

Milestone 5 Report

Team 28





Testing APIs

Discourse CRUD APIs Tests

Test Scenario	Testing of Discourse Post CRUD APIs	Tester	Ansh Goyal	Date		
ID	Test Case	Pre Condition	Test Data Inputs	Expected Output	Actual Output	Status
1	Getting List of Existing Posts	1. The Discourse should be running. 2. Api_key and Api_username should be correct	No Inputs	List all the Posts content(raw) with their ids	As Expected	Successful
2	Getting specific Post	The Post(id) the user is looking for must exist	1. post_id which Exists. 2. post_id doesnt exist	1. content of post with post id... 2. error with input id	As Expected	Successful
3	Creating a post	To Create a new ticket .. we must give Api_key and Api_username in header... Category is general(4)	1. length(Title) < 15.... 2.length(Content) < 20 letters... 3.Already Existing Title... 4.Non-Existing title	1. "Title less than 15 letters". 2."content less than 20 letters" 3.returns "{'Error', ['Title has already been used']}" 4.('Title created successfully', <post_id>)	As Expected	Successful
4	Updating a post	To update a ticket .. we must give Api_key and Api_username in header...	1. No input, 2.length(content)<20, 3.id not existing...4.id existing and content length more than 20	{'error': 'Failed to update ticket on Discourse'} for test cases 1,2, and 3... {'message': 'Ticket updated successfully'} for test case 4	As Expected	Successful
5	Deleting a Post	we must give Api_key and Api_username in header. ticket_id needed	1. ticket id doesn't exist... 2.ticket_id exists	1.'error': 'Failed to delete ticket on Discourse' 2. 'message': 'Ticket deleted successfully'}), 200	Always getting error	Failed

```
test_apis.py::test_get_post 40 --- Ticket not found on Discourse
PASSED
test_apis.py::test_get_posts 34 raw content string ecdkjifndoas cenrsinsdzxcie cnfixncdsd
PASSED
test_apis.py::test_create_post ('Error', ['Title has already been used'])
PASSED
test_apis.py::test_update_ticket {'error': 'Failed to update ticket on Discourse'}
PASSED
test_apis.py::test_delete_ticket 500
PASSED
```

Functions and Results

```
def test_get_posts(inventory):  
    response = get_all_tickets()  
    assert response.status_code == 200  
    data = response.json()['latest_posts']  
    for i in data:  
        print(i['id'], i['raw'])  
        print('\n')
```

```
@app.route('/get_all_tickets', methods=['GET'])  
def get_all_tickets():  
    headers = {  
        'api_key': DISCOURSE_API_KEY,  
        'api_username': DISCOURSE_API_USERNAME  
    }  
    response = requests.get(f'{DISCOURSE_API_BASE_URL}/posts.json')  
    return response
```

```
===== 1 passed in 2.90s =====  
(.env) astrisk@Anshs-MacBook-Air Backend % pytest -v -s  
===== test session starts =====  
platform darwin -- Python 3.9.6, pytest-8.1.1, pluggy-1.4.0 -- /Users/astrisk/Documents/iitm/soft-engg-project-jan-2024-se-jan-28/Milestone  
-6-Final-Submission/Backend/.env/bin/python3  
cachedir: .pytest_cache  
rootdir: /Users/astrisk/Documents/iitm/soft-engg-project-jan-2024-se-jan-28/Milestone-6-Final-Submission/Backend  
collected 1 item  
  
test_apis.py::test_get_posts 33 raw content string ecdkjifndoas cenrsinsdzcxcie cnfixncdsd  
  
32 raw content string ecdkjifndoas cenrsinsdzcxcie cnfixncdsd  
  
31 Put all your BDM doubts here  
  
30 raw is the content im putting  
  
24 repy it is yesn jvhv  
  
15 contentfedcbj evc ecabsdhbersvdc eracsbjes
```

