ASSIGNMENT # 5

About the assignment

In the class we saw Page Object pattern and used it to develop test cases for wordpress.com site. We also saw how to upload files and switch between frames / windows / iframes.

We will be extending the suite of test cases provided for wordpress by either adding more functionality or adding new tests.

Prerequisite

Because we will be testing wordpress site (specifically the administrative area of a wordpress site), we will need a installation of wordpress that can be tested.

The easiest one is to go to http://www.wordpress.com and sign up for one. Within few minutes a wordpress installation that you can test will be ready.

Since we will be testing administrator area, the starting point of your test will be login of the administrator area.

In my case,

my wordpress blog: https://jatin146.wordpress.com/
my admin area: https://jatin146.wordpress.com/wp-admin

If you already have a wordpress account, it is still a good idea to create a new account (using some other email address) for this assignment since it will be isolated and you can develop tests without messing with existing blog.

What is already provided:

I have provided a starter code and one test which does the following:

- Log into wordpress blog and go to dashboard
- Navigate to post dashboard (where all the posts are listed)
- Click on the “Add New” to open a page which allows you to create a new post
- Enter some title (leave content blank) and publish the post

Note about code changes:

This is a fairly big assignment, please feel free to make changes to any or all part of the code as you see fir. There is no set structure that you need to follow. If you feel the current code organization rigid, feel free to delete it. Add new class as you wish. Change existing code to fit your need.

How to start

Just like all the previous assignments, you will be downloading the assignment5.zip attached along with this write-up.

All the code other than the actual test is in the following package:
edu.ucsc.extension.wtest.support

The tests are in the following package:
edu.ucsc.extension.wtest.seleniumtests (Specifically class called Tests.java)

You will be adding / updating existing test in this file (Tests.java)

What to turn in

Make sure that when you run “gradle test” all the tests are executed one after the other.

Please update the included readme.txt file to give me an overview of the code organization, what is working well and what still needs more work, what was challenging.

Zip the entire folder that you downloaded (after adding more code, changing existing code)

NOW LETS SEE WHAT TESTS WE SHOULD BE WRITING

Test # 1: Modification to the existing test

There is only one existing test (the described above , which creates a post without content)

The signature of the method which publishes the post is as follows:

```
void publish(String title)
```

We will be changing the signature to the following:

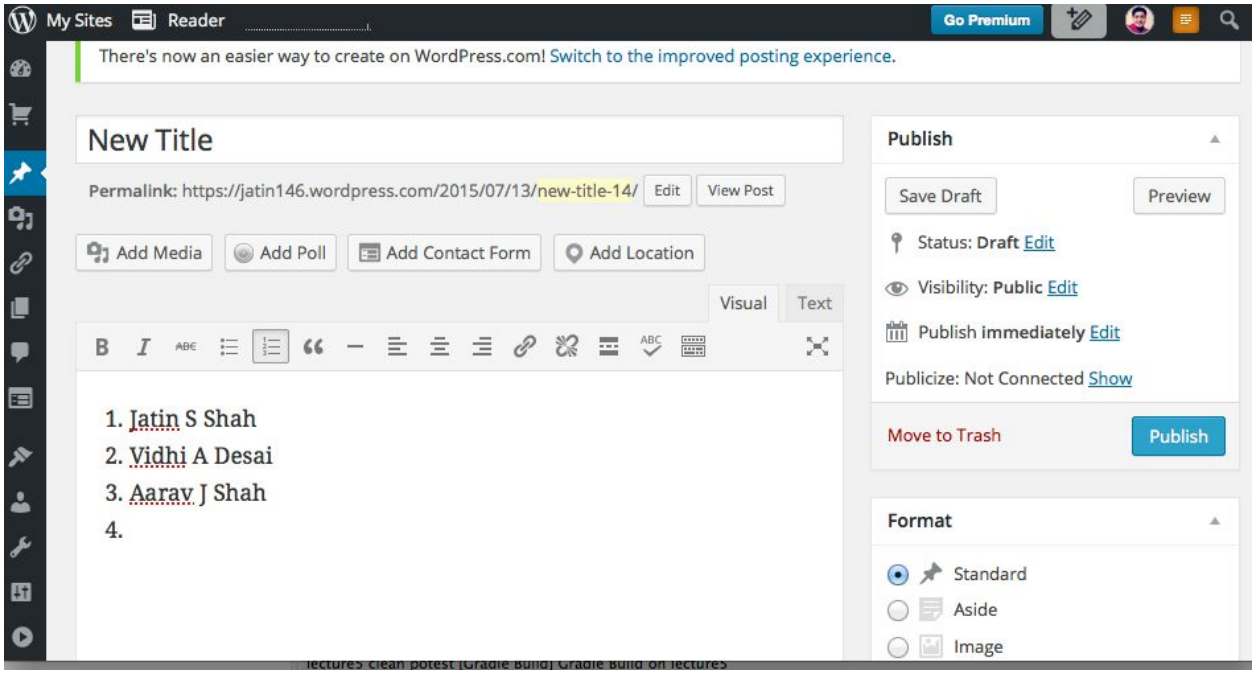
```
int publish(String title, List<String> contentBullets)
```

Where contentBuller is the list of string which will be used as the bullet points in the content.

So if contentBullet is a list of string with following three strings:

- Jatin S Shah
- Vidhi A Desai
- Aarav J Shah

Your code that publishes the post should fill in both title as well as content as shown in the screenshot:



The method should return the id of the post that it just created (will be very useful in other tests)

Note: Writing to content area is not as straightforward as getting a text area and sending keys to it. This is a JavaScript based tinyMCE rich editor which is inside an iframe.

The above change may get tricky, I will be interested in finding out what all ideas you guys come up with and we can discuss them on the forums.

Also add the verification in the test which makes sure that the post count increases by 1 after the post is created.

Test # 2 Fixing getPostCount function

There is a bug in this function which returns a incorrect number of posts when there are no posts, please fix that.

Test # 3: Delete a post by ID

Provide a function with the following signature (in Post Dashboard page)

```
public PostDashboard deletePostById(int postId)
```

This function deletes the post by an ID. Use the following two functions:

- publish
- deletePostById

to develop two tests which does the following:

- Create a post and then delete it
- Create three posts and then delete all of them (one after the other)

Test # 4: Delete All Posts

Write a function in PostDashboard with the following signature:

```
public PostDashboard deleteAllPosts()
```

This function deletes all the posts on the page (or none if there are no posts)

Use this to develop a test which does the following:

- Create 5 posts and delete all of them in single method call

Test # 5

In the posts dashboard area, there is a way to quick edit a post (which allows a user to quickly edit the post's title)

Develop a test which tests this functionality

- Create a post
- Edit its title
- Delete the post

Test # 6

Wordpress has a concept of media library. If you click on the "Media" link on the Dashboard it provides two options

- Library (Which is your media library)
- Add New (Allows you to upload a new file)

Lets write a test which does the following:

- Upload 3 photos (Go to Add New Media page)
- Delete them after they are uploaded (Go to Media Library)

Note about the uploader:
The workpress uses a JavaScript based multi file uploader (AJAX uploader) and does not associate the input HTML element of type file with the file upload button.

You may need to click on the "browser uploader" to activate the old style browser uploader.

Then you can use the Uploader to upload the file.

Every time you upload a file, you will be taken to the media page. You may need to navigate between "library" and "add new media" to accomplish uploading 3 images.