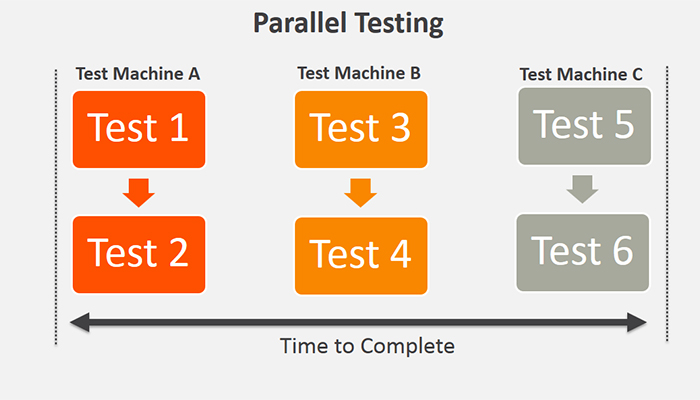
**Parrallel Testing**

**Thread Local**



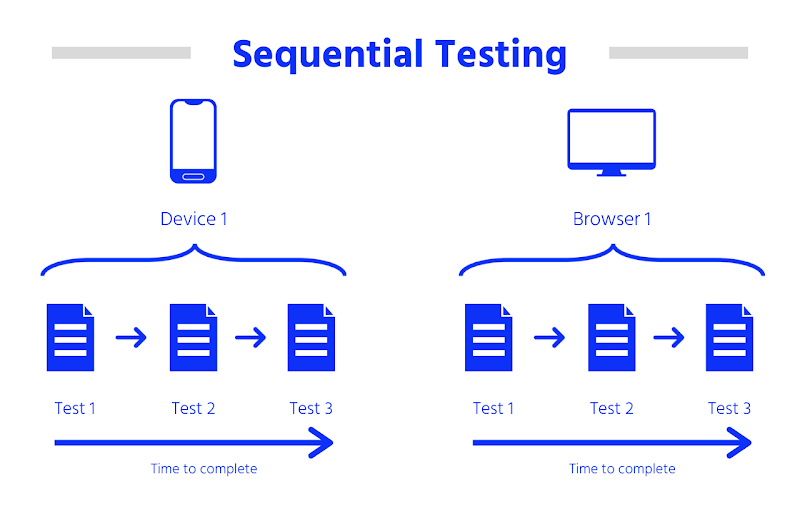
***DAO MINH DAM  
Email:***[*daominhdam@gmail.com*](mailto:daominhdam@gmail.com)

***Skype:****daominhdam (Đào Minh Đảm)*

[***https://automationfc.vn***](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://automationfc.vn%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211332000%26amp;usg%3DAOvVaw2yql4KCN4_N46ySOka4rXC&sa=D&source=editors&ust=1628670211396370&usg=AOvVaw2gtVUd-xiC_YLmmMN4PizQ)*-*[***https://automationfc.com***](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://automationfc.com%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211333000%26amp;usg%3DAOvVaw3raTFLVo4FWbohJ_rWehAB&sa=D&source=editors&ust=1628670211396551&usg=AOvVaw13ykLIdHWN15Pl8VoAqXS9)

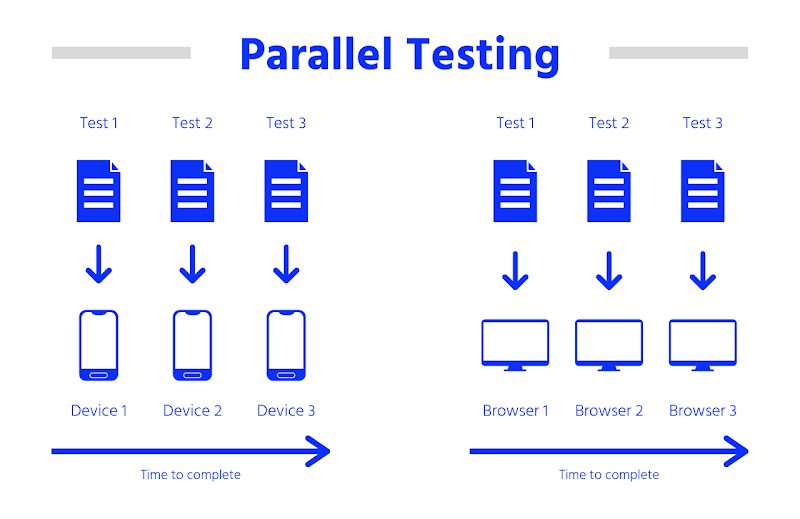
***Tài liệu này chỉ sử dụng trong khóa học - vui lòng không chia sẻ công khai***  
-------------------------------------------------------------------------------------------------------------------------------

