

Quiz 1 Marking Guide

Test cases might use different DBs. In the list below, DB specifies the DB that is used for testing the case. You need to copy it to your folder and rename it as QuizDatabase.sqlite. The DBs can be downloaded at https://canvas.auckland.ac.nz/files/11595613/download?download_frd=1.

For each test case, the URL and the data used for the case are shown below.

If you think you have been marked wrongly for a test case, please check the test case as below:

1. Start your Quiz programs (these must be the files that you have submitted)
2. Invoke the endpoint through the swagger interface.
 - a. In the swagger interface, check whether the URL used for accessing the endpoint is the same as the one given for the test case in your report.
 - b. Check whether the expected response and the expected header (if any) given in the marking report match the ones shown in the swagger interface.
 - c. If the answer to 2a and 2b are both yes, you should contact the tutor to check the case/cases that you believe have been marked wrongly. When contacting the lecturer, please clearly indicate the test case/cases that you believe has/have been marked wrongly.

NOTE: Experience show that many students request remarking without carefully checking the test cases. To reduce the amount of time being wasted, we apply a penalty to people who do not carefully check the test cases. The penalty scheme is as below:

1. If any of the test cases is marked wrongly, no penalty.
2. If a student requests remarking one test case and the test case has been marked correctly, 2% off the total mark of the quiz.
3. If a student requests remarking more than one test cases and none of the test cases have been marked wrongly, 4% off the total mark of the quiz.

Test Cases

Case GetVersion

URL: <https://localhost:8080/api/GetVersion>

DB: QuizDatabase.sqlite

Mark for this case: 0.500

Expected result is the string specified for Endpoint 1, e.g., V26672221QZ

Case GetAllPosts1 (multiple posts)

URL: <https://localhost:8080/api/GetAllPosts>

DB: QuizDatabase.sqlite

Mark for this case: 2.5

Expected result:

```
[
  {
    "id": 10,
    "content": "test 1",
    "writer": "u1",
    "time": "2023-07-24 19:11:40",
    "likes": "",
    "idOfRepliedPost": 0
  },
  {
    "id": 11,
    "content": "test 2",
    "writer": "u2",
    "time": "2023-07-24 19:12:27",
    "likes": "u1",
    "idOfRepliedPost": 0
  },
  {
    "id": 12,
    "content": "test 3",
    "writer": "u3",
    "time": "2023-07-24 19:12:36",
    "likes": "u1,u2,u4",
    "idOfRepliedPost": 0
  },
  {
    "id": 13,
    "content": "test 4",
    "writer": "u4",
    "time": "2023-07-24 19:12:40",
    "likes": "",
    "idOfRepliedPost": 0
  },
  {
    "id": 14,
    "content": "test 5",
    "writer": "u1",
    "time": "2023-07-24 19:12:43",
    "likes": "",
    "idOfRepliedPost": 0
  }
]
```

Case GetAllPosts2 (no post)

URL: <https://localhost:8080/api/GetAllPosts>

DB: GetAllPosts2.sqlite

Mark for this case: 2.5

Expected result:

```
[[]]
```

Case GetAllPosts3 (one post)

URL: <https://localhost:8080/api/GetAllPosts>

DB: GetAllPosts3.sqlite

Mark for this case: 2.5

Expected result:

```
[
  {
    "id": 10,
    "content": "test 1",
    "writer": "u1",
    "time": "2023-07-24 19:11:40",
    "likes": "",
    "idOfRepliedPost": 0
  }
]
```

Case GetPostById 1 (no likes)

URL: <https://localhost:8080/api/GetPostById/10>

DB: QuizDatabase.sqlite

Mark for this case: 2

Expected result:

```
{
  "id": 10,
  "content": "test 1",
  "writer": "u1",
  "time": "2023-07-24 19:11:40",
  "likesCount": 0,
  "idOfRepliedPost": 0
}
```

Case GetPostById 2 (one like)

