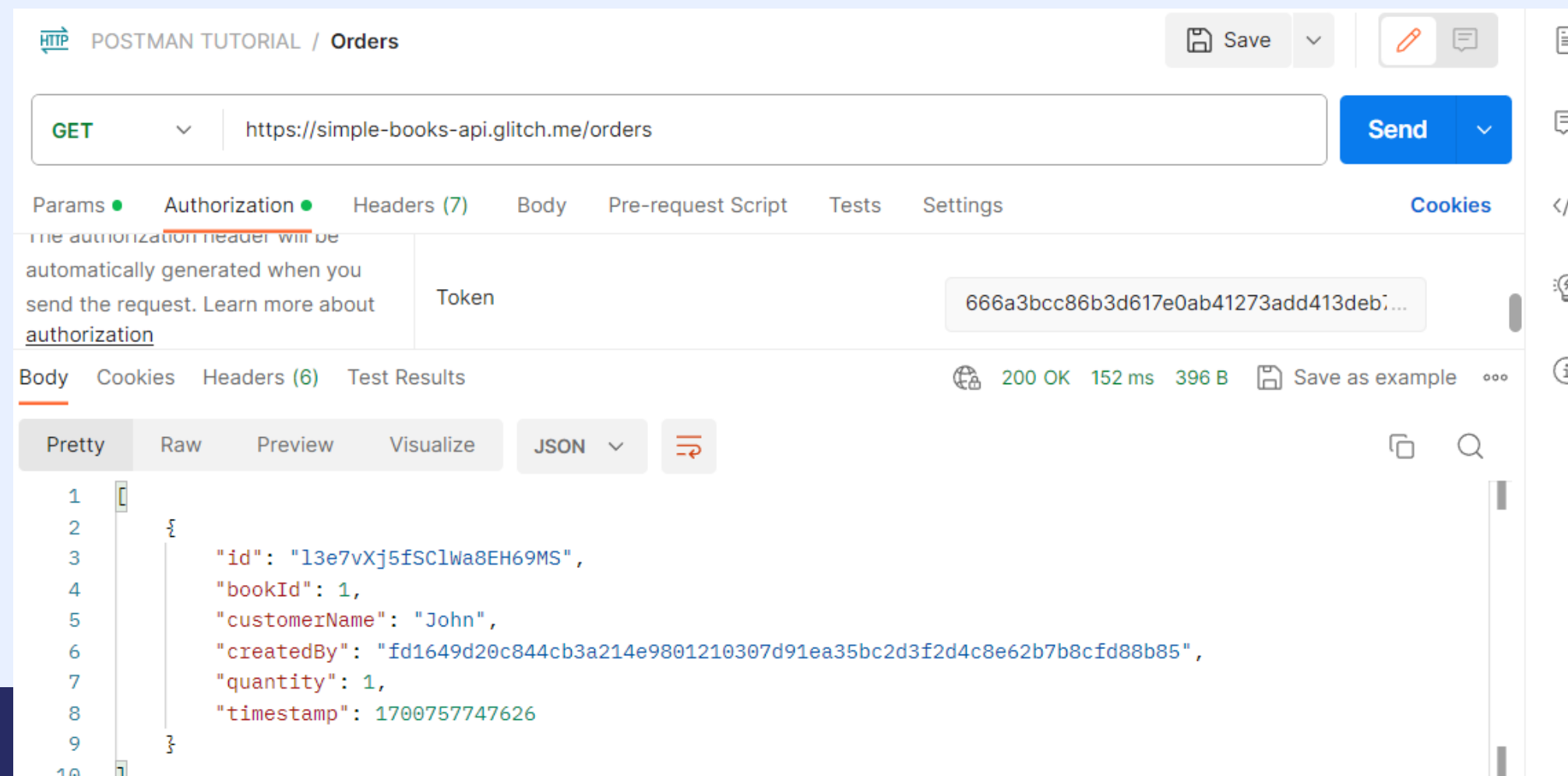


POST- <https://simple-books-api.glitch.me/orders>



РЕЗУЛЬТАТИ POST

GET- <https://simple-books-api.glitch.me/api-clients/>



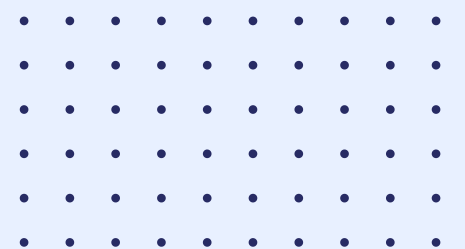
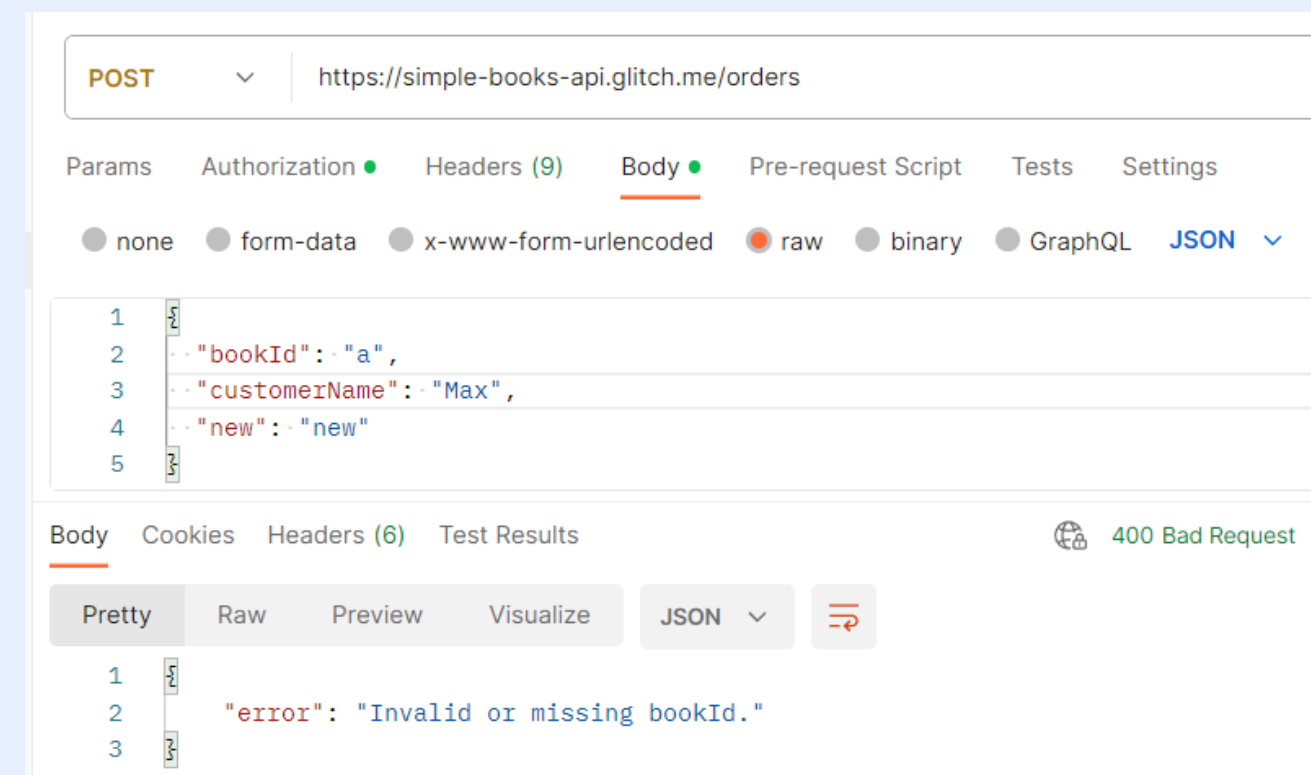
ПОМИЛКИ

Помилка при відправленні некоректного запиту

```
▼ POST http://localhost:5000/api/auth
Error: connect ECONNREFUSED 127.0.0.1:5000

▼ Request Headers ⓘ
  Content-Type: "application/json"
  User-Agent: "PostmanRuntime/7.35.0"
  Accept: "*/*"
  Postman-Token: "7b7e66ac-a502-45bb-b4fc-437347af4e7a"
  Host: "localhost:5000"
  Accept-Encoding: "gzip, deflate, br"
  Connection: "keep-alive"
► Request Body ↗
```

Помилка при введенні некоректної змінної



ВИСНОВОК

У процесі вивчення та практики з програмами SOAP UI та POSTMAN для роботи з SOAP і REST запитами, я отримав навички створення правильних та ефективних запитів до веб-сервісів.

Додатково, важливим етапом є опанування мистецтва створення некоректних запитів, що викликають помилки. Це дозволить ретельно тестувати та валідувати реакцію веб-сервісу на неправильні вхідні дані, що в свою чергу сприятиме підвищенню якості та надійності вашого програмного забезпечення.

