

Test ID: 1

Requirement Being Tested:

We are testing if the user data(first name, last name, password) matches what is stored in the database

Component Being Tested:

check_user_data(self, user, data, orig_user_data=None)

Test inputs:

jettiejr@g.cofc.edu, josh, jettie

Expected outcomes:

True, True, True

because it is asserting true if the email, first and last name all match the provided data in the database.

Path to file:

unisubs/apps/api/tests/test_users.py

Test ID: 2

Requirement Being Tested:

instantiates a blank user. A user with no information.

Component Being Tested:

test_create_user_blank_data(self):

Test inputs:

There is not really inputs, we just dont input any information

Expected outcomes:

```
'username': 'test-user',  
  'email': 'test@example.com',  
  'first_name': '',  
  'last_name': '',  
  'full_name': '',  
  'bio': '',  
  'homepage': ''
```

By not entering in information we are expecting that all the data to be empty strings.

Path to file:

unisubs/apps/api/tests/test_users.py

Test ID: 3

Requirement Being Tested:

We are testing to see if the function `user_can_edit_subtitles()` gives the correct output of false when we give it a user who is not staff

Component Being Tested:

`user_can_edit_subtitles(user, video, language_code)`

Test inputs:

User object who is not staff and for whom the `user_can_edit_subtitles()` function for workflow objects returns false, a video with the url:

<https://www.youtube.com/watch?v=dQw4w9WgXcQ>, en

Expected Outcomes:

False, as the supplied user should not have permission to edit subtitles.

Path to File:

apps/subtitles/permissions.py

Test ID: 4

Requirement Being Tested:

updating the comments section when given a POST request

Component Being Tested:

`update_comments(request)`

Test inputs:

A Django request object with a POST command for a comment object

Expected Outcomes:

The displayed view should contain the new comment object at the bottom of the screen.

Path to File:

apps/comments/views.py

Test ID: 4

Requirement Being Tested:

updating the comments section when given a POST request

Component Being Tested:

`update_comments(request)`

Test inputs:

A Django request object with a POST command for a comment object

Expected Outcomes:

The displayed view should contain the new comment object at the bottom of the screen.

Path to File:

`apps/comments/views.py`

Test ID: 5

Requirement Being Tested:

We are testing if we can create a video from a specific URL

Component Being Tested:

`test_create_videos(self):`

Test inputs:

`"url": "http://www.example.com/sdf.mp4", "langs"`

Expected outcomes:

True which means the video was created and ran from `testhelpers.views`

Path to file:

`unisubs/apps/testhelpers/tests.py`