

NobleProg



The World's Local Training Provider

NobleProg

NobleProg

Selenium and Katalon Studio

Trainer - Saurabh Dhingra

The World's Local Training Provider

NobleProg® Limited 2017
All Rights Reserved

A faint, dotted world map serves as the background for the slide, centered behind the main text.

Let's have some social agreements

1. Let us know each other!
2. Breaktime

Saurabh Dhingra

- 10+ years of Manual and Automation Testing experience
- 4+ years of DevOps experience
- Worked with multiple MNCs and Startups as Test Automation and DevOps Consultant
- 6+ years of Teaching Experience
- Have taken more 100 live batches
- Have taught more than 7000 professionals
- Author of QA Tech Hub website!



Akanksha & Joy Photo Studios

Agenda (Day 1)

1. Introduction to Automation Testing
2. Introduction to Selenium and its components
3. Project setup
4. Working with multiple browsers - Chrome, Edge
5. Identifying web elements - Locators
6. Mastering Xpaths
7. Working with form elements - textbox, dropdown(select), button, checkbox, radio buttons
8. Handling multiple windows, alerts, and iframes

Agenda (Day 1)

- 9. Mouse handling
- 10. Waits in Selenium
- 11. Taking screenshots
- 12. Generating reports
- 13. Parallel Testing
- 14. Troubleshooting - Debugging code



Module 1 - Introduction to Automation Testing

NobleProg® Limited 2017
All Rights Reserved

NobleProg

What do you mean by software testing?

Software testing is the process of investigating an application and finding errors in it.

The difference between testing and simply exploring is that testing involves comparing the application output to an expected standard and determining whether the application functions as expected.



Types of Testing

1. Manual Testing
2. Automation Testing

Manual Testing

Manual Testing is a type of Software Testing where QA's manually execute test cases without using any automation tools.

Manual Testing is the most primitive of all testing types and helps find bugs in the software system.

Any new application must be manually tested before its testing can be automated. Manual Testing requires more effort but is necessary to check automation feasibility.