Get all
posts

```
def test_get_post(inventory):
    id = 40
    response = get_ticket(id)
    if response[1] == 200:
        print(id, '---', response[0].json['raw'])
    elif response[1] == 404:
        print(id, '---', response[0].json['error'])
```

```
@app.route('/get_ticket/<int:topic_id>', methods=['GET'])
def get_ticket(topic_id):
    # Fetch topic details from Discourse
    discourse_response = requests.get(f'{DISCOURSE_API_BASE_URL}/posts/{topic_id}.json')
    if discourse_response.status_code == 200:
        topic_data = discourse_response.json()
        return jsonify(topic_data), 200
    else:
        return jsonify({'error': 'Ticket not found on Discourse'}), 404
```

```
(.env) astrisk@Anshs-MacBook-Air Backend % pytest -v -s
```

```
===== test session starts =====
platform darwin -- Python 3.9.6, pytest-8.1.1, pluggy-1.4.0 -- /Users/astrisk/Documents/iitm/soft-engg-project-2024-25-Milestone-6-Final-Submission/Backend/.env/bin/python3
cachedir: .pytest_cache
rootdir: /Users/astrisk/Documents/iitm/soft-engg-project-jan-2024-se-jan-28/Milestone-6-Final-Submission/Backend
collected 1 item
```

```
test_apis.py::test_get_post 30 --- raw is the content im putting
PASSED
```

```
===== 1 passed in 1.20s =====
(.env) astrisk@Anshs-MacBook-Air Backend % pytest -v -s
===== test session starts =====
platform darwin -- Python 3.9.6, pytest-8.1.1, pluggy-1.4.0 -- /Users/astrisk/Documents/iitm/soft-engg-project-2024-25-Milestone-6-Final-Submission/Backend/.env/bin/python3
cachedir: .pytest_cache
rootdir: /Users/astrisk/Documents/iitm/soft-engg-project-jan-2024-se-jan-28/Milestone-6-Final-Submission/Backend
collected 1 item
```

```
test_apis.py::test_get_post 40 --- Ticket not found on Discourse
PASSED
```

Get specific post


```
def test_create_post():
    result = create_ticket({"title": "title is the new title", "raw": "raw content string ecdkjifndoas cenrsinsdzxcie cnfixncds"})
    print(result)
```

```
@app.route('/create_ticket/<int:ticket_id>', methods=['POST'])
```

```
def create_ticket(data):
    title = data['title']
    raw = data['raw']
    if len(title)<15:
        return 'title should be greater than 15 letter', 422
    elif len(raw)<20:
        return 'body should be greater than 20 letter', 422
```

```
# Headers
headers = {
    'api_key': DISCOURSE_API_KEY,
    'api_username': DISCOURSE_API_USERNAME
}
```

```
# Create a new topic (ticket) on Discourse
discourse_payload = {
    'title': title,
    'raw': raw,
    "category": 4
}
```

```
response = requests.post(f'{DISCOURSE_API_BASE_URL}/posts.json', json=discourse_payload)
```

```
if response.status_code == 200:
    return 'Title created successfully', response.json()['id']
else:
    return 'Error', response.json()['errors']
```

```
platform darwin -- Python 3.9.6, pytest-8.1.1, pluggy-1.4.0 -- /Users/astrisk/anaconda3/bin/python3
--6-Final-Submission/Backend/.env/bin/python3
cachedir: .pytest_cache
rootdir: /Users/astrisk/Documents/iitm/soft-engg-project-jan-2024-se-jan-28/Milestone-6-Final-Submission/Backend
collected 1 item
```

```
test_apis.py::test_create_post ('Title created successfully', 34)
PASSED
```

```
===== 1 passed in 0.01s =====
(.env) astrisk@Anshs-MacBook-Air Backend % pytest -v -s
```

```
===== test session starts =====
platform darwin -- Python 3.9.6, pytest-8.1.1, pluggy-1.4.0 -- /Users/astrisk/anaconda3/bin/python3
--6-Final-Submission/Backend/.env/bin/python3
cachedir: .pytest_cache
rootdir: /Users/astrisk/Documents/iitm/soft-engg-project-jan-2024-se-jan-28/Milestone-6-Final-Submission/Backend
collected 1 item
```

```
test_apis.py::test_create_post ('Error', ['Title has already been used'])
PASSED
```

```
===== 1 passed in 1.30s =====
(.env) astrisk@Anshs-MacBook-Air Backend % pytest -v -s
```

```
===== test session starts =====
platform darwin -- Python 3.9.6, pytest-8.1.1, pluggy-1.4.0 -- /Users/astrisk/Documents/iitm/soft-engg-project-jan-2024-se-jan-28/Milestone-6-Final-Submission/Backend/.env/bin/python3
cachedir: .pytest_cache
rootdir: /Users/astrisk/Documents/iitm/soft-engg-project-jan-2024-se-jan-28/Milestone-6-Final-Submission/Backend
collected 1 item
```

```
test_apis.py::test_create_post ('title should be greater than 15 letter', 422)
PASSED
```

```
===== 1 passed in 1.20s =====
(.env) astrisk@Anshs-MacBook-Air Backend % pytest -v -s
```

```
===== test session starts =====
platform darwin -- Python 3.9.6, pytest-8.1.1, pluggy-1.4.0 -- /Users/astrisk/Documents/iitm/soft-engg-project-jan-2024-se-jan-28/Milestone-6-Final-Submission/Backend/.env/bin/python3
cachedir: .pytest_cache
rootdir: /Users/astrisk/Documents/iitm/soft-engg-project-jan-2024-se-jan-28/Milestone-6-Final-Submission/Backend
collected 1 item
```

```
test_apis.py::test_create_post ('body should be greater than 20 letter', 422)
PASSED
```

```
===== 1 passed in 0.96s =====
```