URL: <https://localhost:8080/api/GetPostById/11>

DB: QuizDatabase.sqlite

Mark for this case: 2

Expected result:

```
{
  "id": 11,
  "content": "test 2",
  "writer": "u2",
  "time": "2023-07-24 19:12:27",
  "likesCount": 1,
  "idOfRepliedPost": 0
}
```

Case GetPostById 3 (three likes)

URL: <https://localhost:8080/api/GetPostById/12>

DB: QuizDatabase.sqlite

Mark for this case: 2

Expected result:

```
{
  "id": 12,
  "content": "test 3",
  "writer": "u3",
  "time": "2023-07-24 19:12:36",
  "likesCount": 3,
  "idOfRepliedPost": 0
}
```

Case GetPostById 4 (post does not exist)

URL: <https://localhost:8080/api/GetPostById/1>

DB: QuizDatabase.sqlite

Mark for this case: 2.000

Response code: 400

Expected result:

```
Post with Id 1 does not exist.
```

Case GetFollowing 1 (no following)

URL: <https://localhost:8080/api/GetFollowing/u1>

DB: QuizDatabase.sqlite

Mark for this case: 1.250

Expected result:

```
[ ]
```

Case GetFollowing 2 (one following)

URL: <https://localhost:8080/api/GetFollowing/u2>

DB: QuizDatabase.sqlite

Mark for this case: 1.250

Expected result:

```
[  
  "u1"  
]
```

Case GetFollowing 3 (many followings)

URL: <https://localhost:8080/api/GetFollowing/u4>

DB: QuizDatabase.sqlite

Mark for this case: 1.250

Expected result:

```
[  
  "u1",  
  "u2",  
  "u3"  
]
```

Case GetFollowing 4 (user does not exist)

URL: <https://localhost:8080/api/GetFollowing/u7>

DB: QuizDatabase.sqlite

Mark for this case: 1.250

Response code: 400

Expected result:

```
UserName u7 does not exist.
```

Case GetFollower 1 (no follower)

URL: <https://localhost:8080/api/GetFollower/u4>

DB: QuizDatabase.sqlite

Mark for this case: 1.000

Expected result:

```
[]
```

Case GetFollower 2 (one follower)

URL: <https://localhost:8080/api/GetFollower/u3>

DB: QuizDatabase.sqlite

Mark for this case: 1.000

Expected result:

```
[  
  "u4"  
]
```

Case GetFollower 3 (many followers)

URL: <https://localhost:8080/api/GetFollower/u1>

DB: QuizDatabase.sqlite

Mark for this case: 1.000

Expected result:

```
[  
  "u2",  
  "u3",  
  "u4"  
]
```

Case GetFollower 4 (user does not exist)

URL: <https://localhost:8080/api/GetFollower/u7>

DB: QuizDatabase.sqlite

Mark for this case: 2.000

Response code: 400

Expected result:

```
UserName u7 does not exist.
```

Case SuspendUser 1 (no authentication header):

URL: https://localhost:8080/api/SuspendUser

DB: QuizDatabase.sqlite

Mark for this case: 1.000

Expected result:

If you don't provide username and password, you should keep on getting the log in window.

Case SuspendUser 2 (normal case):

URL: https://localhost:8080/api/SuspendUser

DB: QuizDatabase.sqlite

Login information

Username: a1

Password: a1pwd

Suspended user information:

username: u4

reason: no reason

Mark for this case: 2.000

Expected result:

Success

Case SuspendUser 3 (already suspended):

URL: <https://localhost:8080/api/SuspendUser>

DB: QuizDatabase.sqlite

Login information

Username: a1

Password: a1pwd

Suspended user information:

username: u5

reason: good reason

Mark for this case: 1.000

Expected result:

The user has already been suspended.

Case SuspendUser 4 (suspended user does not exist):

URL: <https://localhost:8080/api/SuspendUser>

DB: QuizDatabase.sqlite

Login information

Username: a1

Password: a1pwd

Suspended user information:

username: u6

reason: whatever

Mark for this case: 1.000

Expected response code: 400

Expected result:

UserName u6 does not exist.