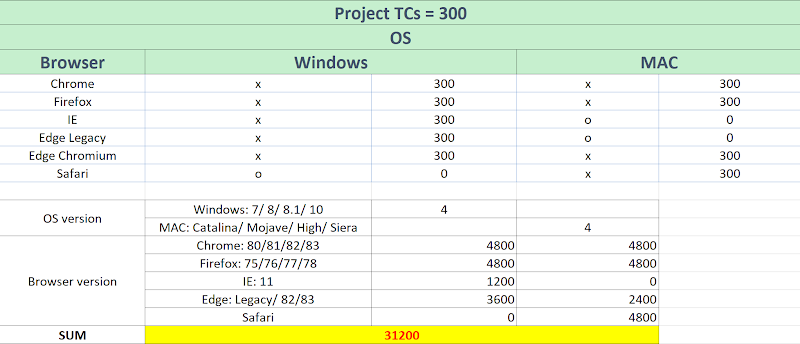
**Sequential/ Queue Testing**

****

* Chạy theo hàng đợi (FIFO - First In First Out) - sắp xếp theo thứ tự trong file testng.xml
* Chỉ phù hợp để chạy ở local vì ổn định - chạy 1 session (thread)/ time
* Ko phù hợp để chạy phân tán
* Nhược điểm: Kết quả test sẽ có chậm hơn