Create Ticket

```
def test_update_ticket():
    result = update_ticket({'id': 40, 'content': 'this is new raw content instead of old one'})
    print(result[0].json)
```

```
@app.route('/update_ticket/<int:ticket_id>', methods=['PUT'])
def update_ticket(data):
    ticket_id = data.get('id')
    content = data.get('content')

    # Headers
    headers = {
        'api_key': DISCOURSE_API_KEY,
        'api_username': DISCOURSE_API_USERNAME
    }

    # Construct the payload for updating the ticket on Discourse
    discourse_payload = {
        "post": {
            "raw": content
        }
    }

    # Make a PUT request to update the ticket on Discourse
    discourse_response = requests.put(f'{DISCOURSE_API_BASE_URL}/posts/{ticket_id}.json', json=discourse_payload, headers=headers)

    if discourse_response.status_code == 200:
        # Ticket updated successfully
        return jsonify({'message': 'Ticket updated successfully'}), 200
    else:
        # Failed to update ticket on Discourse
        return jsonify({'error': 'Failed to update ticket on Discourse'}), 500
```

```
on starts =====
astrisk/Documents/iitm/soft-engg-project-jan-2024-

rootdir: /Users/astrisk/Documents/iitm/soft-engg-project-jan-2024-se-jan-28/Milestone-6-Final-Submission/Backend
collected 1 item

test_apis.py::test_update_ticket {'message': 'Ticket updated successfully'}
PASSED

===== 1 passed in 2.66s =====
(.env) astrisk@Anshs-MacBook-Air Backend % pytest -v -s
platform darwin -- Python 3.9.6, pytest-8.1.1, pluggy-1.4.0 -- /Users/astrisk/Documents/iitm/soft-engg-project-jan-2024-6-Final-Submission/Backend/.env/bin/python3
cachedir: .pytest_cache
rootdir: /Users/astrisk/Documents/iitm/soft-engg-project-jan-2024-se-jan-28/Milestone-6-Final-Submission/Backend
collected 1 item

test_apis.py::test_update_ticket {'error': 'Failed to update ticket on Discourse'}
PASSED

===== 1 passed in 1.26s =====
```

Update Post

Delete Post

```
def test_delete_ticket():  
    result = delete_ticket(31)  
    # assert (result[1].status_code) == 200  
    print(result[1])
```

```
(.env) astrisk@Anshs-MacBook-Air backend % pytest -v -s  
===== test session start =====  
platform darwin -- Python 3.9.6, pytest-8.1.1, pluggy-1.4.0 -- /Users/astrisk/Development/Backend/.env/bin/python3  
cachedir: .pytest_cache  
rootdir: /Users/astrisk/Documents/iitm/soft-engg-project-jan-2024-se-jan-28  
collected 1 item  
  
test_apis.py::test_delete_ticket FAILED
```

```
@app.route('/delete_ticket/<int:ticket_id>', methods=['DELETE'])  
def delete_ticket(ticket_id):  
    # Headers  
    headers = {  
        'api_key': DISCOURSE_API_KEY,  
        'api_username': DISCOURSE_API_USERNAME,  
    }  
  
    # Make a DELETE request to delete the ticket on Discourse  
    discourse_response = requests.delete(f'{DISCOURSE_API_BASE_URL}/t/{ticket_id}.json', headers=headers)  
  
    if discourse_response.status_code == 200:  
        # Ticket deleted successfully  
        return jsonify({'message': 'Ticket deleted successfully'}), 200  
    else:  
        # Failed to delete ticket on Discourse  
        return jsonify({'error': 'Failed to delete ticket on Discourse'}), 500
```