

**BỘ TÀI NGUYÊN VÀ MÔI TRƯỜNG**  
**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**  
**KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THĂM**



**ĐỒ ÁN MÔN HỌC :**  
**ĐẢM BẢO CHẤT LƯỢNG VÀ KIỂM THỬ PHẦN MỀM**  
**ĐỀ TÀI: SỬ DỤNG SCRIPT ĐỂ KIỂM THỬ**



Giảng viên hướng dẫn: **Ths. Nguyễn Thị Ngọc Thanh**

Sinh viên thực hiện: **Trần Gia Huy ~ 0650080100**

**Võ Thị Thu Thảo ~ 0650080121**

Lớp : **06 ĐHCNTT3**

Khoá : **2017 – 2021**

*Tp.Hồ Chí Minh, ngày 30 tháng 12 năm 2019*

**BỘ TÀI NGUYÊN VÀ MÔI TRƯỜNG**  
**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**  
**KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THĂM**



**ĐỒ ÁN MÔN HỌC :**  
**ĐẢM BẢO CHẤT LƯỢNG VÀ KIỂM THỬ PHẦN MỀM**  
**ĐỀ TÀI: SỬ DỤNG SCRIPT ĐỂ KIỂM THỬ**



Giảng viên hướng dẫn: **Ths. Nguyễn Thị Ngọc Thanh**

Sinh viên thực hiện: **Trần Gia Huy ~ 0650080100**

**Võ Thị Thu Thảo ~ 0650080121**

Lớp : **06 ĐHCNTT3**

Khoá : **2017 – 2021**

*Tp.Hồ Chí Minh, ngày 30 tháng 12 năm 2019*

## LỜI CẢM ƠN

Trong suốt thời gian học tập tại trường Đại học Tài Nguyên và Môi Trường TP.Hồ Chí Minh , chúng em đã được học thêm rất nhiều kiến thức bổ ích. Những kiến thức liên quan đến chuyên ngành và cả những kiến thức từ việc chia sẻ về đời sống thực tế của thầy cô, học hỏi từ bạn bè và xã hội. Chúng em được trải nghiệm về đời sống sinh viên, những trải nghiệm chưa từng có trước đây khi còn là học sinh... Và quan trọng hơn là chúng em đã được trang bị rất nhiều các kiến thức, kỹ năng về chuyên ngành Công Nghệ Thông Tin để làm nền tảng bước vào đời với ngành nghề. Chúng em xin chân thành bày tỏ lòng biết ơn sâu sắc đến cô Nguyễn Thị Ngọc Thanh, người đã trực tiếp hướng dẫn chúng em về cách thực hiện đề tài báo cáo cũng như đưa ra những ý kiến để chúng em có thể hoàn thiện tốt hơn bài báo cáo của mình.

Trong quá trình làm báo cáo, chúng em đã cố gắng làm một cách tốt nhất nhưng vẫn còn những điểm hạn chế về kiến thức và kỹ năng nên có thể còn thiếu sót nhiều trong bài báo cáo. Vì vậy, chúng em rất mong nhận được sự thông cảm cũng như đóng góp ý kiến của cô để chúng em có thể hoàn thiện báo cáo một cách tốt nhất.

Cuối cùng, chúng em xin kính chúc cô có nhiều sức khỏe, hạnh phúc và thành đạt trong công việc và cuộc sống.

Một lần nữa chúng em xin chân thành cảm ơn cô!

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên sinh viên thực hiện:

Trần Gia Huy ~ 0650080100

Võ Thị Thu Thảo ~ 0650080121

Lớp: 06CNTT3

Tên đề tài: SỬ DỤNG SCRIPT ĐỂ KIỂM THỬ

1. Tiến độ và thái độ thực hiện của sinh viên:

- Mức độ thực hiện:

.....  
.....

2. Nội dung báo cáo:

➤ Thực hiện các nội  
dung:.....

➤ Khả năng hiểu biết về thực tế và lý thuyết:.....

3. Hình thức trình bày:.....

4. Một số ý kiến khác: .....

5. Đánh giá của giáo viên hướng dẫn:.....

ĐIỂM

Điểm sinh viên 1: .....

Điểm sinh viên 2: .....

# MỤC LỤC

Chương 1 : Tổng quan về kiểm thử phần mềm .....	1
1.1. Khái niệm kiểm thử phần mềm : .....	1
Chương 2 : Tổng quan về Script trong việc kiểm thử .....	3
2.1. Khái niệm về Script : .....	3
2.2. Khái niệm Script trong kiểm thử : .....	3
2.3. Khái niệm kiểm thử tự động ( Automation Testing ) : .....	3
Chương 3 : Thực nghiệm sử dụng công cụ để viết các Script cho quá trình kiểm thử. ....	4
3.1. Viết Script và Test bằng công cụ JUnit 4: .....	4
3.1.1. Giới thiệu về công cụ JUnit : .....	4
3.1.2. Giới thiệu về MAVEN : .....	9
3.1.3. Demo viết Script và kiểm thử bằng công cụ JUnit4: .....	16
3.2. Viết Script và Test bằng công cụ Appium : .....	20
3.2.1. Giới thiệu về Appium : .....	20
3.2.2. Demo viết Script và kiểm thử bằng công cụ Appium : .....	26

# LỜI MỞ ĐẦU

Trước khi đi sâu vào nghiên cứu bài báo cáo môn học, nhóm em tiến hành làm rõ những vấn đề sau đây: Lý do chọn đề tài; Mục tiêu nghiên cứu; Đối tượng và phạm vi nghiên cứu; Phương pháp nghiên cứu và kết cấu của bài báo cáo môn học.

## 1. Lý do chọn đề tài :

Với sự phát triển không ngừng nghỉ của ngành công nghệ thông tin nói chung và các công nghệ phần mềm nói riêng, việc phát triển các phần mềm đang ngày càng được hỗ trợ bởi nhiều các công cụ tân tiến, giúp việc xây dựng các phần mềm dễ dàng hơn và hiệu quả hơn. Tuy nhiên, vì mức độ phức tạp của các phần mềm cũng như những giới hạn về thời gian và kinh phí nên các hoạt động đảm bảo chất lượng phần mềm nói chung và kiểm thử nói riêng dù ngày càng củng cố và khoa học nhưng vẫn không thể đảm bảo các sản phẩm phần mềm không có lỗi. Lỗi vẫn luôn tiềm ẩn trong mọi sản phẩm phần mềm, có thể gây những thiệt hại đáng kể.

Kiểm thử phần mềm là một quá trình phải xảy ra liên tục trong mọi giai đoạn phát triển phần mềm mục đích để đảm bảo rằng phần mềm thoả mãn các yêu cầu thiết kế và các yêu cầu đó thoả mãn các nhu cầu của người dùng, việc kiểm thử phần mềm đã trở thành quy trình bắt buộc trong các dự án phát triển phần mềm trên thế giới. Kiểm thử phần mềm là một hoạt động rất tốn kém, mất thời gian, và khó có thể phát hiện được hết lỗi. Vì vậy, việc kiểm thử phần mềm đòi hỏi phải có chiến lược phù hợp, một kế hoạch hợp lý và việc thực hiện được quản lý chặt chẽ. Kiểm thử phần mềm là một lĩnh vực rất rộng, nhưng có thể được phân loại thành hai loại thử nghiệm là kiểm thử thủ công (manual testing) và kiểm thử tự động (automation testing). Trong đó kiểm thử tự động sau khi có kịch bản test thì Tester sẽ viết script để chạy tự động tất cả các bước bao gồm nhập thông tin, click vào nút và kiểm tra kết quả, so sánh kết quả chạy thực tế với kết quả mong muốn. Nên việc viết một Test Script rất quan trọng trong quá trình kiểm thử tự động. Đó là lý do nhóm em chọn đề tài “Sử dụng Script để kiểm thử” làm đề án báo cáo môn học “Đảm bảo chất lượng và kiểm thử phần mềm”.

## 2. Mục đích nghiên cứu:

Đề tài tìm hiểu cơ sở lý thuyết về kiểm thử nói chung và kiểm thử tự động nói riêng cũng như cách triển khai công cụ kiểm thử phần mềm tự động để giảm nhân lực kiểm thử và đảm bảo chất lượng phần mềm hơn với công việc kiểm thử bằng tay. Mục tiêu chính của đề tài là nghiên cứu về sử dụng Script để kiểm thử

### **3. Đối tượng và phạm vi nghiên cứu:**

Đề án nghiên cứu lý thuyết kiểm thử phần mềm. Bên cạnh đó, nghiên cứu các vấn đề về kiểm thử phần mềm tự động bởi các Script.

Phương pháp nghiên cứu: Nghiên cứu tổng quan về kiểm thử phần mềm và các kỹ thuật kiểm thử từ đó áp dụng vào kiểm thử phần mềm tự động, viết các kịch bản script để có thể thực hiện quá trình kiểm thử.

### **4. Kết cấu bài báo cáo :**

Với mục tiêu đặt ra như vậy, những nội dung và kết quả nghiên cứu chính của đề án được trình bày trong ba chương như sau:

Chương 1: Tổng quan về kiểm thử phần mềm:

Chương 2: Tổng quan về Script trong việc kiểm thử.

Chương 3: Thực nghiệm sử dụng công cụ để viết các script cho quá trình kiểm thử.

Đề án báo cáo môn học “Đảm bảo chất lượng và kiểm thử phần mềm” với đề tài “Sử dụng Script để kiểm thử”. Trong quá trình thực hiện đề án, do thời gian cũng như trình độ của em còn có những hạn chế nhất định nên không thể tránh khỏi những sai sót. Rất mong nhận được sự góp ý của cô và các bạn để đề án hoàn thiện hơn. Em xin chân thành cảm ơn sự hướng dẫn, và giúp đỡ tận tình của giảng viên ThS. Nguyễn Thị Ngọc Thanh đã giúp đỡ em trong quá trình học tập cũng như trong quá trình làm đề án.