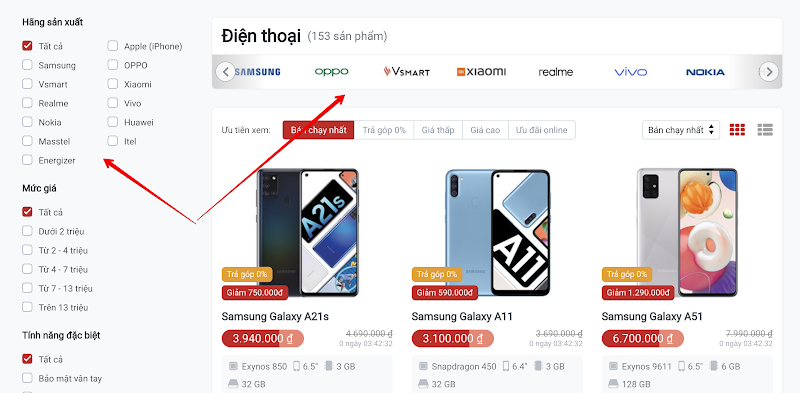
**Parallel Testing**

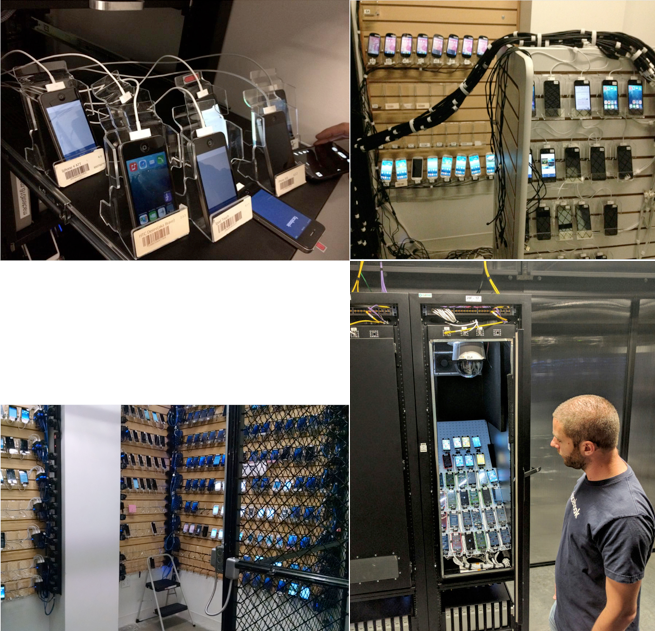




* **Build Lab for Development/ Testing**
* Đối với 1 số công ty lớn, có điều kiện và tiềm lực tài chính, con người, họ mua và chọn lựa tất cả các thiết bị có trên thị trường về để test app/ system. Một ví dụ tiêu biểu và nổi tiếng là Facebook đã thiết lập 1 khu test riêng với hơn 2000 thiết bị





