

Выполнить метод аутентификации через swagger в режиме инкогнито, получить список заявок в swagger

The image shows two separate instances of the Swagger UI interface, both running on localhost:8080. The top window displays a POST request to '/login' with a curl command, a Request URL of 'http://localhost:8080/login', and a Server response showing a JSON object with a token and user details. The bottom window displays a GET request to '/api/quantum_tasks' with a Request URL of 'http://localhost:8080/api/quantum_tasks', and a Server response showing a list of tasks.

Далее в insomnia/postman выполнить GET списка заявок: 401/403 для гостя, для создателя только его заявки

The screenshot shows the Postman application interface. A GET request is being made to 'http://localhost:8080/api/quantum_tasks'. The 'Authorization' tab is selected, and the 'Bearer Token' option is chosen. A 'Token' field is present but empty. The response status is '401 Unauthorized' with a message: 'empty auth header' and 'status: fail'.

Выполнить GET списка заявок - для модератора все заявки.

The screenshot shows two separate API requests made in Postman against the same endpoint: `http://localhost:8080/api/quantum_tasks`. Both requests are GET operations and include an Authorization header with a Bearer Token. The responses are both 200 OK, indicating success. The JSON data returned contains task details such as id_task, task_status, creation_date, id_user, conclusion_date, task_description, res_coeff_0, res_coeff_1, and gates_degrees.

```
200 OK | 36 ms | 5.63 KB | Save Response | 200 OK | 25 ms | 468 B | Save Response
```

```
[{"id_task": 1, "task_status": "new", "creation_date": "2025-10-12T15:54:26.194924+03:00", "id_user": 1, "conclusion_date": "2025-10-12T15:54:26.271286+03:00", "task_description": "turu-puru", "res_coeff_0": 1, "res_coeff_1": 0, "gates_degrees": 1}, {"id_task": 2, "task_status": "new", "creation_date": "2025-10-12T15:54:26.194924+03:00", "id_user": 1, "conclusion_date": "2025-10-12T15:54:26.271286+03:00", "task_description": "turu-puru", "res_coeff_0": 1, "res_coeff_1": 0, "gates_degrees": 1}, {"id_task": 3, "task_status": "new", "creation_date": "2025-10-12T15:54:26.194924+03:00", "id_user": 6, "conclusion_date": "2025-10-23T02:38:24.131402+03:00", "task_description": "mango bulk", "res_coeff_0": 0.4880824, "res_coeff_1": 0.9329453, "gates_degrees": 1}, {"id_task": 4, "task_status": "new", "creation_date": "2025-10-12T15:54:26.194924+03:00", "id_user": 6, "conclusion_date": "2025-10-23T02:38:24.131402+03:00", "task_description": "mango bulk", "res_coeff_0": 0.4880824, "res_coeff_1": 0.9329453, "gates_degrees": 1}
```

Выполнить PUT завершения заявки: для создателя 403 статус, для модератора успех и обновление полей.

The screenshot shows a single PUT request made in Postman to the endpoint `http://localhost:8080/api/quantum_tasks/31/resolve`. The request includes an Authorization header with a Bearer Token. The response is a 403 Forbidden status, indicating that the user lacks the necessary moderator rights to perform this action. The JSON response message states "access denied: moderator rights required".

```
403 Forbidden | 4 ms | 200 B | Save Response
```

```
{"message": "access denied: moderator rights required", "status": "fail"}
```

The screenshot shows the Postman interface with a successful API call to `Task_PUT_final`. The response body is a JSON object:

```

1  {
2     "id_task": 3,
3     "task_status": "запущен",
4     "creation_date": "2025-10-23T01:49:39.595864+03:00",
5     "id_user": 6,
6     "conclusion_date": "2025-10-23T02:38:24.1314822+03:00",
7     "task_description": "ноги буки",
8     "res_coeff_0": 0.48880284,
9     "res_coeff_1": 0.91949483,
10    "gates_degrees": null
11 }

```

Показать содержимое Redis через cmd, для сессий показать пользователя

The screenshot shows the Docker Desktop interface with the `redis-redis-1` container running. The logs tab displays Redis command interactions:

```

127.0.0.1:6379> AUTH
(error) ERR wrong number of arguments for 'auth' command
127.0.0.1:6379> AUTH password
OK
127.0.0.1:6379> KEYS
(error) ERR wrong number of arguments for 'keys' command
127.0.0.1:6379> KEYS *
(empty array)
127.0.0.1:6379> KEYS *
(empty array)
127.0.0.1:6379> KEYS *
(empty array)
127.0.0.1:6379> KEYS
(error) ERR wrong number of arguments for 'keys' command
127.0.0.1:6379> KEYS *
(empty array)
127.0.0.1:6379> KEYS *
(empty array)
127.0.0.1:6379> KEYS *
1) "rlp_service.jwt eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOiE3NjEyMTC3MTysImlhCI6Mtc2MTIxNDExNtwiaRFdXNlciI6OCviaXNFYnRtaW4iOnRydiV9.4f6AmY7gh4r6njk39dhqN89fdg1bhKmW0bark33kzIM"
127.0.0.1:6379> KEYS "rlp_service.jwt.*"
Invalid argument(s)
127.0.0.1:6379> KEYS "rlp_service.jwt.*"
1) "rlp_service.jwt eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOiE3NjEyMTC3MTysImlhCI6Mtc2MTIxNDExNtwiaRFdXNlciI6OCviaXNFYnRtaW4iOnRydiV9.4f6AmY7gh4r6njk39dhqN89fdg1bhKmW0bark33kzIM"
127.0.0.1:6379>

```