Sinh viên thực hiện :

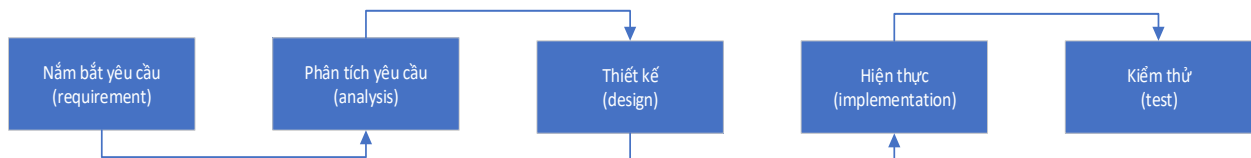
Trần Gia Huy ~ 0650080100

Võ Thị Thu Thảo ~ 0650080

## Chương 1 : Tổng quan về kiểm thử phần mềm

### 1.1. Khái niệm kiểm thử phần mềm :

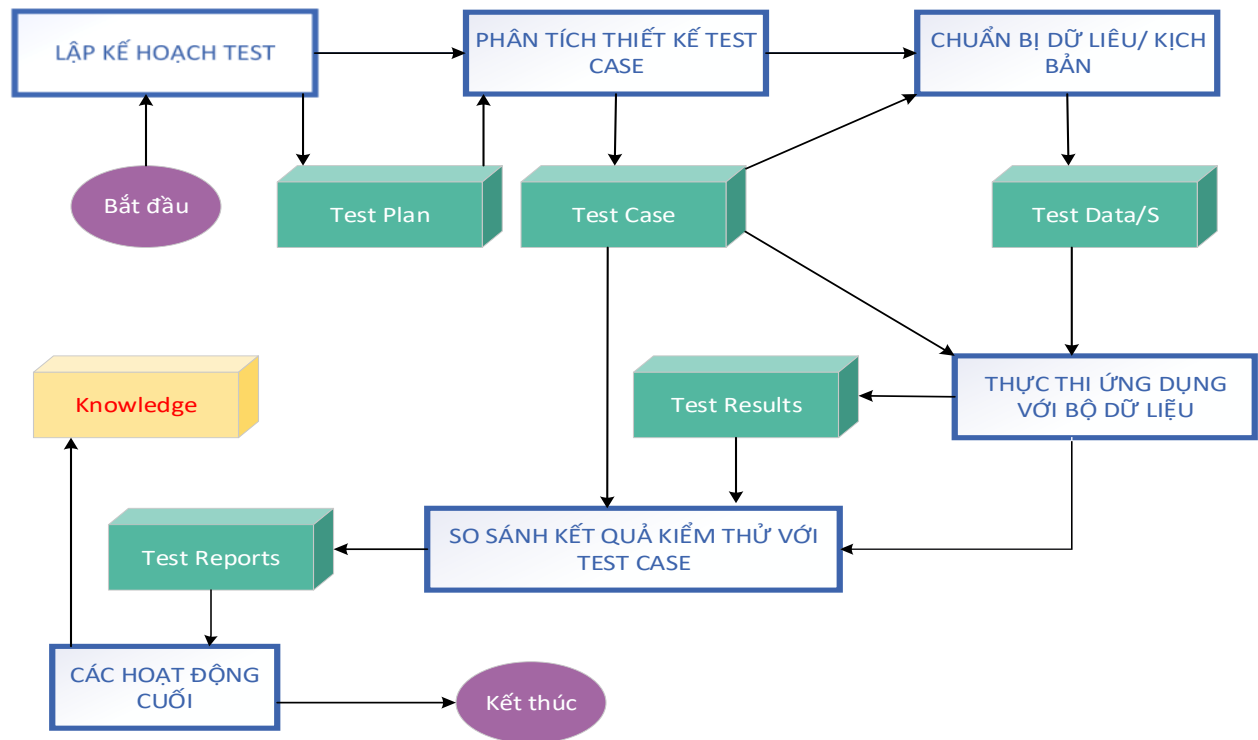
- Kiểm thử phần mềm là quá trình kiểm thử chương trình với mục tiêu tìm ra các lỗi, đảm bảo phần mềm ứng dụng hoạt động chính xác, đầy đủ chức năng và đáp ứng đầy đủ yêu cầu cũng như nhu cầu của khách hàng và người sử dụng.
  - Kiểm thử phần mềm cung cấp những cái nhìn độc lập về phần mềm giúp việc đánh giá và hiểu được phần nào rủi ro khi xây dựng ứng dụng.
  - Việc kiểm thử được bắt đầu khi có các yêu cầu được xác định.
  - Hoạt động kiểm thử bao gồm việc lên kế hoạch và kiểm soát việc kiểm thử, lựa chọn các điều kiện, thiết kế và xây dựng các test case, lưu vết kết quả, xác định khi nào nên dừng việc kiểm thử, xây dựng báo cáo và tổng kết về kết quả kiểm thử khi hoàn thành việc kiểm thử.
- ❖ Mục tiêu việc kiểm thử :
- Tìm kiếm được lỗi.
  - Đảm bảo được mức độ chất lượng sản phẩm .
  - Cung cấp thông tin để đưa ra các quyết định
  - Ngăn ngừa các lỗi trong lúc xây dựng.
- ❖ Các hoạt động khi tạo phần mềm :



Hình 1. Các hoạt động khi tạo phần mềm

- ❖ Mục đích việc kiểm thử :
- Đánh giá chất lượng hoặc mức độ chấp nhận của sản phẩm.
  - Phát hiện ra lỗi hoặc vấn đề bất thường của sản phẩm.





Hình 2. Sơ đồ tổng quan việc thực hiện kiểm thử

## Chương 2 : Tổng quan về Script trong việc kiểm thử

### 2.1. Khái niệm về Script :

- Script là bản thảo viết dưới dạng mã kịch bản theo ngôn ngữ lập trình.
- Script là chương trình có thể chạy ngay khi bạn viết source code.
- Thông thường chương trình sau khi được viết ra thông qua bộ biên dịch (compiler) sẽ được chuyển đổi thành code của ngôn ngữ máy tính mà máy tính có thể dịch được, sau đó nó kết hợp với các library thích hợp, từ đó hình thành nên một chương trình chạy được.

### 2.2. Khái niệm Script trong kiểm thử :

- Test Script là một tập cách lệnh được viết bằng Script thực hiện trên một hệ thống được kiểm tra để xác minh hệ thống hoạt động như mong đợi
- Test Script được sử dụng trong Automation Testing.
- Thường mọi người sẽ lầm tưởng khái niệm và chức năng của Test Script và Test Case, thực ra cả hai đều có chung một số khái niệm, điểm khác nhau giữa chúng chính là : Test case được viết ra để kiểm thử bằng manual test, còn test script viết lên để chạy trong automation test.

### 2.3. Khái niệm kiểm thử tự động ( Automation Testing ):

- Automation Testing là việc kiểm thử phần mềm bằng một chương trình và ít sự tương tác của con người trong quá trình test.
- Trong Automation Testing, các tests được chạy để so sánh kết quả thực tế với kết quả mong đợi.
- Automation tests hoạt động hiệu quả khi thực hiện các kiểm tra lặp lại và hồi quy để đảm bảo ứng dụng vẫn hoạt động bình thường sau khi có vài đổi mới.

## Chương 3 : Thực nghiệm sử dụng công cụ để viết các Script cho quá trình kiểm thử.

### 3.1. Viết Script và Test bằng công cụ JUnit 4:



#### 3.1.1. Giới thiệu về công cụ JUnit :

- JUnit là một framework đơn giản dùng cho việc tạo các unit testing tự động và chạy các test có thể lặp đi lặp lại.
- JUnit là một chuẩn mực cho việc unit testing trong Java.

Ở đây nhóm em chọn phiên bản JUnit 4 vì nó có nhiều phương thức giả định hơn JUnit 5, và được nhiều người sử dụng hơn JUnit5

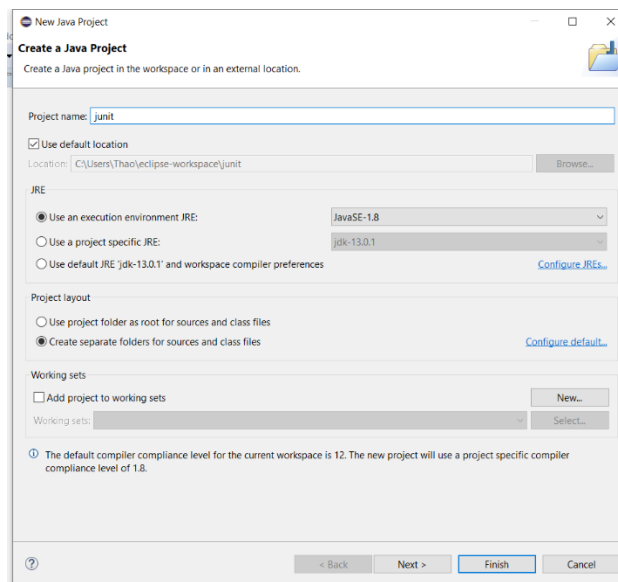
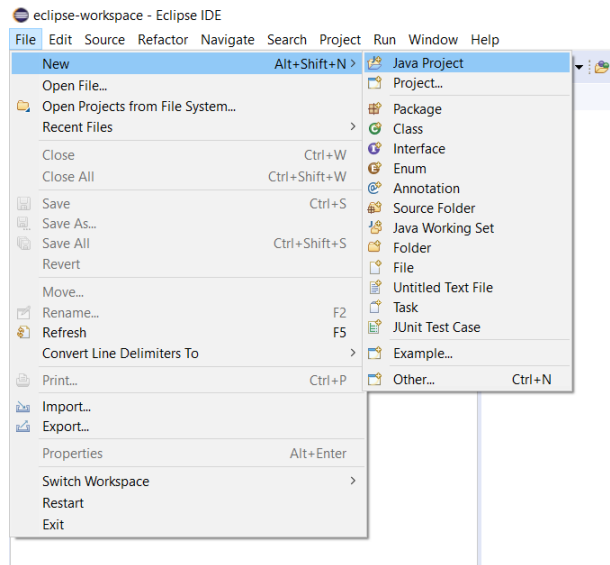
(tìm hiểu thêm sự khác nhau: <https://howtodoinjava.com/junit5/junit-5-vs-junit-4/> )

- Unit Test case : là 1 chuỗi code để đảm bảo các đoạn code dễ kiểm thử làm việc như mong muốn. Mỗi function sẽ gồm nhiều test case, ứng với mỗi trường hợp function chạy.
- Setup : là hàm được chạy trước khi test case, thường dùng để chuẩn bị dữ liệu để chạy test.
- Teardown : là hàm dùng khi chạy test case xong, dùng để xóa dữ liệu, giải phóng bộ nhớ.
- Mock : là đối tượng ảo, mô phỏng tính chất và hành vi giống với đối tượng thực được truyền vào trong khối mã đang vận hành nhằm kiểm tra so sánh tính chính xác của các hoạt động bên trong.
- Test Suite : là một tổ hợp các test.
- Assert : Mỗi test case thường xuất hiện một hay nhiều câu lệnh Assert, công dụng của nó là để kiểm tra tính đúng đắn của mỗi hàm. Sau đây là mô tả các phương thức assertXXX() khác nhau có trong lớp junit.Assert.
  - assertEquals(): lệnh so sánh hai giá trị xem xét chúng có bằng nhau không. Test sẽ đúng nếu các trị đem so sánh bằng nhau.