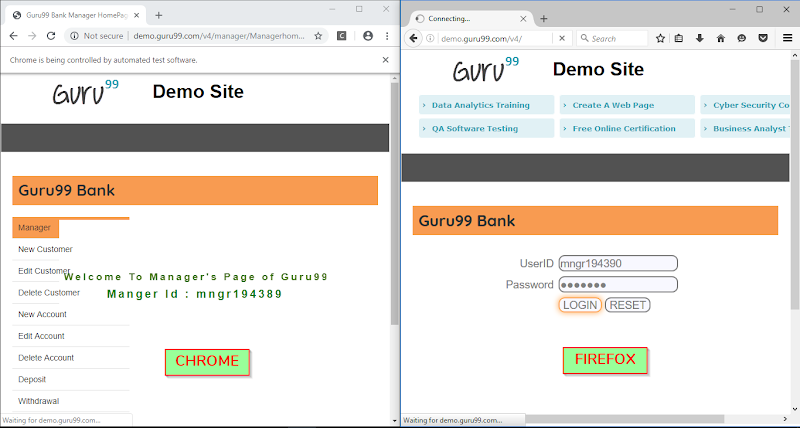


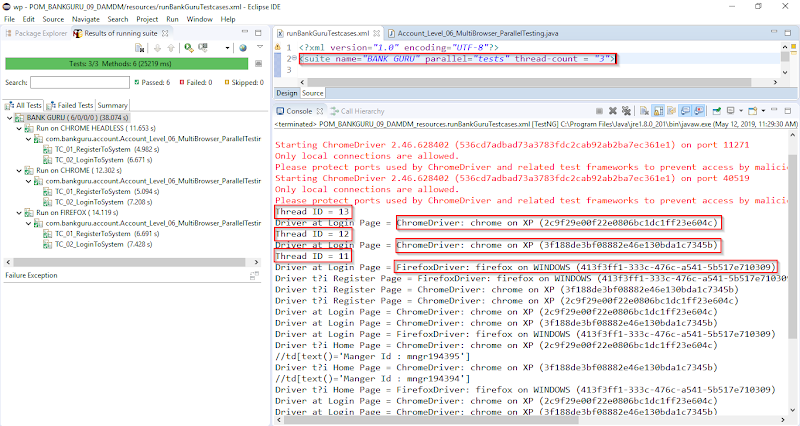


* Điểm mạnh:
* Reproduce đúng lỗi
* Không bỏ sót bất cứ loại hình thiết bị nào
* Điểm yếu:
* Tiêu tốn quá nhiều tài chính và công sức - phải cần 1 bộ phận chuyên gia đủ giỏi để setup một khu vực test desktop/ mobile - đảm bảo được các vấn đề về nguồn/ share màn hình/ truyền tín hiệu
* Khó hỗ trợ nhiều version cho browser/ device
* **Cloud Testing:**
* Service:
* **Browserstack**
* **AWS Device Farm**
* Firebase (Google)
* Xamarin Cloud Test (Microsoft - Visual Studio App Center)
* **Saucelab**
* Kobiton (KMS)
* Cross Browser Testing (Smartbear)
* Perfecto
* Đặc điểm:
* Cung cấp dịch vụ dạng SaaS - thanh toán theo giờ hoặc theo tháng
* Cho phép chạy đồng thời (parallel) trên nhiều thiết bị thực hoặc ảo
* Kịch bản test có thể ở 3 dạng: tương tác trực tiếp trên web browser (chạy chậm)/ automation (tự động tạo test script và điền dữ liệu test vào các form) và programming (tạo kịch bản theo test script)
* Kết quả trả về ở các dạng: ảnh chụp screenshot/ video quay màn hình/ kết quả (pass hoặc fail/ trong đó điều kiện pass/ fail không rõ ràng/ chỉ fail khi bị crash và ko config được)
* Test được cả với Android và iOS, sử dụng file apk hoặc ipa
* Điểm mạnh:
* Kế thừa các thế mạnh của cloud, cách làm này có thể dùng được luôn mà không cần đầu tư gì, bật/tắt và thanh toán theo yêu cầu. Các thông số môi trường có thể config dễ dàng
* Chạy được đa dạng các loại thiết bị - chạy tự động ko mất nhiều công sức con người, kết quả ở dạng video trực quan và ảnh chụp màn hình
* Ko phải mua sắm thiết bị - theo đó tiết kiệm được chi phí đầu tư
* Điểm yếu:
* Chỉ hỗ trợ các thương hiệu lớn và phổ biến, ko hỗ trợ test trên các loại thiết bị trôi nổi và yếu hơn như Oppo, Vinsmart, Xiaomi, Huawei, Pantech,.. mà hiện nay tỉ lệ các thiết bị này ở Việt nam là rất lớn
* Kết quả pass/fail trả về không có nhiều giá trị do ko config được điều kiện fail, do đó việc chạy tự động không hiệu quả, phải tự phân tích trên video
* Chi phí vận hành không nhỏ lên tới từ 100 USD/ tháng
* **Parallel Testing:**
* Dùng để phân tán (distribute) số lượng testcase lớn qua nhiều browser và chạy song song
* FIREFOX
* CHROME
* HEADLESS
* ..
* Nhược điểm:
* Cấu hình Server/ PC run test ko đủ mạnh để chạy nhiều session/ 1 time
* Tranh chấp/ bị xung đột các sự kiện chuột/ bàn phím (User Interaction) nếu đều sử dụng cùng lúc
* Data Test khó control (tăng lên/ giảm xuống/ hoặc dùng chung bộ data) nếu chạy các class test cùng lúc
* **Session 1 - Firefox**: Tạo dữ liệu (Cùng chức năng)
* **TC\_01\_Create -> Tạo ra 1 User**
* **Session 2 - Chrome**: Search (Cùng chức năng)
* TC\_01\_Search: 1000 Users
* TC\_02\_Search: 1000 Users # 1001 Users sau khi **TC\_01\_Create**finish
* **Session 3 - Headless**: Edit (Cùng chức năng)
* **Solution:** Quản lí data test **ko phụ thuộc lẫn nhau giữa nhiều class test**
* **testng.xml**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> <**suite** name="BANK GURU" parallel="tests" thread-count = "3">          <**test** name="Run on FIREFOX">                 <**parameter** name="browser" value="firefox" />                 <**classes**>                         <**class** name="com.bankguru.account.Account\_Level\_06\_MultiBrowser\_ParallelTesting" />                 </**classes**>         </**test**>          <**test** name="Run on CHROME">                 <**parameter** name="browser" value="chrome" />                 <**classes**>                         <**class** name="com.bankguru.account.Account\_Level\_06\_MultiBrowser\_ParallelTesting" />                 </**classes**>         </**test**>                  <**test** name="Run on CHROME HEADLESS">                 <**parameter** name="browser" value="chromeheadless" />                 <**classes**>                         <**class** name="com.bankguru.account.Account\_Level\_06\_MultiBrowser\_ParallelTesting" />                 </**classes**>         </**test**> </**suite**> |

