

Welcome to E Commerce website Testing

Set Up The Framework:

1.Important things:

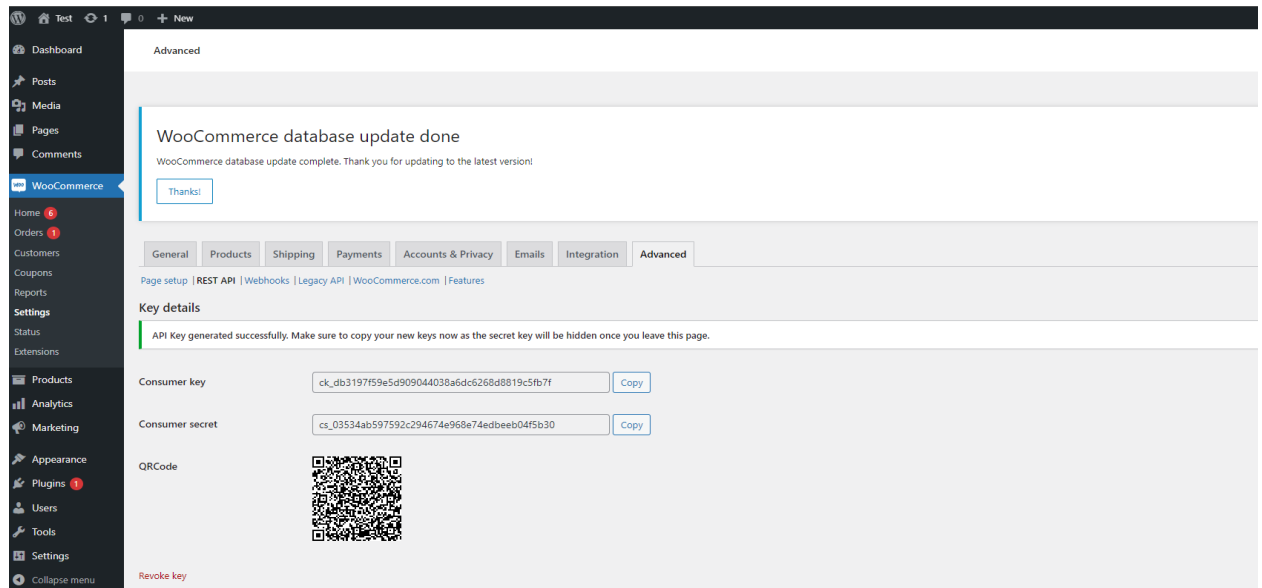
- The project supports only chrome.
- The project supports only WordPress sites.
- You must install WooCommerce in your WordPress through the plugins.
- Check your browser version for the web driver. (Version 107 included in the project)
- If your using Local to run WordPress locally you will need to fill the port option.

Must do in WordPress :

- 1) Download the zip file from [here](#) if u run WordPress locally.
- 2) Pull the sample_products.xml from the zip then Go back to WordPress: Tools -> WordPress -> install -> run Importer -> choose file -> provide the location of the XML file.
- 3) Settings -> reading -> click on A static page ->change homepage to shop
- 4) Settings -> general -> check Anyone can register.
- 5) WooCommerce -> settings -> Accounts & Privacy -> “Guest checkout” check all, “Account Creation” check all but the last one.
- 6) Must create coupon code: marketing -> coupon (coupon type percentage discount)

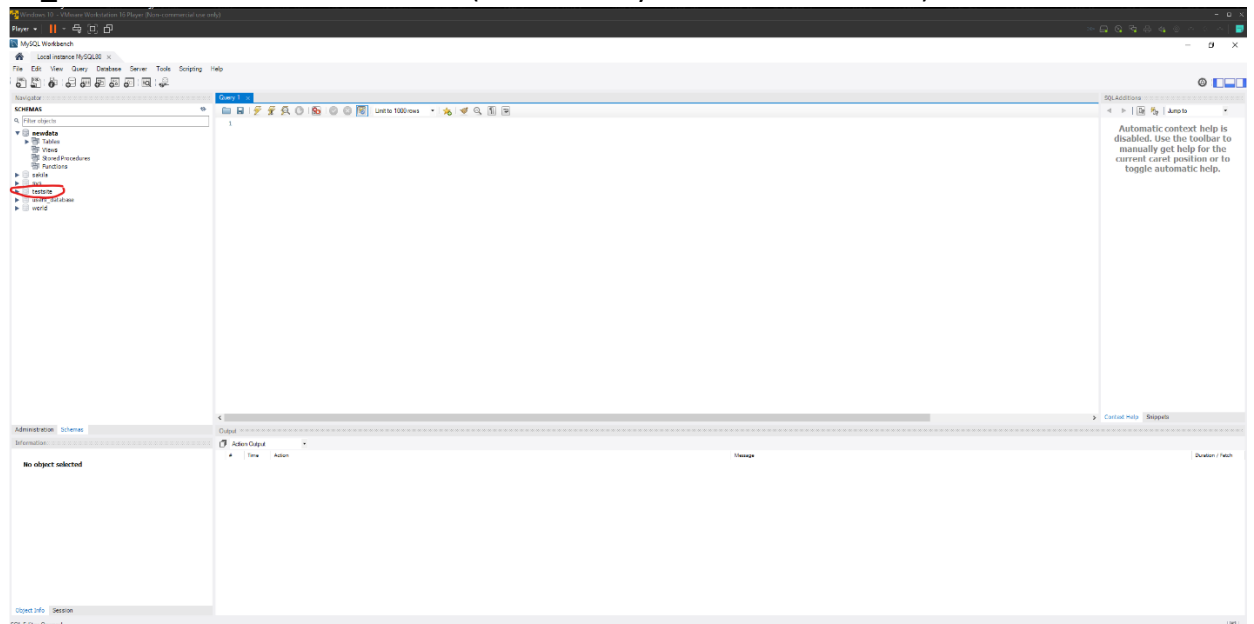
2.set up the Credantials.ps1:

- WC_KEY – WordPress public key
- WC_SECRET – WordPress secret key



*NOTE: im using MYSQL.

- DB_USER – Database Login Name (default root)
- DB_PASSWORD - Database Password (default root)
- DB_SERVER – Database Host(default localhost, if you're using VMware for DB you will need to provide the IP of the iso.)
- SB_NAME – Database Table Name. (same DB as you're WordPress site)



- SELPATH – Selenium Driver Path. ([click here](#) or you can set up manually)
- ENVSEL – The URL of the site.
- COUPON – the name of your coupon.

- PORT = the port you're using to run WordPress Database. (optional – for local users)


3. Set up The Testing Environment:

1) Create python virtual environment:

1.1) pip install virtualenv

1.2) py -m venv {Name for your virtual environment.}

1.3) open IDE make the virtual environment as main virtual environment.

*)PyCharm – file -> settings -> Project -> python interpreter ->  -> add -> new environment -> click on Location -> paste you're python.exe path

1.4) activate your virtual environment.

1.5) install/develop setup.py, command: python setup.py install.

4. Set Up The Selenium Path:

1) copy the chrome driver.exe to the “front” folder.

1.1) execute the getpath.py.

1.2) copy the path you got in the terminal to **Credantials.ps1** (SELPATH=” whatever path you have copied”).

5. Testing Section:

BackEnd Section:

Commands:

- tcid stands for – test case id.
- tcidc stands for – test case id for customers.
- tcidp stands for – test case id for products.
- tcido stands for – test case id for orders.
- customers – pytest -m customers. (Run all the costumer tests)
- products – pytest -m products. (Run all the product tests.)
- orders – pytest -m orders. (Run all the orders tests)

Selenium Section:

- tcids stands for – test case id for selenium.

Run all tests - pytest

*) check the table below for every test and for its id.

Section	ID	Type	COMMAND	FunctionName
BACKEND	tcidc1	customers	Pytest -m tcidc1	test_create_customer_only_email_password()
BACKEND	tcidc2	customers	Pytest -m tcidc2	test_create_customer_fail_for_existing_email()
BACKEND	tcidc3	customers	Pytest -m tcidc3	test_get_all_customers()
BACKEND	tcidp1	products	Pytest -m tcidp1	test_list_all_products()
BACKEND	tcidp2	products	Pytest -m tcidp2	test_list_product_with_specific_id()
BACKEND	tcidp3	products	Pytest -m tcidp3	test_create_single_product():
FRONTEND	tcids1	Smoke	Pytest -m tcids1	test_login_none_existin_user(self)
FRONTEND	tcids2	Smoke	Pytest -m tcids2	test_register_new_user(self):
FRONTEND	tcids3	Smoke	Pytest -m tcids3	test_register_new_user_failed(self):
FRONTEND	tcids4	End to end	Pytest -m tcids4	test_end_to_end_checkout_guest_user(self)

- if you would like to add more tests here the docs.
- <https://woocommerce.github.io/woocommerce-rest-api-docs/#introduction>

Docker Section:

- Make sure you're on the root folder or provide the path of your Docker file instead of the dot in the first step.
- Im copying the Credentials manually you can skip step 3 and 4 just by adding to your Docker file "COPY ./ Credentials.sh /automation" **without the quotes**.
- [Docker Official Site](#).

Linux Section (Container):

1. docker build -t automation • (pay attention to the dot)
2. docker run -it automation /bin/bash
3. open new terminal (don't close the first one)-> docker ps _> copy the name of the container
4. run this command:
docker cp Credentials.sh {the name you've copied}:/automation
5. go back to the first terminal.
6. Source Credentials.sh
7. pytest

Now you can run the tests the way you want.

Windows Section (Container):

- All the steps are the same but the second step.
- **/bin/sh** replace it with /cmd.exe or /powershell.exe .