

Getting started with Visual Testing using Applitools

The Test Automation Pyramid is not a new concept. While Automation helps validate functionality of your product, the look & feel / user-experience (UX) validation is still mostly manual.

With everyone wanting to be Agile, doing quick releases, this look & feel / UX validation becomes the bottleneck, and also is a very error-prone activity which causes brand, revenue and leads diluting your user-base.

In this workshop, we will learn how to extend Functional Automation implemented using Selenium WebDriver tests, or Appium tests to include Visual Validation using Applitools Eyes. Here are the details:

- Quick introduction to the What, Why of Visual Testing
- Glance through the capabilities of Applitools
- Write a small test for web / Android app
- Integrate Applitools with the above test to accomplish visual testing
- Fail the test if any Visual differences are found – this will ensure your CI builds will also fail if there are any visual differences found while running functional automation

Prerequisites:

- Bring your own laptop with software setup required to run Web / Mobile tests
- Example setup:
 - JDK 1.8
 - IDE – IntelliJ Idea / Eclipse /
 - If interested in Mobile automation integration, all requisite setup for running Appium tests (node, Android-SDK with relevant build-tools, platform-tools, etc., Appium, appium-doctor). Also, setup your emulators, or get device with cable
 - For web, ensure you have the right version of ChromeDriver available on your machine
 - Download Applitools dependency by adding the following lines in your maven's pom.xml

```
<!-- This is the Applitools Selenium Java SDK -->
<dependency>
  <groupId>com.applitools</groupId>
  <artifactId>eyes-selenium-java3</artifactId>
  <version>RELEASE</version>
</dependency>
```

Takeaways:

- Importance of Visual Testing
- Techniques to include Visual Testing part of your Testing Strategy

- Different options for automating Visual Testing
- Learn how to integrate Applitools in your Functional Automation to automate Visual Testing

Machine setup instructions

Please run the following commands on your laptops to ensure connectivity to the Applitools server.

The response status code for each of these methods should be 2xx / 3xx.

Instructions for Windows OS:

Run the following commands in PowerShell window and note the response status code:

- `curl -Method GET https://eyes.applitools.com`
- `curl -Method GET https://eyesapi.applitools.com`

If they get an error in the console / terminal window with message such as FORBIDDEN / ACCESS DENIED / PROXY ERROR / etc., then they should try the same commands by providing the proxy details:

NOTE: Based on your network configuration, the -ProxyCredential parameter may need to be specified

- `curl -Method GET -Proxy <proxyUrl> -ProxyCredential <proxyCredential> https://eyes.applitools.com`
- `curl -Method GET -Proxy <proxyUrl> -ProxyCredential <proxyCredential> https://eyesapi.applitools.com`

Instructions for Linux / OSX OS:

Run the following commands in PowerShell window and note the response status code:

- `curl -I https://eyes.applitools.com`
- `curl -I https://eyesapi.applitools.com`

If they get an error in the console / terminal window with message such as FORBIDDEN / ACCESS DENIED / PROXY ERROR / etc., then they should try the same commands by providing the proxy details:

NOTE: Based on your network configuration, the --U parameter may need to be specified

- `curl -I -p <proxyUrl> -U <user:password> https://eyes.applitools.com`
- `curl -I -p <proxyUrl> -U <user:password> https://eyesapi.applitools.com`

If you are still getting an error response, then you will need to get the following URLs whitelisted on your network:

- <https://render-wus.applitools.com>

- <https://eyesapi.applitools.com>
 - <https://eyes.applitools.com>
 - <https://eyespublicwusi0.blob.core.windows.net>
-

Machine Setup:

After you have ensured connectivity from your laptop to the Applitools server, follow the below steps to get your machine setup ready:

These steps are for Selenium-Java based Test Automation. If you are using any other combination, please contact anand.bagmar@applitools.com with specific details.

1. Install JDK 1.8 or JDK 11
2. Based on the browser of your choice, download the corresponding browser driver for WebDriver
3. Clone this git repo (<https://github.com/anandbagmar/getting-started-with-visualtesting>) on your laptop
4. Open the cloned project in your IDE as a Maven project. This will automatically download all the dependencies
5. Once all dependencies are downloaded, run the `Selenium_HelloWorld_Base` test directly from the IDE