* **Result**

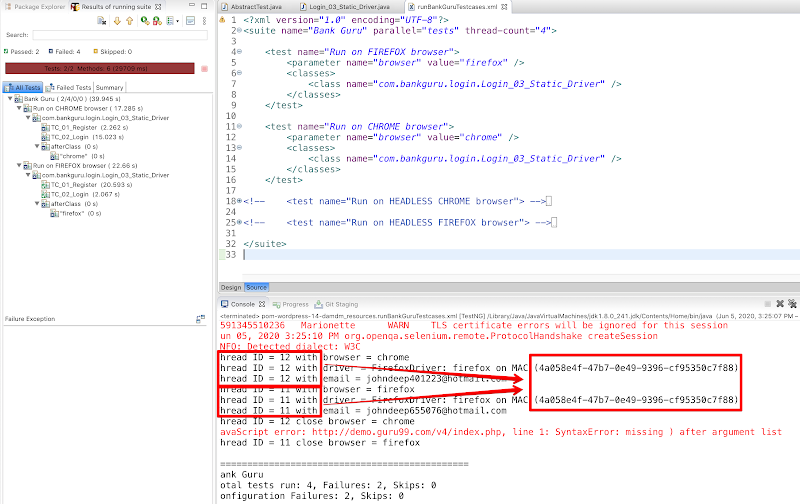




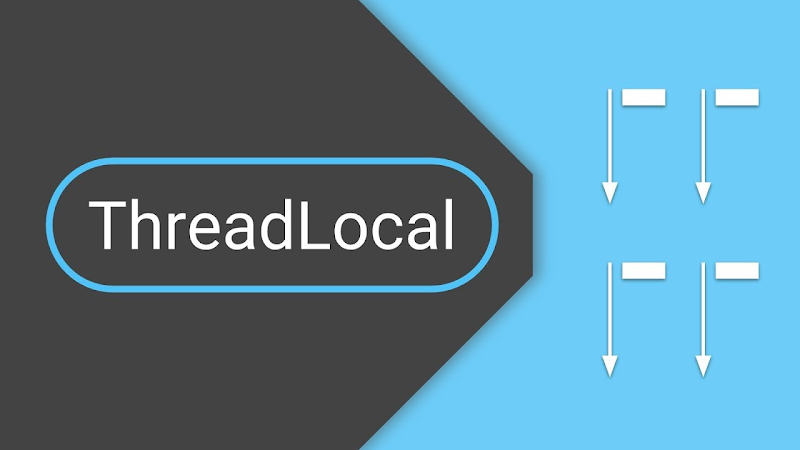
**Parallel with static driver**

* Nhiều framework design với biến driver là static trong BaseTest và gọi qua sử dụng ở các Test Class tránh việc khởi tạo nhiều đối tượng driver -> Chỉ đúng vs trường hợp chạy Sequential
* Biến static thuộc Class quản lí - ko thuộc về instance của Class
* Chia sẻ state/ data xuyên suốt hệ thống từ lúc Start JVM (Java Virtual Machine) cho đến khi kết thúc
* Được sử dụng để quản lý/ tiết kiệm bộ nhớ tốt hơn và nó có thể được truy cập trực tiếp thông qua lớp mà không cần khởi tạo
* **NOTE:**
* Chỉ nên sử dụng static khi cần chia sẻ các phương thức/ thuộc tính cho toàn bộ hệ thống
* Data không thống nhất -> nếu biến/ method là static nó sẽ luôn lấy lại dữ liệu của lần gán giá trị gần nhất
* *Run multi-thread -> data không nhất quán (ko threadsafe - dữ liệu của thread này nhưng thread khác vẫn sử dụng được)*
* Không sử dụng static -> dữ liệu của đối tượng sẽ khác nhau/ không liên quan với nhau
* Bảo mật/ phân quyền: Được dùng như biến global nên phạm vi rộng, trong việc thiết kế phần mềm thì ko nên sử dụng bừa bãi