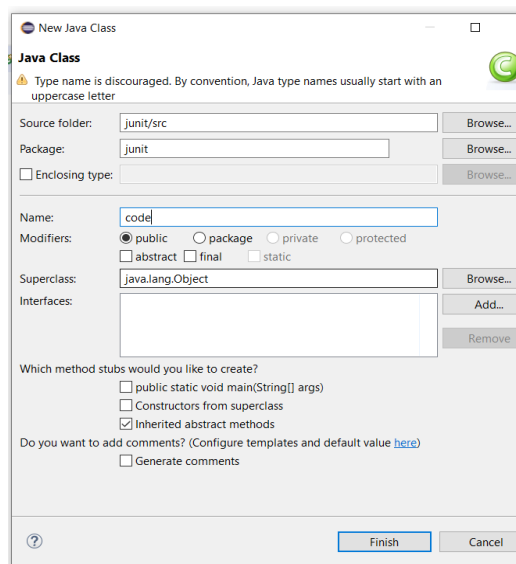
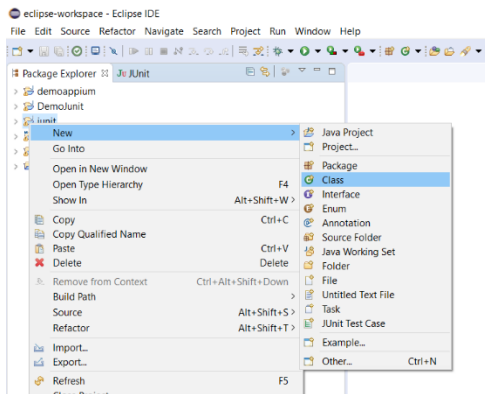
- `assertTrue()` : dùng để đánh giá một biểu thức logic. Test sẽ đúng nếu như biểu thức đúng, test sẽ `fail()` khi phương thức làm cho test hiện hành bị thất bại. Phương thức này thường được sử dụng để xử lý các biệt lệ.
- `assertFalse()` : dùng để đánh giá một biểu thức logic. Test sẽ được chấp thuận nếu biểu thức sai.
- `assertNull()` : dùng để so sánh tham chiếu của một đối tượng với giá trị `NULL`. Test sẽ đúng nếu như tham chiếu là `NULL`.
- `assertNotNull()` : dùng để so sánh tham chiếu của một đối tượng với giá trị `NULL`. Test sẽ đúng nếu như tham chiếu khác `NULL`.
- `assertSame()` : dùng để so sánh địa chỉ tham chiếu của hai đối tượng. Test sẽ đúng nếu cả hai cùng chiếu đến một đối tượng.
- `assertNotSame()` : dùng để so sánh địa chỉ tham chiếu của hai đối tượng. Test sẽ đúng nếu cả hai không chiếu đến một đối tượng.
- `assertThat()` : dùng để so sánh giá trị cụ thể và giá trị dự kiến có trùng khớp hay không. Test sẽ đúng nếu hai giá trị trùng khớp.
- **Assume** : Khi chúng ta sử dụng phương thức này thì khi một test không đúng, nó sẽ dừng test trường hợp đó và tiếp tục test các test khác, nếu bạn khai báo `@Ignore` cho hàm test, trường hợp test đó sẽ không được kiểm tra, tiếp tục thực hiện các test khác.
  - `Assume.assumeTrue()` : test sẽ không hợp lệ nếu các điều kiện đánh giá là sai.
  - `Assume.assumeFalse()` : test sẽ không hợp lệ nếu các điều kiện đánh giá là đúng.
  - `Assume.assumeNotNull()` : test sẽ không hợp lệ nếu tham chiếu của một đối tượng có giá trị là `NULL`.
  - `Assume.assumeThat()` : test sẽ không hợp lệ nếu giá trị cụ thể và giá trị dự kiến không trùng khớp.
  - `Assume.assumeNoException()` : test sẽ không hợp lệ nếu đối tượng thuộc `Exception` là `NULL`, tức là không có báo lỗi trong lúc test.
- **Parameterized** : là một phương thức test được thực hiện qua việc lặp đi lặp lại bằng cách sử dụng các tham số test khác nhau. Hai cách truyền đối số và phương thức unit test :
  - Truyền thông qua đối số của constructor ( `Constructor injection` ).
  - Đánh dấu thông qua annotation `@Parameter` trên các trường ( `Fields injection` ).
- **Test Suite** : kiểm thử với việc gom các test case hoặc test suite lại chạy cùng 1 lúc
  - `@RunWith (Suite.class)`
  - `@SuiteClasses({testClass1.class,...})`

## ❖ Các bước tạo và chạy một Junit test :

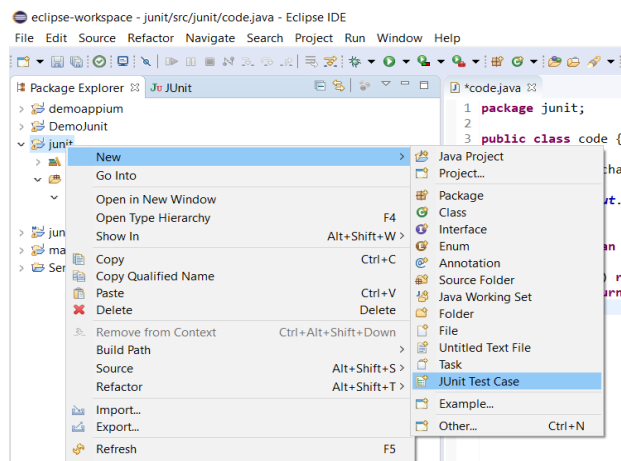
- Đầu tiên hãy khởi động eclipse, khi giao diện eclipse xuất hiện, chọn File → New → Java Project, đặt tên cho project, đặt tên theo ý muốn, ở đây em đặt junit, nhấn Finish

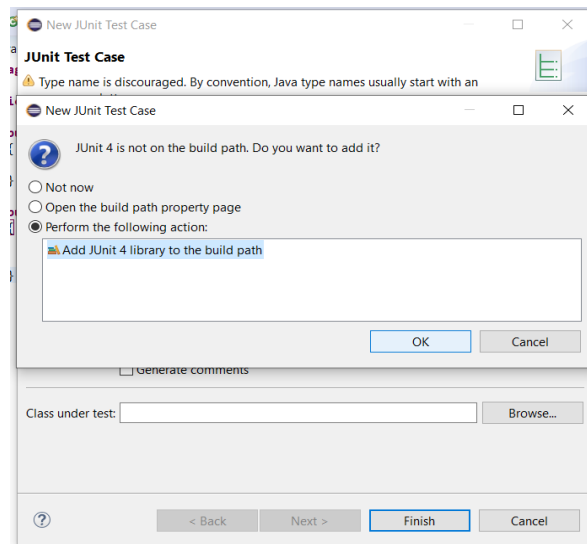
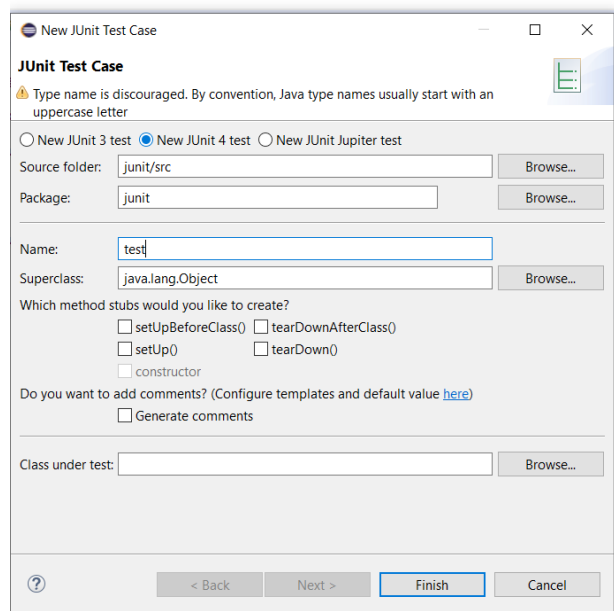


➤ Bắt đầu tạo một class để ghi các hàm muốn test



➤ Tạo một Junit Test Case để tiến hành ghi các hàm kiểm tra, làm theo các hướng dẫn bên dưới





Ngoài ra JUnit có khả năng kiểm thử một web app, cùng qua phần 3.1.2 để tìm hiểu sơ nét về khả năng này.

### 3.1.2. Giới thiệu về MAVEN :



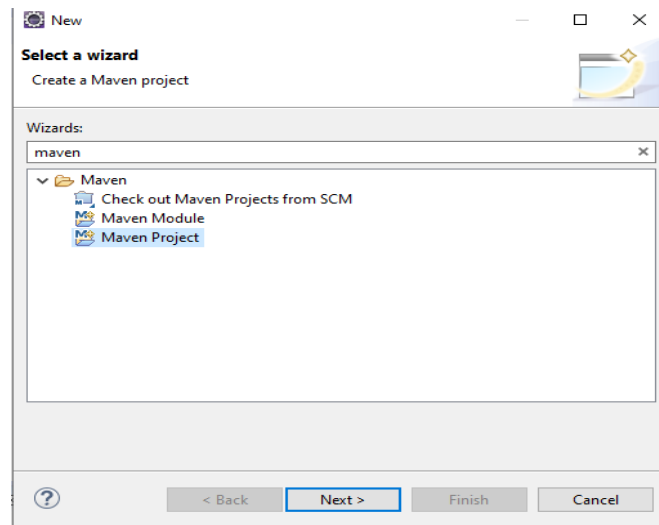
#### a. Tổng quan về Maven :

- Apache maven : là một chương trình quản lý dự án cho phép người phát triển có thể quản lý các version, các thư viện sử dụng trong dự án (dependencies), quản lý build, tự động tải về các Javadoc và source chương trình,...
- Repository : nơi để chương trình maven để download các dependency (thư viện dự án)
- Cấu hình file pom.xml : file pom.xml là nơi khai báo tất cả những gì liên quan đến dự án được cấu hình qua maven, chẳng hạn việc khai báo thư viện, phiên bản sử dụng của dự án, tên dự án, repository.