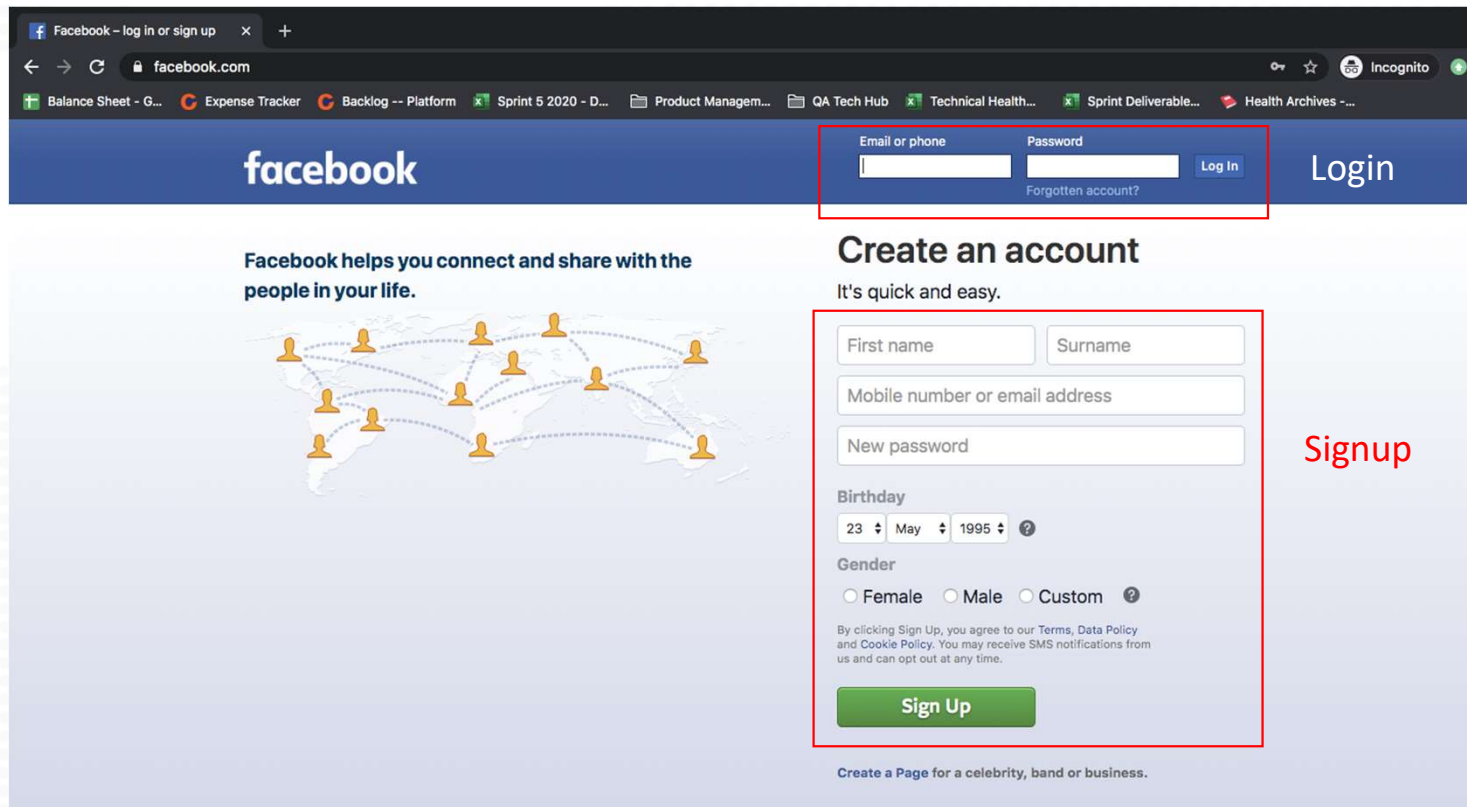
Automation Testing

Automated testing is the automatic execution of software testing by a special program with little or no human interaction. Automated execution guarantees that no test action will be skipped; it relieves testers of having to repeat the same boring steps over and over.

Basic test sequence

1. Understanding the functionalities
2. Defining the test data (input data)
3. Defining the expected output.
4. Performing test actions (feeding the appropriate input).
5. Gathering the application output and comparing it to expected result (baseline data).
6. Notifying developers or managers if the comparison fails.

How manual testing works?



The image shows a screenshot of the Facebook website's login and sign-up interface. The browser's address bar shows 'facebook.com'. The page has a dark blue header with the Facebook logo on the left. On the right side of the header, there is a login section with a red box around it. This section contains two input fields: 'Email or phone' and 'Password', followed by a 'Log In' button and a link for 'Forgotten account?'. Below the header, the main content area is divided into two columns. The left column features the text 'Facebook helps you connect and share with the people in your life.' above a graphic of a world map with user avatars connected by lines. The right column is titled 'Create an account' and 'It's quick and easy.' Below this, there is a sign-up form with a red box around it. The form includes input fields for 'First name', 'Surname', 'Mobile number or email address', and 'New password'. It also has a 'Birthday' section with dropdowns for day (23), month (May), and year (1995), and a 'Gender' section with radio buttons for 'Female', 'Male', and 'Custom'. A green 'Sign Up' button is at the bottom of the form. To the right of the sign-up form, the word 'Signup' is written in red. At the bottom of the page, there is a link that says 'Create a Page for a celebrity, band or business.'

facebook

Email or phone Password Log In
Forgotten account?

Facebook helps you connect and share with the people in your life.