|  |
| --- |
| *// AbstractTest* protected static WebDriver driver;  *// Testcase* @Parameters("browser") @BeforeClass public void beforeClass(String browserName) {         email = "johndeep" + randomNumber() + "@hotmail.com";         getBrowserDriver(browserName, "http://demo.guru99.com/v4/index.php");         System.out.println("Thread ID = " + Thread.currentThread().getId() + " with browser = " + browserName);         System.out.println("Thread ID = " + Thread.currentThread().getId() + " with driver = " + driver.toString());         System.out.println("Thread ID = " + Thread.currentThread().getId() + " with email = " + email);         loginPage = new LoginPageObject(driver);         loginPageUrl = loginPage.getLoginPageUrl(); }  @Parameters("browser") @AfterClass(alwaysRun = true) public void afterClass(String browserName) {         System.out.println("Thread ID = " + Thread.currentThread().getId() + " close browser = " + browserName);         driver.quit(); }  *// testng.xml* <?xml version="1.0" encoding="UTF-8"?> <suite name="Bank Guru" parallel="tests" thread-count="2">          <test name="Run on FIREFOX browser">                 <parameter name="browser" value="firefox" />                 <classes>                         <class name="com.bankguru.login.Login\_03\_Static\_Driver" />                 </classes>         </test>          <test name="Run on CHROME browser">                 <parameter name="browser" value="chrome" />                 <classes>                         <class name="com.bankguru.login.Login\_03\_Static\_Driver" />                 </classes>         </test>  </suite> |

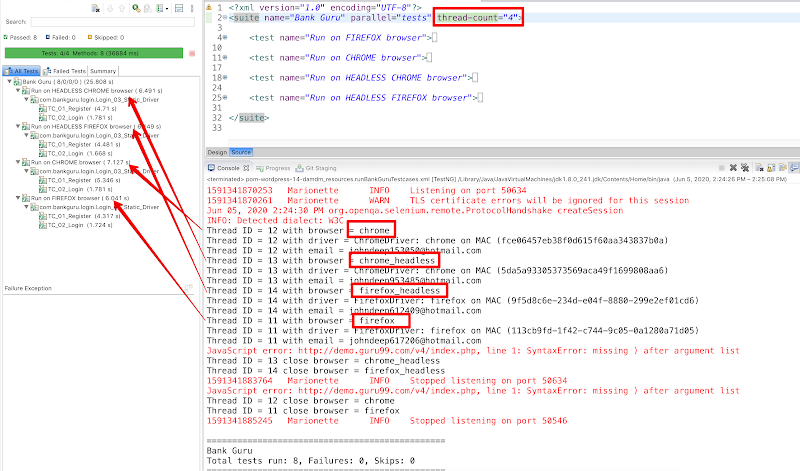
****

**Parallel with ThreadLocal driver**



* Thread safe với ThreadLocal: Mỗi luồng sẽ tạo ra data độc lập (vẫn sử dụng biến static)
* [https://gpcoder.com/3484-lap-trinh-da-luong-trong-java-java-multi-threading/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://gpcoder.com/3484-lap-trinh-da-luong-trong-java-java-multi-threading/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211354000%26amp;usg%3DAOvVaw0kDHieUj1Sl07XzhkjrUAi&sa=D&source=editors&ust=1628670211406712&usg=AOvVaw120pK2YWo1ZcVxllV7tLwO)
* [https://gpcoder.com/3514-dong-bo-hoa-cac-luong-trong-java/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://gpcoder.com/3514-dong-bo-hoa-cac-luong-trong-java/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211355000%26amp;usg%3DAOvVaw0RXIWToCA4n96oIEodbw1D&sa=D&source=editors&ust=1628670211407026&usg=AOvVaw3uLMxGJ60WrpXdNryFXdDt)