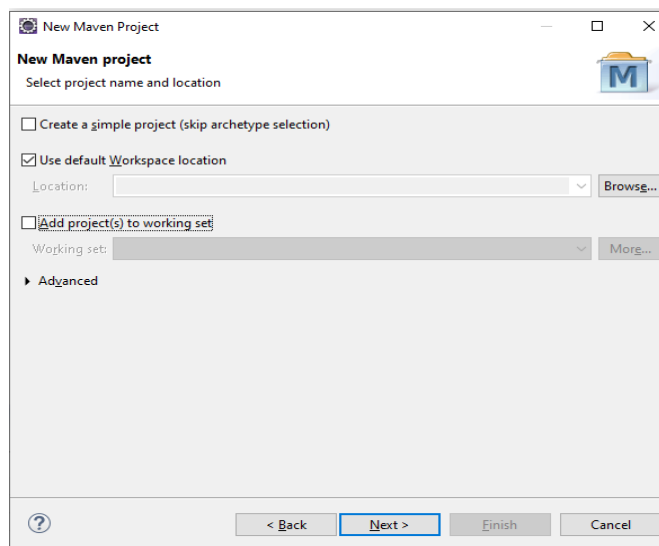
```
<?xml version="1.0" encoding="UTF-8" ?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.huythao</groupId>
  <artifactId>junitwebapp</artifactId>
  <packaging>war</packaging>
  <version>0.0.1-SNAPSHOT</version>
  <name>junitwebapp Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <finalName>junitwebapp</finalName>
  </build>
</project>
```

#### ❖ Các bước tạo một Maven :

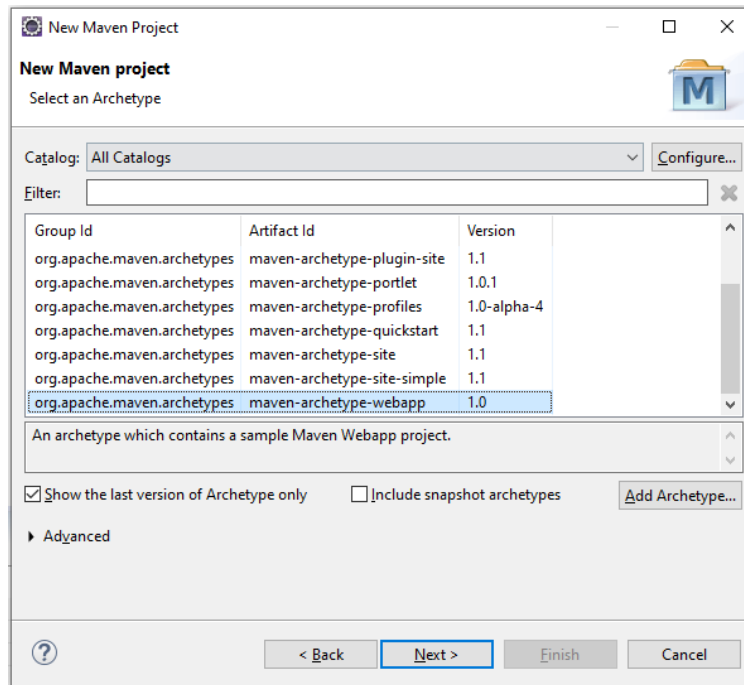




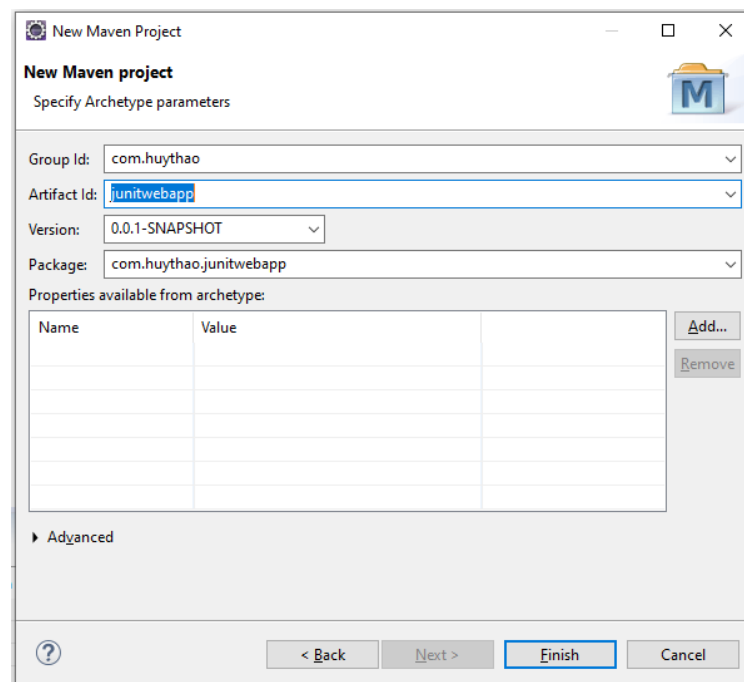
Tìm Maven Project ( hoặc muốn nhanh hãy nhấn tìm như trên ) và nhấn Next.



Đây là nơi bạn muốn lưu project, mặc định sẽ lưu tại workspace mà bạn đã tạo để lưu cái project. Nhấn next nếu bạn không muốn thay đổi gì cả.

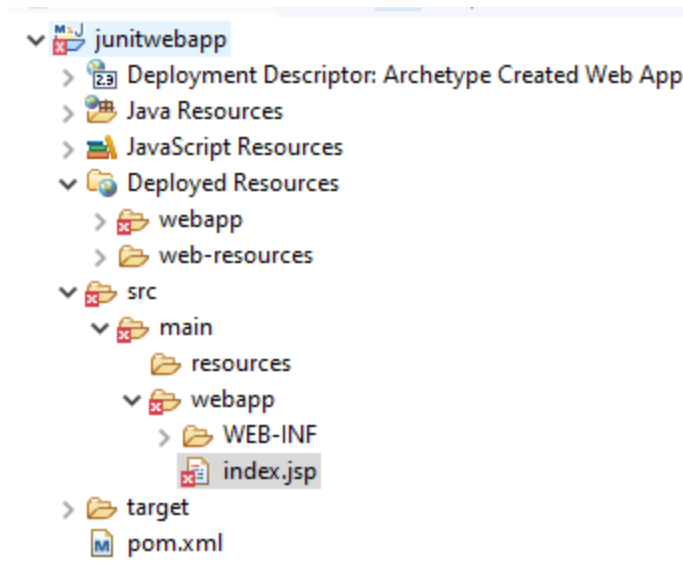


Tại đây bạn lướt xuống và tìm maven-archetype-webapp ( vì ở đây mình ví dụ về test web-app), sau đó nhấn Next .



Ở bước này là nơi bạn sẽ đặt tên cho project.

Sau khi thực hiện các bước sau sẽ ra được kết quả như dưới đây :



#### b. Ứng dụng JUnit trong test webApp :

- Nguyên nhân gây ra lỗi trên là vì chúng ta chưa khai báo đủ các thư viện cần thiết.
- Để fix được các lỗi trên chúng ta cấu hình thêm vào file pom.xml
- Các thư viện cần thêm :
  - jwebunit-htmlunit-plugin : plugin html cho Junit.
  - Junit : phiên bản 4.12
  - slf4j : Simple Logging Façade for java

```
<dependency>
    <groupId>net.sourceforge.jwebunit</groupId>
    <artifactId>jwebunit-htmlunit-plugin</artifactId>
    <version>3.3</version>
    <scope>test</scope>
</dependency>
```

```
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
</dependency>
```

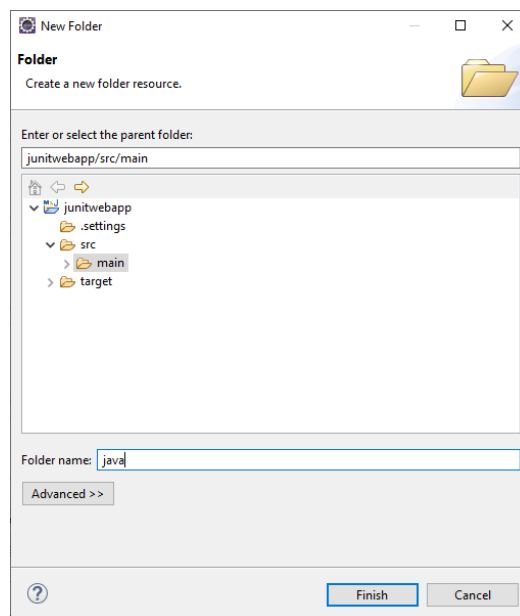
```
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-jdk14</artifactId>
```

```
<version>1.7.25</version>
</dependency>

<properties>
  <maven.compiler.source>1.8</maven.compiler.source>
  <maven.compiler.target>1.8</maven.compiler.target>
</properties>
```

Sau khi thêm vào file pom.xml, save file lại và các thư viện trên sẽ được tải về tự động.