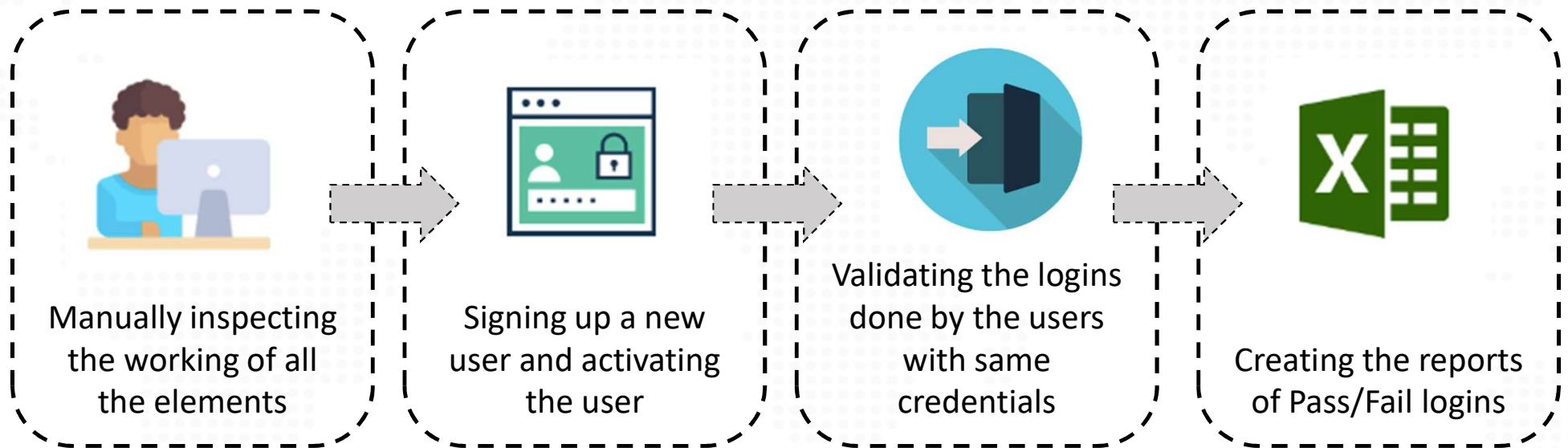
Create an account
It's quick and easy.

First name Surname
Mobile number or email address
New password
Birthday
23 May 1995
Gender
Female Male Custom
By clicking Sign Up, you agree to our Terms, Data Policy and Cookie Policy. You may receive SMS notifications from us and can opt out at any time.
Sign Up

Signup

Create a Page for a celebrity, band or business.

How manual testing works?