|  |
| --- |
| *// AbstractTest* private static ThreadLocal<WebDriver> tDriver = new ThreadLocal<WebDriver>();  protected WebDriver getDriver(String browserName, String appUrl) {                 if (browserName.equalsIgnoreCase("firefox")) {                         WebDriverManager.firefoxdriver().setup();                         tDriver.set(new FirefoxDriver());                 } else if (browserName.equalsIgnoreCase("chrome")) {                         WebDriverManager.chromedriver().setup();                         tDriver.set(new ChromeDriver());                 } else if (browserName.equalsIgnoreCase("edge")) {                         WebDriverManager.edgedriver().setup();                         tDriver.set(new EdgeDriver());                 } else if (browserName.equalsIgnoreCase("ie")) {                         WebDriverManager.iedriver().arch32().setup();                         tDriver.set(new EdgeDriver());                 } else if (browserName.equalsIgnoreCase("safari")) {                         tDriver.set(new SafariDriver());                 } else if (browserName.equalsIgnoreCase("chrome\_headless")) {                         WebDriverManager.chromedriver().setup();                         ChromeOptions options = new ChromeOptions();                         options.addArguments("headless");                         options.addArguments("window-size=1920x1080");                         tDriver.set(new ChromeDriver(options));                 } else if (browserName.equalsIgnoreCase("firefox\_headless")) {                         WebDriverManager.firefoxdriver().setup();                         FirefoxOptions firefoxOpt = new FirefoxOptions();                         firefoxOpt.addArguments("-headless");                         firefoxOpt.addArguments("window-size=1920x1080");                         tDriver.set(new FirefoxDriver(firefoxOpt));                 } else {                         new RuntimeException("Please choose the browser name!");                 }                 tDriver.get().manage().timeouts().implicitlyWait(GlobalConstants.LONG\_TIMEOUT, TimeUnit.SECONDS);                 tDriver.get().get(appUrl);                 return tDriver.get();         }          protected void removeDriver() {                 tDriver.get().quit();                 tDriver.remove();         }  *// Test Class* private WebDriver driver; driver = getDriver(browserName, "http://demo.guru99.com/v4/index.php"); |



**Singleton and ThreadLocal**

* [https://www.linkedin.com/pulse/journey-singleton-aminul-karim/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://www.linkedin.com/pulse/journey-singleton-aminul-karim/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211361000%26amp;usg%3DAOvVaw3rpiPGpE-OMlojeSzITlbX&sa=D&source=editors&ust=1628670211410151&usg=AOvVaw37jYa08InVLMXXnqANL47r)
* [https://amiyasahu.github.io/singleton-class-per-thread-in-java.html](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://amiyasahu.github.io/singleton-class-per-thread-in-java.html%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211362000%26amp;usg%3DAOvVaw0a3-WAVXV8vCAmptxz6IQ8&sa=D&source=editors&ust=1628670211410409&usg=AOvVaw3SDlnEwI5mT0zO41wHlh0v)