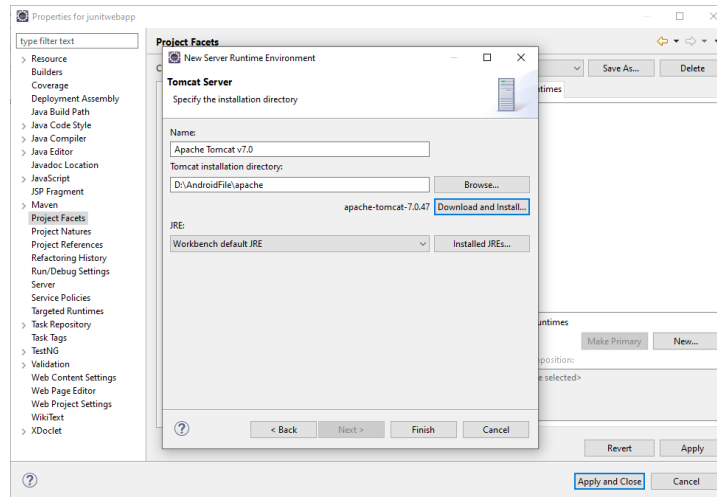
Lưu ý: cần tạo một file Folder có tên java vào đường dẫn như hình : (tên file maven/src/main)



Lưu ý : chúng ta cần phải có một ứng dụng để triển khai các ứng dụng Java Web. Ở đây chúng em sử dụng ứng dụng Apache Tomcat Server .

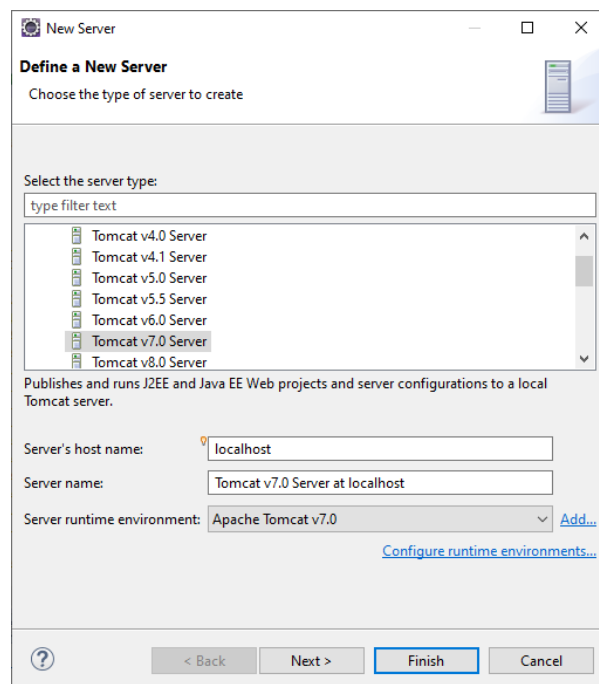
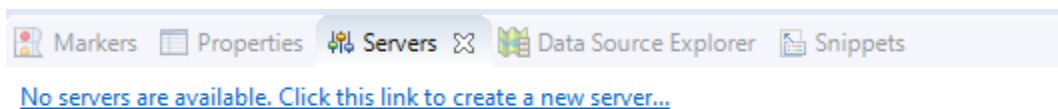
Giới thiệu sơ nét Apache Tomcat : là một Java Servlet được phát triển bởi Apache Software Foundation (ASF). Tomcat thi hành các ứng dụng Java Servlet và JavaServer Pages (JSP) từ Sun Microsystems, và cung cấp một máy chủ HTTP cho ngôn ngữ Java thuần túy để thực thi các chương trình viết bằng ngôn ngữ Java. ( tìm hiểu thêm tại : <https://ngockhuong.com/java/huong-dan-cai-dat-va-cau-hinh-server-apache-tomcat-tren-eclipse.html> )

Ở đây em sử dụng Apache Tomcat v7.0 Server.

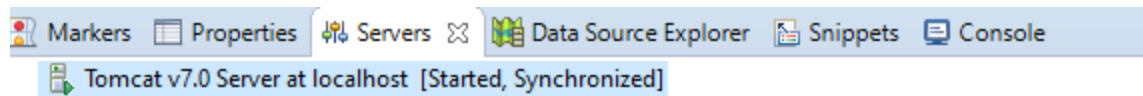


Lưu ý: lưu đường dẫn cho server tại nơi dễ tìm .

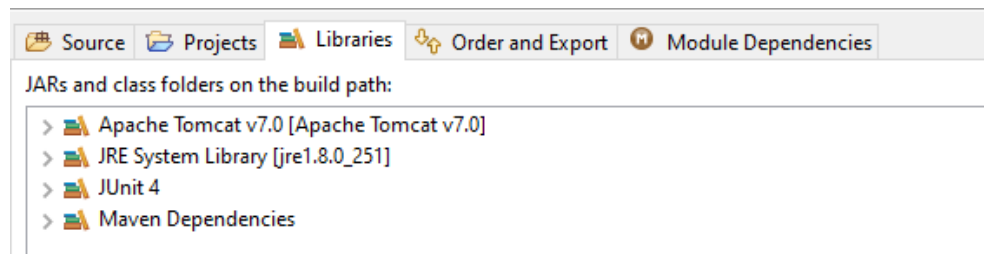
Sau đó mở hộp thoại Servers vào tạo 1 server chung cho junitwebapp.



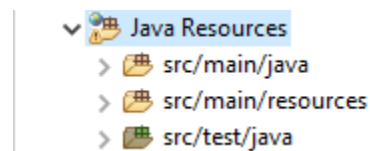
Lưu ý : ghi nhớ phải bật server mỗi lần mở lại eclipse và restart lại khi có sự thay đổi trong gói.



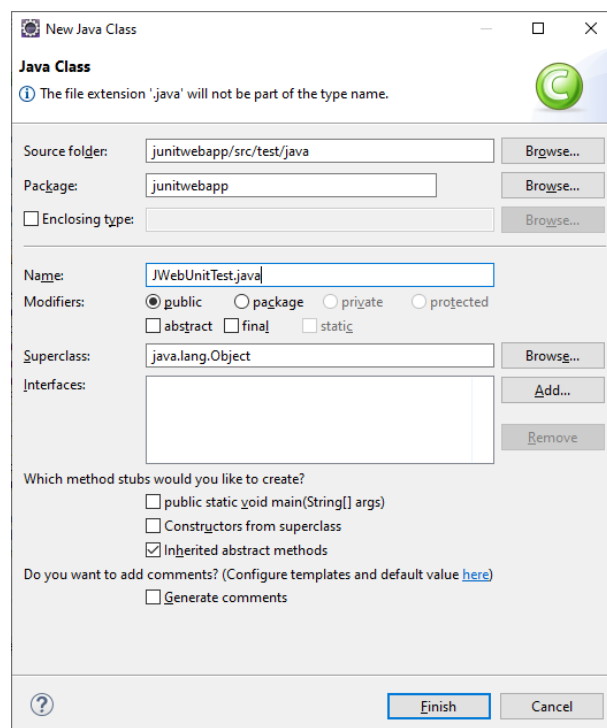
Thêm vào thư viện JUnit ( ở đây em sử dụng JUnit 4 ) vào thư viện chung của project.

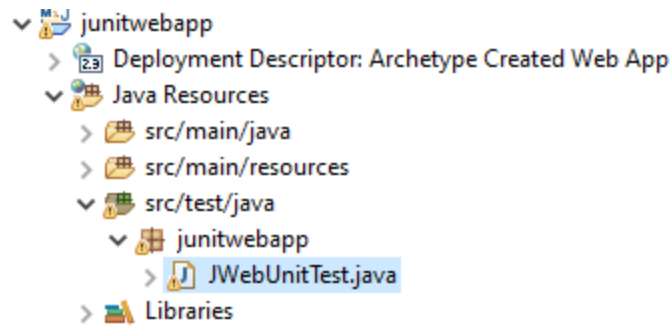


Sau khi thêm thư viện JUnit, tại src sẽ hiện ra src/test/java với hình package màu xanh lá.



Sau đó ta tạo một class.java trong đường dẫn trên.

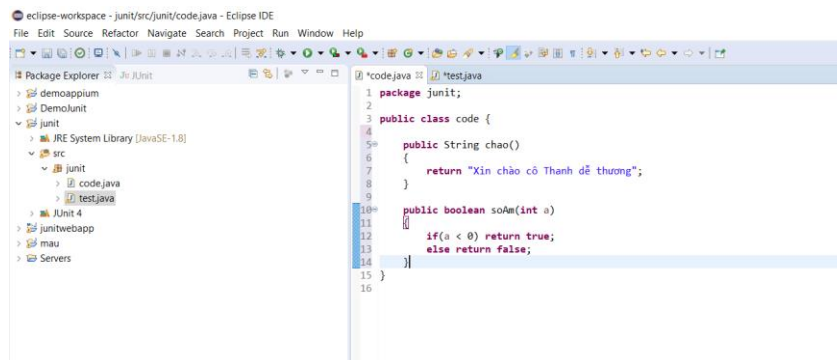




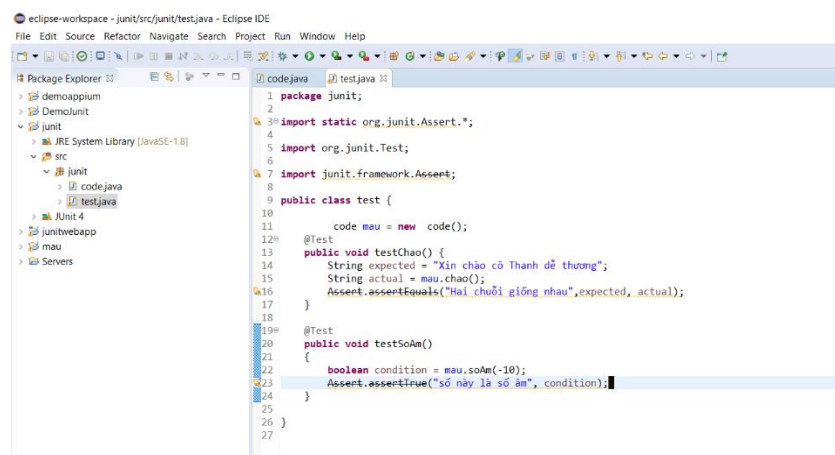
### 3.1.3. Demo viết Script và kiểm thử bằng công cụ JUnit4:

#### a. Kiểm thử phương thức cơ bản :

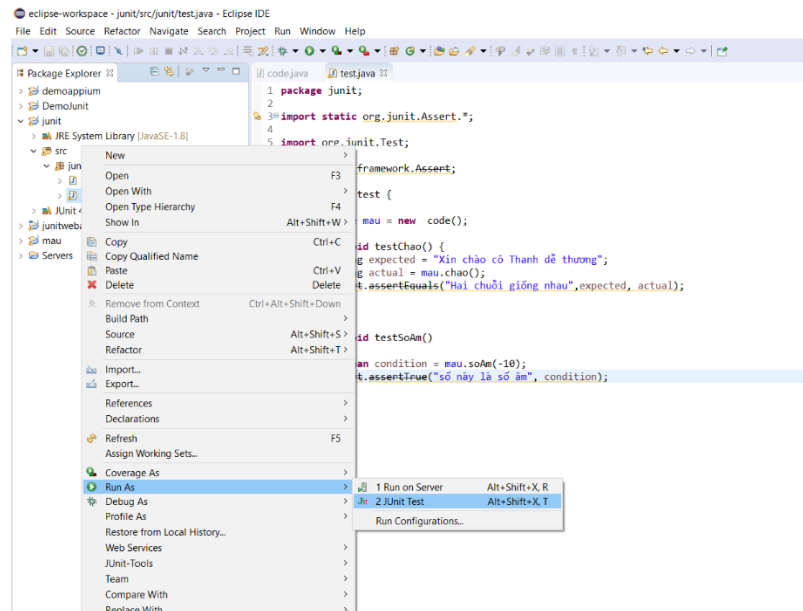
- Theo hướng dẫn ở trên các bạn tạo 1 class để ghi các hàm cần test



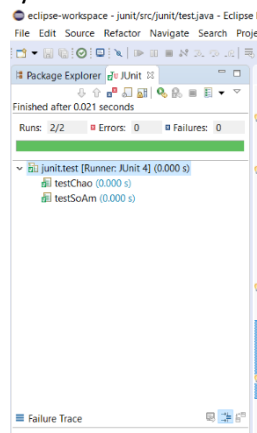
- Tạo 1 Junit Test Case và ghi các hàm test



➤ Tiến hành chạy Junit Test Case:



➤ Đây là Test Result trả về màu xanh nếu Pass và màu đỏ nếu Fail





## b. Kiểm thử Web App cơ bản :

Tạo thử nghiệm 3 trang web đơn giản để dễ tìm hiểu .

Trang 1 : index.jsp

```
1<html>
2<head>
3  <title>Login</title>
4  </head>
5<body>
6  <p>Welcome miss Thanh to login page</p>
7  <a href="home.jsp" id="home">Please click here to get to Home page</a>
8  </body>
9  </html>
10
```

Trang 2 : home.jsp

```
1<html>
2<head>
3  <title>Home</title>
4  </head>
5<body>
6  <h2>Welcome to Home Page!</h2>
7  <p>Welcome miss Thanh to the Home Page</p>
8  <li>
9  <a href="today.jsp" id="Today">Click here if you want to know what time is it now</a>
0  </body>
.
```

Trang 3 : today.jsp

```
1<html>
2  <head><title>Today</title></head>
3<body>
4<div>
5  <p>Today's date: <%= (new java.util.Date()).toLocaleString()%>
6  </p>
7  <li><a href="index.jsp" id="login">Please click here to get to Login page</a>
8  </body>
9  </html>
```

Đây là các thư viện cần có trong quá trình test 3 trang web trên

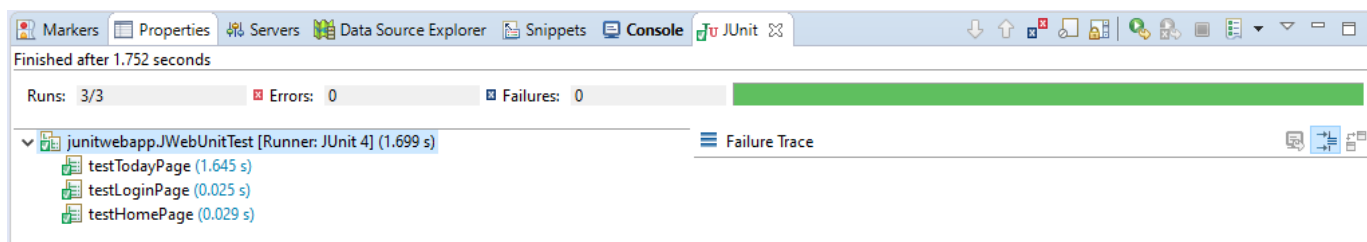
```
1 package junitwebapp;
2 import static net.sourceforge.jwebunit.junit.JWebUnit.assertLinkPresent;
3 import static net.sourceforge.jwebunit.junit.JWebUnit.assertTitleEquals;
4 import static net.sourceforge.jwebunit.junit.JWebUnit.beginAt;
5 import static net.sourceforge.jwebunit.junit.JWebUnit.clickLink;
6 import static net.sourceforge.jwebunit.junit.JWebUnit.setBaseUrl;
7 import static net.sourceforge.jwebunit.junit.JWebUnit.setTestingEngineKey;
8 import static org.junit.Assert.assertThat;
9
10 import org.junit.Before;
11 import org.junit.Test;
12
13 import net.sourceforge.jwebunit.util.TestingEngineRegistry;
```

```
public class JWebUnitTest {  
    @Before  
    public void prepare() {  
        setTestingEngineKey(TestingEngineRegistry.TESTING_ENGINE_HTMLUNIT);  
        setBaseUrl("http://localhost:8080/junitwebapp");  
    }  
}
```

Mục tiêu kiểm thử : Test các liên kết a href của từng trang .

```
@Test  
public void testLoginPage() {  
    beginAt("index.jsp");  
    assertEquals("Login");  
    assertLinkPresent("home");  
    clickLink("home");  
    assertEquals("Home");  
}  
@Test  
public void testHomePage() {  
    beginAt("home.jsp");  
    assertEquals("Home");  
    assertLinkPresent("today");  
    clickLink("today");  
    assertEquals("Today");  
}  
@Test  
public void testTodayPage() {  
    beginAt("today.jsp");  
    assertEquals("Today");  
    assertLinkPresent("login");  
    clickLink("login");  
    assertEquals("Login");  
}
```

Và đây là kết quả sau khi chạy Test Script kiểm thử các đường link trang web bằng công cụ JUnit 4.



### 3.2. Viết Script và Test bằng công cụ Appium :

#### 3.2.1. Giới thiệu về Appium :



- Appium là một công cụ mã nguồn mở được sử dụng để automation testing các native app, mobile web app và hybrid app trên nền tảng iOS và Android.
- Appium hỗ trợ “đa nền tảng” (cross-platform) cho phép bạn sử dụng API giống nhau để viết test cho các nền tảng khác nhau (iOS và Android). Điều này khá là tiện lợi khi bạn muốn sử dụng lại các test suites của mình.
- Appium hỗ trợ kiểm thử tự động phần mềm trên cả emulator, simulator và devices
- Appium được phát triển dựa trên các điểm sau: Không nên biên dịch lại app để tự động hóa nó Không nên bị khóa vào một ngôn ngữ hay một framework cụ thể Nên là mã nguồn mở Không cần thiết phải kiểm thử tự động khi đã là các API tự động hóa.

a. Các bước cài đặt :

➤ Bước 1 : Cài đặt Java JDK và Android SDK

- Java Runtime (JRE) : Chúng em sử dụng phiên bản jre1.8.0\_251 (jre-8u251-windows-x64 : <https://www.oracle.com/java/technologies/javase-jre8-downloads.html>).
- Java Development Kit (JDK) : Chúng em sử dụng phiên bản jdk-13.0.2 (jdk-13.0.2\_windows-x64\_bin : <https://www.oracle.com/java/technologies/javase-jdk13-downloads.html> ).
- Cài đặt Android SDK :

Android SDK : Android SDK (Software Development Kit) – hay còn gọi tắt là devkit, là tập hợp các công cụ phát triển phần mềm. SDK cho phép tạo ra các ứng

dụng cho một gói phần mềm, framework, nền tảng phần cứng, hệ thống máy tính, thiết bị video game, hệ điều hành, hay nền tảng phát triển nào đó tương tự. (<https://developer.android.com/studio>).

## Command line tools only

If you do not need Android Studio, you can download the basic Android command line tools below. You can use the included [sdkmanager](#) to download other SDK packages.

These tools are included in Android Studio.

Platform	SDK tools package	Size	SHA-256 checksum
Windows	<a href="#">commandlinetools-win-6514223_latest.zip</a>	82 MB	408cbc88f1a077521e20574ecc0a3c45d2d2a9e099011bb0493cb22779811cc3
Mac	<a href="#">commandlinetools-mac-6514223_latest.zip</a>	82 MB	d11cca1138d35852cc786ecb3cc26d0ce4fbfc718a6fb2f369d4bb21af230ce7
Linux	<a href="#">commandlinetools-linux-6514223_latest.zip</a>	82 MB	ef319a5afdb41822cb1c88d93bc7c23b0af4fc670abca89ff0346ee6688da797

➤ Bước 2 : Tải về các gói SDKmanager phù hợp : Sau khi cài đặt androidSDK, mở cmd và vào đường dẫn (nơi bạn lưu androidSDK)\tools\bin ( đường dẫn mặc định sau khi cài xong androidSDK)

```
D:\>cd D:\AndroidFile\sdk\tools\bin\${ANDROID_HOME}\tools\bin
D:\AndroidFile\sdk\tools\bin\${ANDROID_HOME}\tools\bin>
```

- ✓ Tại đây ta tải thêm cho hệ thống các build-tools, platform-tools và platforms; android phù hợp với phiên bản android của thiết bị di động của bạn, Chẳng hạn tải platform api 28 : sdkmanager "platform-tools" "platforms;android-28"

