

## Test case: Creating a new task in postman with valid data

### Test steps:

1. Start the program, MySQL and Postman.
2. In Postman, make sure you are using POST.
3. Enter the correct URL, in this case: <http://localhost:8080/tasks>
4. In BODY enter the Title of the task followed by the description and ending with the status in that order. Make sure you are using the correct syntax.
5. In HEADER, use the key CONTENT-TYPE and value JSON.
6. Click SEND.

### Test data:

Title: title1    Description: description1    Status: STARTED

### Expected results:

A new task should be created in the database.

### Actual results:

As Expected

### Pass/Fail:

Pass

## Test case: Creating a new task in postman with invalid data

### Test steps:

1. Start the program, MySQL and Postman.
2. In Postman, make sure you are using POST.
3. Enter the correct URL, in this case: <http://localhost:8080/tasks>
4. In BODY enter the Title of the task followed by the description and ending with the status in that order. Make sure you are using the correct syntax.
5. In HEADER, use the key CONTENT-TYPE and value JSON.
6. Click SEND.

### Test data:

Title: title2   Description: description2   Status: TEST

### Expected results:

An error message pointing out that STATUS was incorrectly typed.

### Actual results:

As Expected

### Pass/Fail:

Pass

## Test case: Skapa en ny user i postman med valid data

### Test steps:

1. Starta applikationen, MySQL och Postman.
2. I Postman, tryck på fliken för POST.
3. Skriv in den korrekta URL:en,. (http://127.0.0.1:8080/users)
4. Skriv in samtliga attribut för en user i BODY. Viktigt att syntaxen är korrekt.
5. Ange JSON som value till key CONTENT-TYPE i headern.
6. Klicka på SEND.

### Test data:

```
{  
  "firstName":"Henrik",  
  "lastName":"Badboll",  
  "userName":"Wiihooo321",  
  "active": true,  
  "userNumber": 1234  
}
```

### Expected results:

Status: 201 Created

En ny user är skapad i databasen.

Actual results:

As Expected

Pass/Fail:

Pass

Test case: Skapa en ny user i postman med invalid data

Test steps:

1. Starta applikationen, MySQL och Postman.
2. I Postman, tryck på fliken för POST.
3. Skriv in den korrekta URL: `http://127.0.0.1:8080/users`
4. Skriv in samtliga attribut för en user i BODY. Viktigt att syntaxen är korrekt.
5. Ange JSON som value till key CONTENT-TYPE i headern.
6. Klicka på SEND.

Test data:

```
{  
  "firstName": "Henrik",  
  "lastName": "Badboll",  
  "userName": "Wii",  
  "active": true,  
  "userNumber": 1235  
}
```

Expected results:

Status: 400 Bad Request.

Samt ett felmeddelande som säger att UserName måste vara minst 10 tecken.

Actual results:

As expected

Pass/Fail:

Pass

Test case: Uppdatera en user med valid data.

Test steps:

1. Kör programmet, MySQL och Postman.
2. Klicka på fliken PUT i Postman.
3. Skriver in korrekt URL i adressfältet <http://127.0.0.1:8080/users/{id}>,  
id är userId.
4. I BODY skriver vi alla attribut som finns för vald User men ändrar userName.
5. I HEADER, key = CONTENT-TYPE och value = JSON.
6. Klicka på SEND.

Test data:

```
{  
  "firstName": "Kalle",  
  "lastName": "Kofot",  
  "userName": "CarlDenFemte",  
  "active": true,  
  "userNumber": 1234  
}
```

Expected results:

Status 200 OK

Actual results:

As Expected

Pass/Fail:

Pass

Test case: Uppdatera en user med invalid data.

Test steps:

1. Kör programmet, MySQL och Postman.
2. Klicka på fliken PUT i Postman.
3. Skriver in korrekt URL i adressfältet <http://127.0.0.1:8080/users/{id}>,

id är userId.

4. I BODY skriver vi alla attribut för vald user men skriver ett avsiktligt fel som värde till active,
5. I HEADER, key = CONTENT-TYPE och value = JSON.
6. Klicka på SEND.

Test data:

```
{  
  "firstName": "Kalle",  
  "lastName": "Kofot",  
  "userName": "CarlDenFemte",  
  "active": Det är sant,  
  "userNumber": 1234  
}
```

Expected results:

Status: 400 Bad Request

Actual results:

As Expected

Pass/Fail:

Pass

Test case: Inaktivera en user i postman med valid data

Test steps:

1. Kör programmet, MySQL and Postman.
2. I Postman, välj PUT.
3. skriv in korrekt URL i adressfältet <http://127.0.0.1:8080/users/{id}/deactivate>
4. I BODY skriv {id} vilken user vill du ändra.
5. I HEADER, use the key CONTENT-TYPE and value JSON.
6. Click SEND.

Test data:

Krävs ingen.

### Expected results:

En user har Inaktiverat i database.

### Actual results:

As expected

### Pass/Fail:

Pass

### Test case: Inaktivera en user i postman med Invalid data

#### Test steps:

1. Kör programmet, MySQL and Postman.
2. I Postman, välja PUT.
3. skriver in korrekt URL i adressfältet <http://127.0.0.1:8080/users/{id}/deactivate>
4. I BODY skriv {id} vilken user vill du ändra.
5. I HEADER, use the key CONTENT-TYPE och value JSON.
6. Klickar på SEND.

#### Test data:

```
{  
  id: 22  
}
```

### Expected results:

An error Message som säger Status: 404 Not Found.

### Actual results:

As expected

### Pass/Fail:

Pass

### Test case: Hämta alla users som ingår i ett specifikt team med valid data

### Test steps:

1. Kör programmet, MySQL och Postman.
2. I Postman, välj fliken Get.
3. Skriver in korrekt URL i adressfältet <http://127.0.0.1:8080/users/getByTeamId/{Id}>, Id är teamId
4. Klicka på SEND.

### Test data:

<http://127.0.0.1:8080/users/getByTeamId/3>

### Expected results:

Användaren bör få alla Users som ingår i samma team.

### Actual results:

As expected

### Pass/Fail:

Pass

### Test case: Hämta alla users som ingår i ett specifikt team med invalid data

### Test steps:

1. Kör programmet, MySQL and Postman.
2. I Postman, välj Get.
3. Skriver in URL i adressfältet <http://127.0.0.1:8080/users/getByTeamId/{Id}> med fel TeamID.
4. Klickar på SEND.

### Test data:

<http://127.0.0.1:8080/users/getByTeamId/6>

### Expected results:

Felmeddelande: "error": "Could not find any user"

Actual results:

As expected

Pass/Fail:

Pass

Känt fel: Uppdatera en user som finns i databasen utan att ange samtliga user variabler.

Test steps:

1. Kör programmet, MySQL and Postman.
2. I Postman, välj PUT.
3. skriv in korrekt URL i adressfältet <http://127.0.0.1:8080/users/{id}> , id är userId
4. I BODY ska användaren skriva bara ett attribut som användaren vill uppdatera.
5. I HEADER, use the key CONTENT-TYPE and value JSON.
6. Klicka på SEND.

Test data:

```
{  
  "active": false  
}
```

Expected results:

Status 500 Internal Server Error

Actual results:

As expected

Pass/Fail:

Pass