# **Generating html report by executing tests via command line:**

(Automation\_VEnv) PS C:\LocalData\CPA\Python46\Source\_code\_and\_notes\Projects\Automation\FirstPythonProject\tests> **pytest -m login --html=../reports/login\_test\_report.html**

This will generate reports/login\_test\_report.html underFirstPythonProject folder.

# **what is http://localhost:63342**

[http://localhost:63342 is a URL that is used by JetBrains IDEs such as WebStorm, PyCharm, and IntelliJ IDEA to host web applications locally for development and testing purposes 1](https://stackoverflow.com/questions/19633778/change-webstorm-liveedit-port-63342)[2](https://wenku.csdn.net/answer/3a1d162a4318b6e1b98a5d8673d731aa)[3](https://blog.csdn.net/qq_16525829/article/details/121290008). [The port number 63342 is the default port used by JetBrains IDEs for this purpose 1](https://stackoverflow.com/questions/19633778/change-webstorm-liveedit-port-63342)[2](https://wenku.csdn.net/answer/3a1d162a4318b6e1b98a5d8673d731aa).

So try hosting you Flask application for testing under pycharm folder. Use it as test application.

# **Providing browser value through command line**

(Automation\_VEnv) PS C:\LocalData\CPA\Python46\Source\_code\_and\_notes\Projects\Automation\FirstPythonProject> **pytest -m login --browser="firefox"**

# **Sending tests to remote SSH accounts**

[Sending tests to remote SSH accounts — pytest-xdist documentation](https://pytest-xdist.readthedocs.io/en/stable/remote.html#multi-platform)

<https://pytest-xdist.readthedocs.io/en/stable/remote.html#multi-platform>

Try this to run test parallelly on remote machine.

(Automation\_VEnv) PS C:\LocalData\CPA\Python46\Source\_code\_and\_notes\Projects\Automation\FirstPythonProject> **pytest -m login -n auto**

This will automatically calculate available CPUs to run tests parallely and distributes test randomly across them.

(Automation\_VEnv) PS C:\LocalData\CPA\Python46\Source\_code\_and\_notes\Projects\Automation\FirstPythonProject> **pytest -m login -n 4**

This will run 4 parallel tests

(Automation\_VEnv) PS C:\LocalData\CPA\Python46\Source\_code\_and\_notes\Projects\Automation\FirstPythonProject> **pytest -m login -n=4**

Alternatively, you can also do this.

# **expected\_conditions.visibility\_of\_element\_located(By.)**

As a general rule use expected\_conditions.visibility\_of\_element\_located(By.)

Over expected\_conditions.presence\_of\_element\_located(By.) because an element may be present in DOM but not visible.

Many expected\_conditions need tuple as input, so need extra (). Otherwise gives syntax error

confirmation\_message = wait.until(expected\_conditions.visibility\_of\_element\_located((By.ID, "confirmation")))