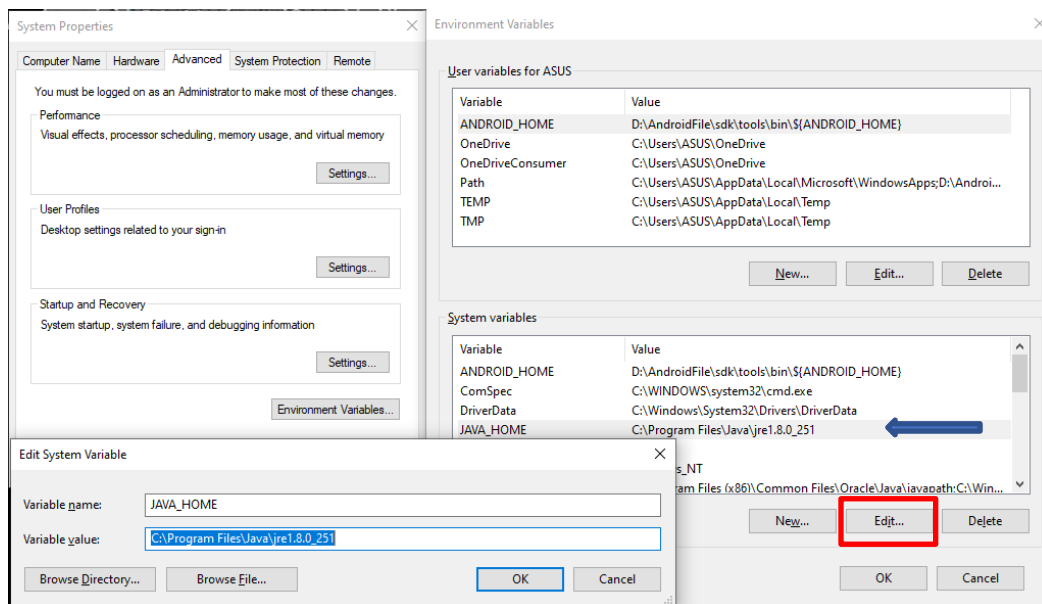
```
D:\AndroidFile\sdk\tools\bin\${ANDROID_HOME}\tools\bin>sdkmanager "platform-tools" "platforms;android-28"
Warning: File C:\Users\ASUS\.android\repositories.cfg could not be loaded.
[=====] 55% Unzipping... platform-tools\sysra
```

- ✓ Sử dụng lệnh sdkmanager –list để xem các phiên bản thư viện đã cài đặt :

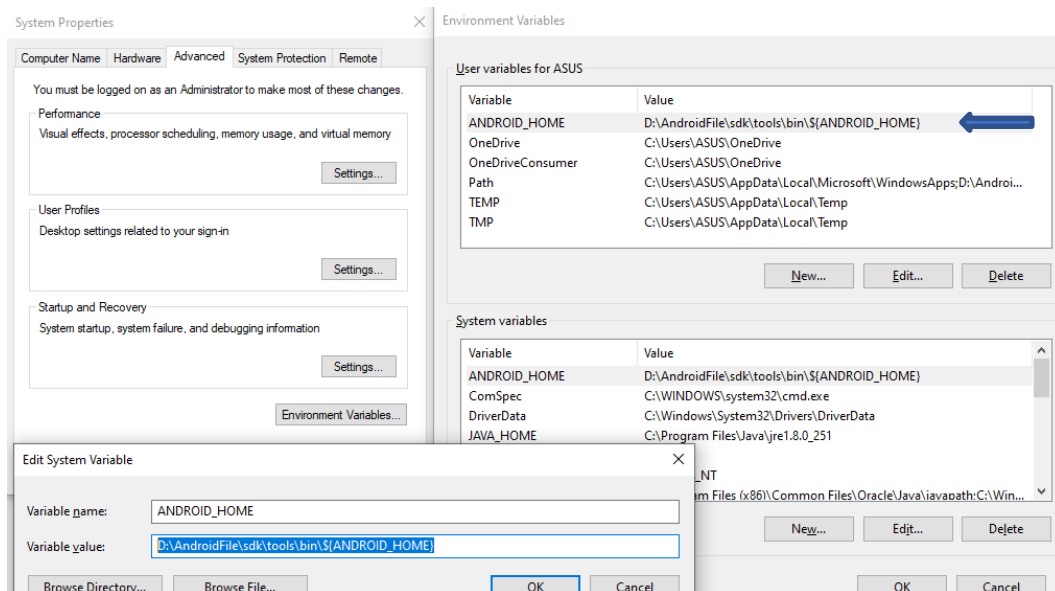
```
Installed packages:=====] 100% Computing updates...
Path | Version | Description | Location
-----|-----|-----|-----
build-tools;24.0.0 | 24.0.0 | Android SDK Build-Tools 24 | build-tools\24.0.0\
build-tools;25.0.0 | 25.0.0 | Android SDK Build-Tools 25 | build-tools\25.0.0\
build-tools;26.0.0 | 26.0.0 | Android SDK Build-Tools 26 | build-tools\26.0.0\
build-tools;28.0.0 | 28.0.0 | Android SDK Build-Tools 28 | build-tools\28.0.0\
emulator | 30.0.5 | Android Emulator | emulator\
patcher;v4 | 1 | SDK Patch Applier v4 | patcher\v4\
platform-tools | 30.0.0 | Android SDK Platform-Tools | platform-tools\
platforms;android-23 | 3 | Android SDK Platform 23 | platforms\android-23\
platforms;android-24 | 2 | Android SDK Platform 24 | platforms\android-24\
platforms;android-25 | 3 | Android SDK Platform 25 | platforms\android-25\
platforms;android-26 | 2 | Android SDK Platform 26 | platforms\android-26\
platforms;android-28 | 6 | Android SDK Platform 28 | platforms\android-28\
```

**Bước 3 : Cấu hình Enviroment Variables :**

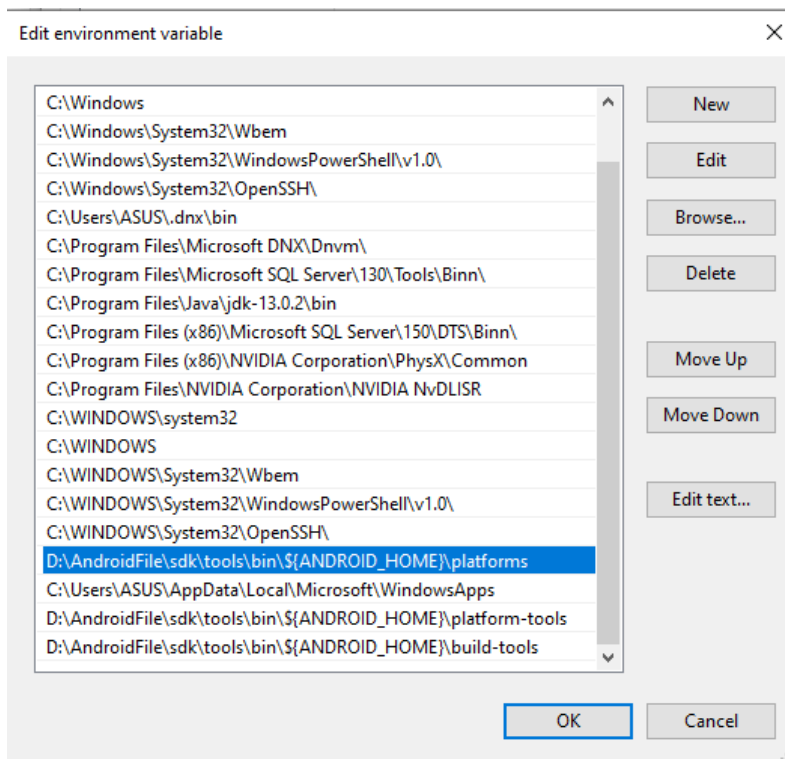
Tạo New và đặt tên Variable là JAVA\_HOME là đường dẫn của jre của bạn :



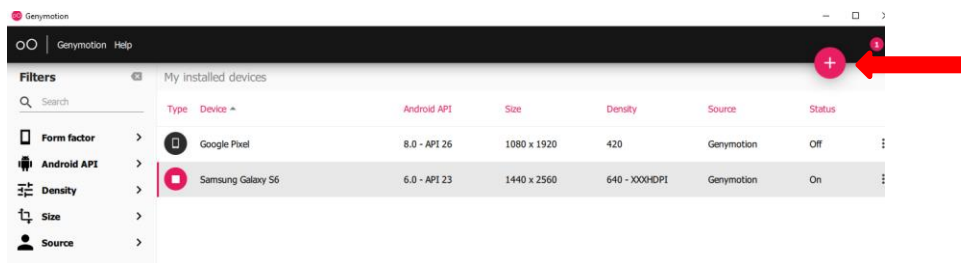
Tương tự đặt thêm ANDROID\_HOME là đường dẫn dẫn đến AndroidSDK:



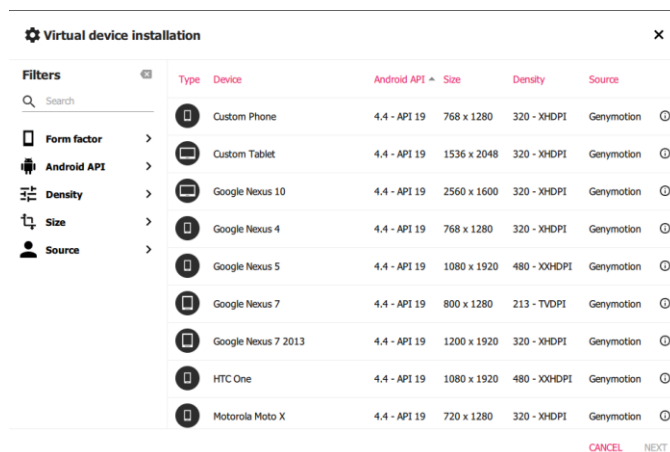
Trong phần Path tạo mới(nhấn vào PATH và chọn New) và đưa các đường dẫn các thư viện sdk cần dùng .



