

Use case: skapa en user

Actor: användare

Scope: Ett todo-system

Användaren skickar en request till systemet med information gällande kontot som användaren vill skapa.

Preconditions:

Användaren måste kunna kalla på systemets API.

??Användaren måste skriva in ett acceptabelt användarnamn och lösenord.

Postconditions:

*Success guarantees

Användarens uppgifter uppfyller alla krav gällande vilka tecken som får användas, att de fält som måste vara unika är unika etc.

Sedan sparas användaren som medlem i databasen, och blir returnerad ett meddelande om att processen gick igenom.

Flöde:

Användaren matar in önskat användarnamn och lösenord.

Användaren skickar en request till systemet.

Systemet kollar ifall fälten som användaren matade in är godkända.

Systemet returnerar till användaren att en medlem har blivit tillagd i databasen.

Felflöde

1. Användaren matar in önskat användarnamn och lösenord.
2. Användaren anger ett användarnamn som redan är i bruk.
3. Användaren skickar en request till systemet.
4. Systemet säger ifrån och skickar en error till användaren om att användarnamnet är upptaget.

Testfall

- Gå till postman
- Gör en post request med inparametrar
username: "Olle37"
firstname: "Olle"
lastname: "Ollesson"
- Ange användarnamn och lösenord
- Klicka på kör

Förväntade resultat:

CREATED

Ifall någonting gick fel:
Fel input

Use case: Lägg till users som "helpers" på en task som är startad.

Actor: användare

Scope: Ett todo-system

Användaren skickar en request till systemet med användarnumret till användaren som ska läggas till i tasken.

Alternativt: (felutfall): Ingen läggs till som helper.

Informationstext visas.

Preconditions:

Användaren måste kunna kalla på systemets API.

Användarnumret som matas in får inte vara detsamma som det som tillhör den som äger tasken.

Användarnumret får inte vara detsamma som användarnumret till någon utav de andra "helpers" i en task.

Trigger: Användaren skickar en request.

Postconditions:

Flöde:

1. Användaren matar in önskat användarnummer på den medlem han vill ange som helper.
2. Användaren skickar en request till systemet.
3. Systemet jämför användarnumret gentemot de andra användarna som är med i tasken.
4. Systemet lägger till användaren som "helper" i tasken.
5. Systemet returnerar till användaren att medlemmen har blivit tillagd som "helper".

Felflöde

1. Användaren matar in ett användarnummer som redan är i bruk i tasken.
2. En request skickas till systemet om att lägga till en helper.
3. Systemet ser att användarnumret redan är i bruk i tasken.
4. Systemet returnerar ett felmeddelande till användaren.

Testfall T01

Lägg till användare som helper för en task, lyckad request

Förutsättningar: Kan skicka request till API.

Steg för genomförande: (Lyckad)

1. Mata in användarnummer för användaren du vill lägga till, se till att användarnumret inte tillhör tasken.
2. Gör ett put anrop i postman.
3. Tryck ok.

Steg för genomförande(Misslyckat pga fel användarnummer)

1. Mata in användarnummer för användaren du vill lägga till, skriv ett nummer som redan är i bruk av någon som är med i tasken.
2. Gör ett put anrop i postman.
3. Tryck ok.

Förväntat utfall: Felmeddelande och ingen person tillagd som helper i tasken.

```
User user = new User("myFirstname", "myLastname", "MyUsernameee", true, 112233L);
```

```
@Before
```

```
public void setUp() {  
    userRepository.save(user);  
}
```

```
@Test
```

```
public void createUserTest() throws Exception {  
    List<User> result = userRepository.findByUserNumber(112233L);  
    assertEquals(!result.isEmpty(), !result.isEmpty());  
}
```

```
@After
```

```
public void tearDown() {  
    userRepository.delete(user);  
}
```