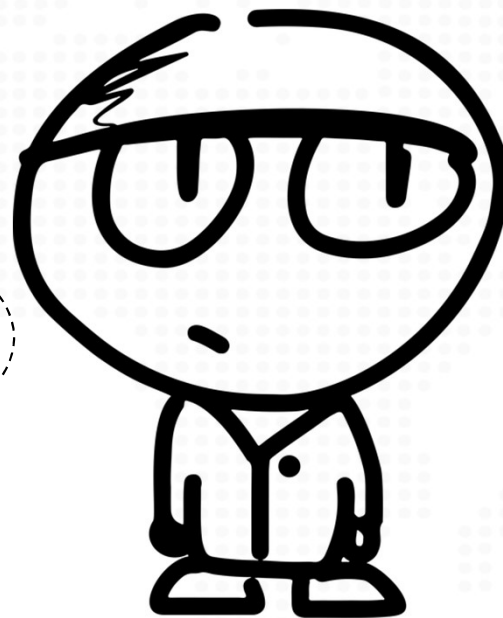


Challenges In Manual Testing

Time
Consuming

Boring

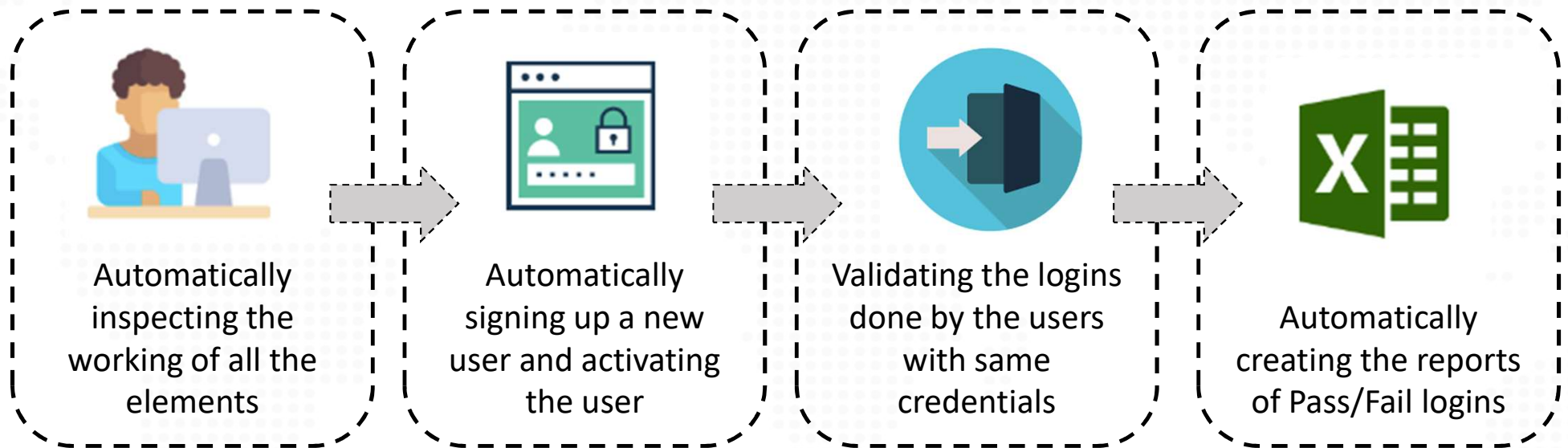
Creating
Manual reports



Tedious

Mistakes,
Errors..

How automated testing works?



How automation testing is better?

	Manual Testing	Automation Testing
Process	Time Consuming	Fast Execution
Dependency on humans	Huge Dependency	Less Dependency
Accuracy	Not always accurate	Accurate
Regression Testing	Not good for regression testing	Good for regression testing
Execution	Once or twice execution	Frequent execution
UI Testing	Useful for UI testing	Not useful for UI testing
Managing results with large data	Difficult	Easy
Executing scripts	Manually	Automated execution

Benefits of automation testing

1. Cost Reduction
2. Reusability
3. Fast Execution
4. Unattended Execution
5. Reliability

Challenges in Automation Testing

1. Synchronization between application under test and automation tool
2. Effective communication and collaboration between team members
3. Selecting a right tool
4. High upfront investment cost
5. Skilled people required.
6. Estimation in Test Automation

Introduction to Selenium

Selenium automates browsers. That's it!

What you do with that power is entirely up to you.

Primarily it is for automating web applications for testing purposes but is certainly not limited to just that.

Boring web-based administration tasks can (and should) also be automated as well.

Introduction to Selenium



Selenium is an open-source automation tool that is free to use by anyone. No cost involved.



Selenium works with most of the famous programming languages — Java, Python, C#, Ruby, Javascript.



Selenium is a platform-independent tool supports all the operating system — Windows, Linux, Mac



Selenium supports all the major vendors of the browsers — Mozilla Firefox, Google Chrome, Opera, Safari, etc.



A large community to help developers

Selenium and its compoents



SELENIUM RC
(REMOTE CONTROL)