Bước 4 ( không bắt buộc ) : Cài đặt máy ảo bằng ứng dụng Gennymotion phiên bản 3.1.0 (<https://www.genymotion.com/download/>)



Tạo thêm thiết bị ảo bằng cách nhấn nút cộng màu hồng và lựa chọn thiết bị bạn muốn tải làm máy ảo



Bước 5 : Các câu lệnh cần thiết :

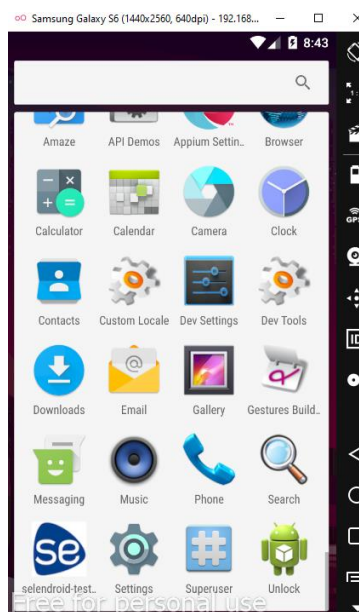
Câu lệnh xem tên của thiết bị di động được kết nối : adb devices

```
C:\Users\ASUS>adb devices
List of devices attached
LC66LYP07007    device
192.168.120.101:5555    device
```

Cài đặt một file apk trong máy tính vào điện thoại : adb install “nơi lưu trữ file apk”

Vd: cài đặt file selendroid-test-app-0.17.0 đường dẫn "D:\AndroidFile\selendroid-test-app-0.17.0.apk" vào thiết bị có tên 192.168.120.101:5555 (lưu ý mỗi lần tải vào chỉ có thể tải cho 1 thiết bị ).

```
C:\Users\ASUS>adb install "D:\AndroidFile\selendroid-test-app-0.17.0.apk"
Performing Push Install
D:\AndroidFile\selendroid-test-app-0.17.0.apk: 1 file pushed, 0 skipped. 661.0 MB/s (802079 bytes in 0.001s)
pkg: /data/local/tmp/selendroid-test-app-0.17.0.apk
Success
```



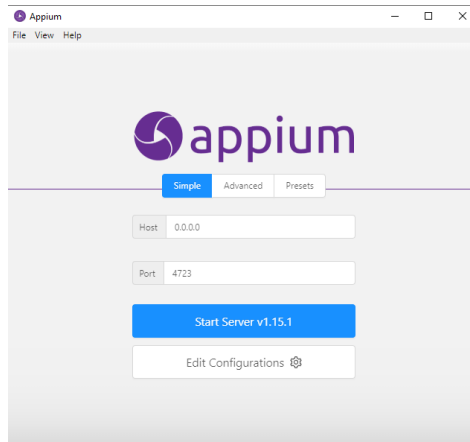
Bước 6 : Cài đặt Appium và Selenium :

Appium desktop : tải về [Appium-windows-1.15.1.exe](https://github.com/appium/appium-desktop/releases/tag/v1.15.1) phiên bản 1.15.1  
(<https://github.com/appium/appium-desktop/releases/tag/v1.15.1>)

Appium client : phiên bản 5.0.0-BETA8 ( <https://jar-download.com/artifacts/io.appium/java-client/5.0.0-BETA8/source-code> )

Selenium : phiên bản 3.4.0  
(<https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java/3.4.0>)

Lưu ý : Sau khi tải về Appium Client và Selenium, giải nén và thêm các file thư viện vào thư viện của project bạn tạo có chứa class testing Appium trong ứng dụng viết code ( ở đây em dùng viết code testing trên công cụ Eclipse (Version: 2019-09 R (4.13.0) Build id: 20190917-1200 ).



Màn hình chính ứng dụng Appium.

❖ Tải về TestNG :

- Phiên bản TestNG chúng em lựa chọn : 6.14.3 (TestNG - <https://dl.bintray.com/testng-team/testng-eclipse-release/6.14.3> ).
- Là một framework test, nó quản lý tạo một test suite / test case , thứ tự run test và report sau khi run test xong.
- Có nhiều loại annotation After và Before.
- TestNG sẽ cho phép bạn config để nhậ ra những packages / Classes / Methods mà bạn muốn hoặc không muốn run. Ngoài ra, nó còn thể nhóm những test Methods ở các Classes khác nhau thành 1 Group, và group này có thể run hoặc ko run tùy vào config.
- Để biết rõ hơn nhiều điều hay từ TestNG, xem thêm tại ( <https://giangtester.com/bai-1-testng-la-gi-va-nhung-thu-co-ban-cua-testng-ma-ban-co-the-chua-biet/>).

b. Một số các lệnh thường sử dụng :

- Find Elements ( findElementsByXXX):
  - AccessibilityID : Đọc định dạng ID cho UI element. Đối với Android, đây là content – desc thuộc tính của element
  - Class name : tên đầy đủ của lớp UIAutomator2 ( chẳng hạn android.widget.TextView ).
  - ID : phần định danh element, là phần resource – id cho android
  - Name : Tên của element



- XPath : tìm kiếm nguồn XML của ứng dụng bằng xpath ( tuy nhiên không nên sử dụng vì sẽ gây vấn đề về hiệu suất ).
- Image : xác định vị trí một phần tử bằng cách ghép nó với tệp hình ảnh được mã hóa ở base 64.
- Các Actions :
  - Click : hành động nhấp một đối tượng của element. Nếu element nằm ngoài khung viewport hoặc bị lỗi, element nào không thể tương tác được sẽ bị trả về.
  - Sendkeys : gửi một chuỗi kí tự vào nơi mà element chỉ đến .
  - Clear : xóa các giá trị của element.
- Attributes :
  - Text : Trả về chuỗi những text có trong element.
  - Name : trả về tag của element.
  - Selected : xác định xem một phần tử trong biểu mẫu có được chọn hay không.
  - Enable : xác định xem element đó có đang được kích hoạt.
  - Display : xác định xem element đó có đang được hiển thị.
  - Location : xác định vị trí của element trên trang hoặc trên screen.
  - Size : xác định pixel của element được chọn .
- Interactions :
  - Hành động Mouse click chủ yếu hay sử dụng nhất : hành động nhấp chuột tại các nút mà chuột đang muốn hướng tới. ( ngoài ra còn có hành động như : Move To, Double Click, Button Down, Button Up).
  - Một số cái kiểu dạng Touch như : Single Tap, Double Tap, Move, Touch Down, Touch Up, Long Press, Scroll, Flick )
- Bạn cũng có thể làm các hành động cho web như đóng, mở, sửa đổi tên cửa sổ window, các hành động trở lại hặc refresh trang web, ngoài ra bạn còn có thể xóa cookie của web mobile, cấu hình frame và excute).

### 3.2.2. Demo viết Script và kiểm thử bằng công cụ Appium :

Cú pháp mẫu về việc kết nối với một thiết bị di động :

```
public class Selendroid {
    AndroidDriver driver;

    @BeforeTest
    public void InitTest() throws MalformedURLException {
        DesiredCapabilities cab=new DesiredCapabilities();
        cab.setCapability("deviceName","LC66LYP07007");
        cab.setCapability("platformName", "Android");
        cab.setCapability("platformVersion", "5.0.2");
        driver.manage().timeouts().implicitlyWait(100, TimeUnit.SECONDS);
    }
}
```

```

@AfterTest
public void EndTest() throws InterruptedException {
    Thread.sleep(5000);
    driver.quit();
}

```

Ví dụ : Test thử việc tìm kiếm trên YouTube điện thoại

Các bước chuẩn bị :

```

@BeforeTest
public void InitTest() throws MalformedURLException {
    DesiredCapabilities cab=new DesiredCapabilities();
    cab.setCapability("deviceName", "LC66LYP07007");
    cab.setCapability("platformName", "Android");
    cab.setCapability("platformVersion", "5.0.2");
    cab.setCapability("appPackage", "com.google.android.youtube");
    cab.setCapability("appActivity", "com.google.android.apps.youtube.app.WatchWhileActivity");
    driver = new AndroidDriver(new URL("http://127.0.0.1:4723/wd/hub"), cab);
    driver.manage().timeouts().implicitlyWait(100, TimeUnit.SECONDS);
}

```

Mục tiêu kiểm thử :

1. Chọn nút Search.
2. Trong khung tìm kiếm của Youtube gõ vào “cruel summer”.
3. Click vào nội dung xuất hiện đầu tiên.

```

@Test
public void TestClick() {

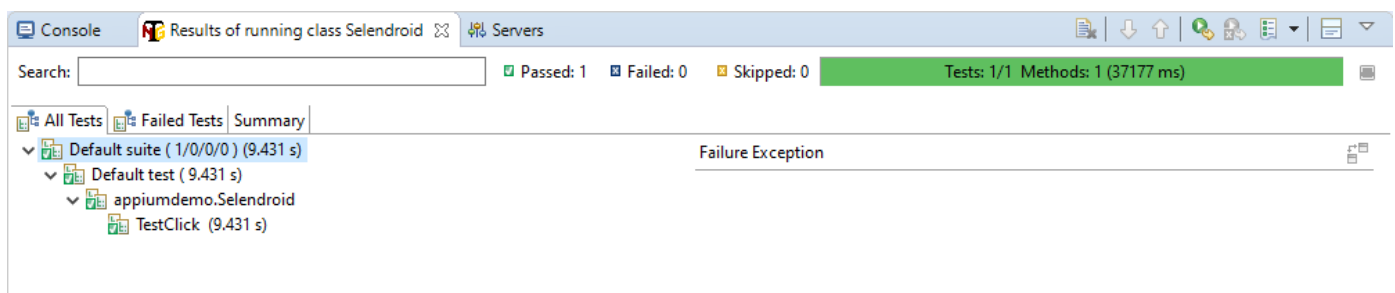
    WebElement chon = driver.findElementByAccessibilityId("Search");
    chon.click();

    WebElement nhap = driver.findElementById("com.google.android.youtube:id/search_edit_text");
    nhap.sendKeys("cruel summer");
    String elText = nhap.getText();
    System.out.println("/n"+elText);

    WebElement enter = driver.findElementById("com.google.android.youtube:id/text");
    enter.click();
}

```

Kết quả trả về :



## TỔNG KẾT

Kết quả đạt được :

Thông qua việc nghiên cứu đề tài, chúng em đã học được rất nhiều kiến thức mới trong việc kiểm thử, nó giúp em có thể hiểu rõ việc hoàn chỉnh một sản phẩm lập trình hợp lý.

Qua việc nghiên cứu sử dụng Script để kiểm thử cho chúng em điều kiện để chia sẻ kinh nghiệm, trao dồi kỹ năng cho bản thân, nhất là trong kiểm thử chương trình. Về test automation việc cài đặt cũng như kiểm thử bằng junit và Appium.. Chúng em hứa sẽ sử dụng các kiến thức đã học trong quá trình học để vận dụng và phát triển bản thân hơn.

Hạn chế của đề tài :

Do kiến thức vẫn còn nhiều hạn chế nên bài thuyết trình của chúng em có thể chưa hoàn chỉnh. Một số khái niệm, ví dụ có thể vẫn chưa ổn, nên chúng em mong cô có thể bỏ qua.

Lời cuối, chúng em chúc cô luôn giữ thật nhiều sức khỏe, gặp thật nhiều may mắn trên con đường sự nghiệp và luôn vui vẻ hạnh phúc bên gia đình.