Case PostMessage 1 (no authentication header):

URL: <https://localhost:8080/api/PostMessage>

DB: QuizDatabase.sqlite

Mark for this case: 0.1

Expected result:

If you don't provide username and password, you should keep on getting the log in window.

Case PostMessage 2 (normal case):

URL: <https://localhost:8080/api/PostMessage>

DB: QuizDatabase.sqlite

Login information:

username = "u4";

password = "u4pw";

Post information:

idOfRepliedPost: 12

message: test content

Mark for this case: 2.100

Expected result:

```
{
  "id": 22,
  "content": "test content",
  "writer": "u4",
  "time": "2023-08-06 17:06:59",
  "likesCount": 0,
  "idOfRepliedPost": 12
}
```

Expected Location header: `location: https://localhost:8080/api/GetPostById/22`

The 'Posts' table in DB is updated.

NOTE:

- The value of id might not be 22. As long as the id matched the id in the Location, it is correct.
- The value of time does not need to be same as shown. But it must be in format yyyy-MM-dd HH:mm:ss

Case PostMessage 3 (IdOfRepliedPost does not exist):

URL: <https://localhost:8080/api/PostMessage>

DB: QuizDatabase.sqlite

Login information:

username = "u4";

password = "u4pw";

Post information:

idOfRepliedPost: 1

message: test content

Mark for this case: 0.600

Expected response code: 400

Expected result:

Post 1 does not exist.

Case PostMessage 4 (suspended user posts):

URL: <https://localhost:8080/api/PostMessage>

DB: QuizDatabase.sqlite

Login information:

username = "u5";

password = "u5pw";

Post information:

idOfRepliedPost: 1

message: test content

Mark for this case: 0.600

Expected response code: 400

Expected result:

You cannot carry out **this** operation **as** your account is suspended.

Case PostMessage 5 (incorrect credential):

URL: `https://localhost:8080/api/PostMessage`

DB: `QuizDatabase.sqlite`

Login information:

`username = "u6";`

`password = "u5pw";`

Post information:

`idOfRepliedPost: 1`

`message: test content`

Mark for this case: 0.050

Click the "Cancel" button of the login window

Expected response code: 401

Expected result:

`userName and password do not match`

Case PostMessage 6 (forbidden):

URL: <https://localhost:8080/api/PostMessage>

DB: QuizDatabase.sqlite

Login information:

username = "a1";

password = "a1pwd";

Post information:

idOfRepliedPost: 1

message: test content

Mark for this case: 0.050

Expected response code: 403

Case GetRecommendedUsers 1 (no authentication header):

URL: <https://localhost:8080/api/GetRecommendedUsers>

DB: QuizDatabase.sqlite

Mark for this case: 0.05

Expected result:

If you don't provide username and password, you should keep on getting the log in window.

Case GetRecommendedUsers 2 (empty list):

URL: <https://localhost:8080/api/GetRecommendedUsers>

DB: QuizDatabase.sqlite

Login information:

username = "u4";

password = "u4pw";

Mark for this case: 0.250

Expected result:

```
[ ]
```

Case GetRecommendedUsers 3 (two recommended users):

URL: <https://localhost:8080/api/GetRecommendedUsers>

DB: GetRecommendedUsers3.sqlite

Login information:

username = "u4";

password = "u4pw";

Mark for this case: 0.250

Expected result:

```
[  
  "u6",  
  "u5"  
]
```

Case GetRecommendedUsers 4 (two recommended users):

URL: <https://localhost:8080/api/GetRecommendedUsers>

DB: GetRecommendedUsers4.sqlite

Login information:

username = "u4";

password = "u4pw";

Mark for this case: 0.100

Expected result:

```
[  
  "u6",  
  "u5"  
]
```

Case GetRecommendedUsers 5 (suspended user):

URL: <https://localhost:8080/api/GetRecommendedUsers>

DB: GetRecommendedUsers4.sqlite

Login information:

username = "u5";

password = "u5pw";

Mark for this case: 0.350

Expected response code: 400

Expected result:

```
You cannot carry out this operation as your account is suspended.
```