**Reference**

* [https://bitbar.com/blog/parallel-testing-what-it-is-and-why-you-should-adopt-it/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://bitbar.com/blog/parallel-testing-what-it-is-and-why-you-should-adopt-it/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211363000%26amp;usg%3DAOvVaw3NfRUIJkmGxT6ufpK5TjKg&sa=D&source=editors&ust=1628670211410729&usg=AOvVaw1S0BlWhUIoGr8BdO1mo4ea)
* [http://makeseleniumeasy.com/2020/04/13/are-your-selenium-scripts-ready-for-parallel-execution/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttp://makeseleniumeasy.com/2020/04/13/are-your-selenium-scripts-ready-for-parallel-execution/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211363000%26amp;usg%3DAOvVaw2fXRjRFbSqZUVA9U6doSxM&sa=D&source=editors&ust=1628670211410973&usg=AOvVaw3NHk22efBmyUcuT9CNTd0p)
* [http://makeseleniumeasy.com/2020/04/11/problems-in-parallel-execution-with-static-webdriver/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttp://makeseleniumeasy.com/2020/04/11/problems-in-parallel-execution-with-static-webdriver/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211364000%26amp;usg%3DAOvVaw2GPr6CO0Oi-xK5NDmosCu3&sa=D&source=editors&ust=1628670211411221&usg=AOvVaw2zBkuiPZDoq5cXlm3B4IKW)
* [http://makeseleniumeasy.com/2020/04/09/importance-of-using-threadlocal-variables-in-parallel-execution-of-automation-scripts/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttp://makeseleniumeasy.com/2020/04/09/importance-of-using-threadlocal-variables-in-parallel-execution-of-automation-scripts/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211364000%26amp;usg%3DAOvVaw3PRaRbXXHOya6Nb27ac3fM&sa=D&source=editors&ust=1628670211411401&usg=AOvVaw1zHkrPOeYkOUEL9LiNEKzB)
* [http://makeseleniumeasy.com/2020/05/27/threadlocal-static-webdriver-for-parallel-execution/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttp://makeseleniumeasy.com/2020/05/27/threadlocal-static-webdriver-for-parallel-execution/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211365000%26amp;usg%3DAOvVaw3M2CwItg2WcClWdnuNAgfD&sa=D&source=editors&ust=1628670211411588&usg=AOvVaw0Gs3FlRFOGUdKNhdAMP04y)
* [https://blogs.bmc.com/what-is-parallel-testing-parallel-testing-explained/?print=print](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://blogs.bmc.com/what-is-parallel-testing-parallel-testing-explained/?print%253Dprint%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211365000%26amp;usg%3DAOvVaw0CeDWIvZ1yP2R9Y45xWRqW&sa=D&source=editors&ust=1628670211411745&usg=AOvVaw1-w9SklKhKt4UP67xBaqie)
* [https://www.swtestacademy.com/selenium-parallel-tests-grid-testng/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://www.swtestacademy.com/selenium-parallel-tests-grid-testng/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211365000%26amp;usg%3DAOvVaw1IPQNhX-Lh_ZK_KYO3jAx6&sa=D&source=editors&ust=1628670211411906&usg=AOvVaw0LFbnUiDMH1AF_kq_zgywg)
* [https://automationrhapsody.com/avoid-multithreading-problems-java-using-threadlocal/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://automationrhapsody.com/avoid-multithreading-problems-java-using-threadlocal/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211366000%26amp;usg%3DAOvVaw38HxW8QJiHqSbaK9o2mwh3&sa=D&source=editors&ust=1628670211412168&usg=AOvVaw1JPQRadeMRVaedayz7b8IH)
* [http://seleniumautomationhelper.blogspot.com/2014/02/initializing-webdriver-object-as-thread.html](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttp://seleniumautomationhelper.blogspot.com/2014/02/initializing-webdriver-object-as-thread.html%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211366000%26amp;usg%3DAOvVaw2ntg44HoDAOKzf6DgAPiq3&sa=D&source=editors&ust=1628670211412414&usg=AOvVaw02pp_Da4MS9qu_0W980S6I)
* [https://www.toolsqa.com/testng/testng-parallel-execution/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://www.toolsqa.com/testng/testng-parallel-execution/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211367000%26amp;usg%3DAOvVaw0Ii-AK855C9yXSjRyIIrL5&sa=D&source=editors&ust=1628670211412645&usg=AOvVaw3j5ugxKYGSdDWqHxFa-so3)
* [https://uxvietnam.com/kiem-thu-ux-cho-ung-dung-mobile-user-test-tren-nhieu-thiet-bi-multi-devices/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://uxvietnam.com/kiem-thu-ux-cho-ung-dung-mobile-user-test-tren-nhieu-thiet-bi-multi-devices/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628670211367000%26amp;usg%3DAOvVaw1OpfSoFpyBDXfQDUrH-BNY&sa=D&source=editors&ust=1628670211412857&usg=AOvVaw1EBf1GuHYt47TqXla9QFJn)