SELENIUM IDE

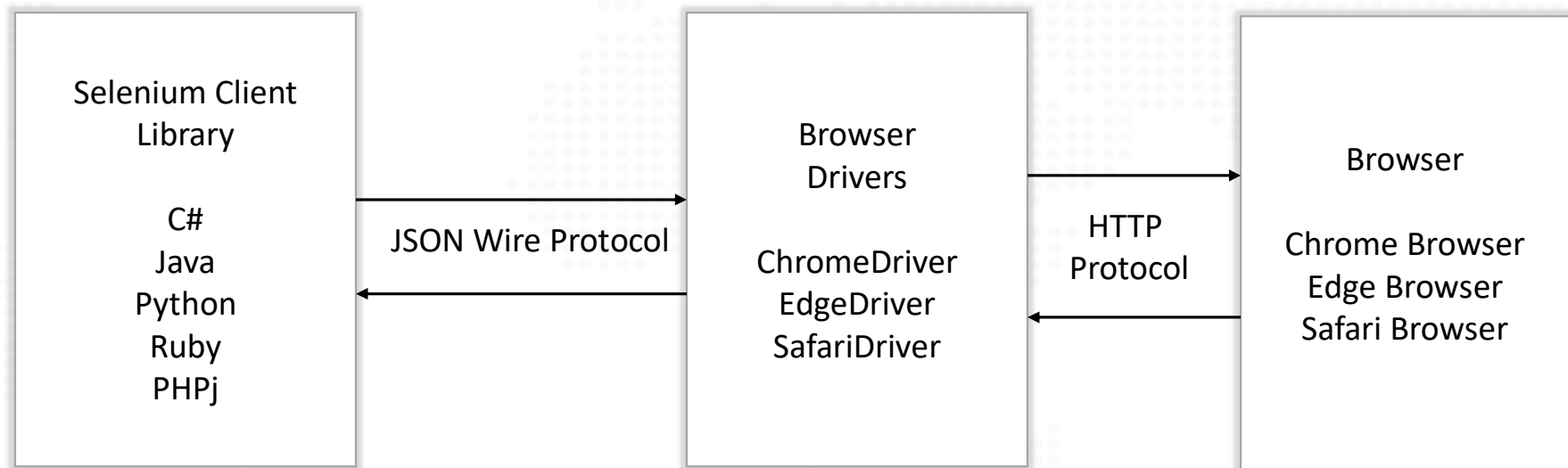


SELENIUM
WEBDRIVER



SELENIUM GRID

Architecture of Selenium WebDriver



Project Setup

1. Download and install Java (JDK and JRE)
2. Download and install Eclipse IDE
3. Create a maven project
4. Add Selenium and TestNG dependencies

Tasks

1. Invoke Browser (Chrome, Edge or of your choice)
2. Interacting with form elements - Textbox, Radio button, checkbox, dropdown, buttons
3. Mouse operation
4. Handling multiple windows, alerts and iframes

Locators

ID

Name

Classname

Tagname

Linktext

Partial Linktext

Xpath

Css Selector

XPath

XPath is the path written using html tags(or XML tags) and their attributes to reach to a node (or web element) in an HTML page or XML page.

Types of XPath -

1. Absolute XPath
2. Relative XPath

Absolute XPath

Absolute XPath starts from the beginning of the page.

As in an html page, the first tag is HTML, so all absolute XPaths always start with an html tag and then access immediate child to reach to a node.

“/” is used to access an immediate child of the parent tag.

Example: `html/body/table/tbody/tr[2]/td/input`

Relative XPath

Relative XPath starts from anywhere on the page.
“//” is used to access any child of the parent tag.

Basic Syntax -

`//htmlTagname[@attribute='value']`

Example:

`//input[@type='text']` - It represents XPath of a Web Element which is represented by an input tag and has an attribute type = 'text'

Which is better?

- Absolute XPath are fast, but they have one disadvantage that if there is an addition or deletion of some nodes in between they fail to work.
- In such a situation relative XPath are much better, and one should be good at writing dynamic XPath.
- There are some methods, operators, and axes available in XPath which can help to locate elements uniquely.

How to identify dynamic Web Element?

- Dynamic Web Elements are one which changes dynamically like - their attributes such as Id or Class name, or text associated change.
- The best possible way to identify dynamic elements is:
- To first search for another web element which is stable and can be identified uniquely
- Then use methods, operators, or axes to reach to the desired node (or web element).

Methods in XPath

1. text()
2. contains()
3. starts-with()

Text Method

text() -

text() method is used when we are searching for web elements with exact text.

Example - `//a[text()='Mobile & Accessories']`

Contains Method

Contains is a method used when the value of any attribute changes dynamically. It can search an element with partial information.

Example - `//input[contains(@type='submit')]`

Contains method also works with `text()` method as an argument to the `contains()` method in XPath.

Example - `//a[contains(text(), 'Mobile & Accessories')]`

Starts-with Method

Starts-with() method is used when we are searching for web elements matching the start of the text of the attributes passed.

text() method can also be used which will match the starting of the text.

Example: `//a[starts-with(text(),'Mobile')]`

Operators in XPath

AND Operator -

AND operator, when applied to multiple attributes, identifies a web element only when all the attributes are pointing to that element.

Example: `//input[@type='text' and @name='uid']`

OR Operator -

OR operator, when applied to multiple attributes, defines a web element only when any one of the attributes points to that element.

Example: `//input[@type='text' or @name='uid']`

Axes in XPath

Axis in xpath points the xpath processor to the direction in which it should navigate in the hierarchy of the html nodes.

Basic Syntax:

```
//html_tag[@attribute='value']//axes::html_tag
```

Frequently used axis in XPath

Ancestor - selects ancestor of the context node.

Following - select the elements which follow after the context node.

Preceding - choose the elements which precede before the context node.

Following-Sibling - choose the sibling coming after the context node.

Preceding-Sibling - choose the sibling coming before the context node.

Parent - contains the parent of the context node.

CSS Selector

Agenda –

- What is CSS Selector locator?
- Basic Syntax?
- Operators in CSS Selectors?

CSS Selector

CSS is "Cascading Style Sheets" and it is defined to display HTML in structured and colorful styles are applied to webpage.

Selectors are patterns that match against elements in a tree, and as such form one of several technologies that can be used to select nodes in an HTML/XML document.

CSS is much faster and simpler than the XPath.

Basic Syntax -

`htmltag[attribute='value']`

Operators in Css Selector

Using class in css

.(dot) operator is used in classes

.small.cbx.btn.btn-s.btn-ter.tab.tgl_button.center_b

Using Id:

(hash) operator is used

#ListViewInner -- example using Id

CSS Selector

- ^ (power symbol) -- its like starts with
 - starting of the string`input[id^=ema]`
- \$ (dolar symbol) -- its like ends with
 - matches ending text
- * (astrick) -- its like contains method
 - matches some part of the string

Best Practices..

CAPTCHA, short for Completely Automated Public Turing test to tell Computers and Humans Apart, is explicitly designed to prevent automation, so do not try!

There are two primary strategies to get around CAPTCHA checks

-
- 1. Disable CAPTCHAs in your test environment
- 2. Add a hook to allow tests to bypass the CAPTCHA

Best Practices..

File Downloads --

Whilst it is possible to start a download by clicking a link with a browser under Selenium's control, the API does not expose download progress, making it less than ideal for testing downloaded files.

This is because downloading files is not considered an important aspect of emulating user interaction with the web platform. Instead, find the link using Selenium (and any required cookies) and pass it to a HTTP request library like [libcurl](#).

Best Practices..

Gmail, email and Facebook logins -

For multiple reasons, logging into sites like Gmail and Facebook using WebDriver is not recommended. Aside from being against the usage terms for these sites (where you risk having the account shut down), it is slow and unreliable.

The ideal practice is to use the APIs that email providers offer, or in the case of Facebook the developer tools service which exposes an API for creating test accounts, friends and so forth.

Although using an API might seem like a bit of extra hard work, you will be paid back in speed, reliability, and stability. The API is also unlikely to change, whereas webpages and HTML locators change often and require you to update your test framework.

Best Practices..

Test dependency

A common idea and misconception about automated testing is regarding a specific test order. Your tests should be able to run in any order, and not rely on other tests to complete in order to be successful.

Best Practices..

Performance testing -

Performance testing using Selenium and WebDriver is generally not advised. Not because it is incapable, but because it is not optimised for the job and you are unlikely to get good results.

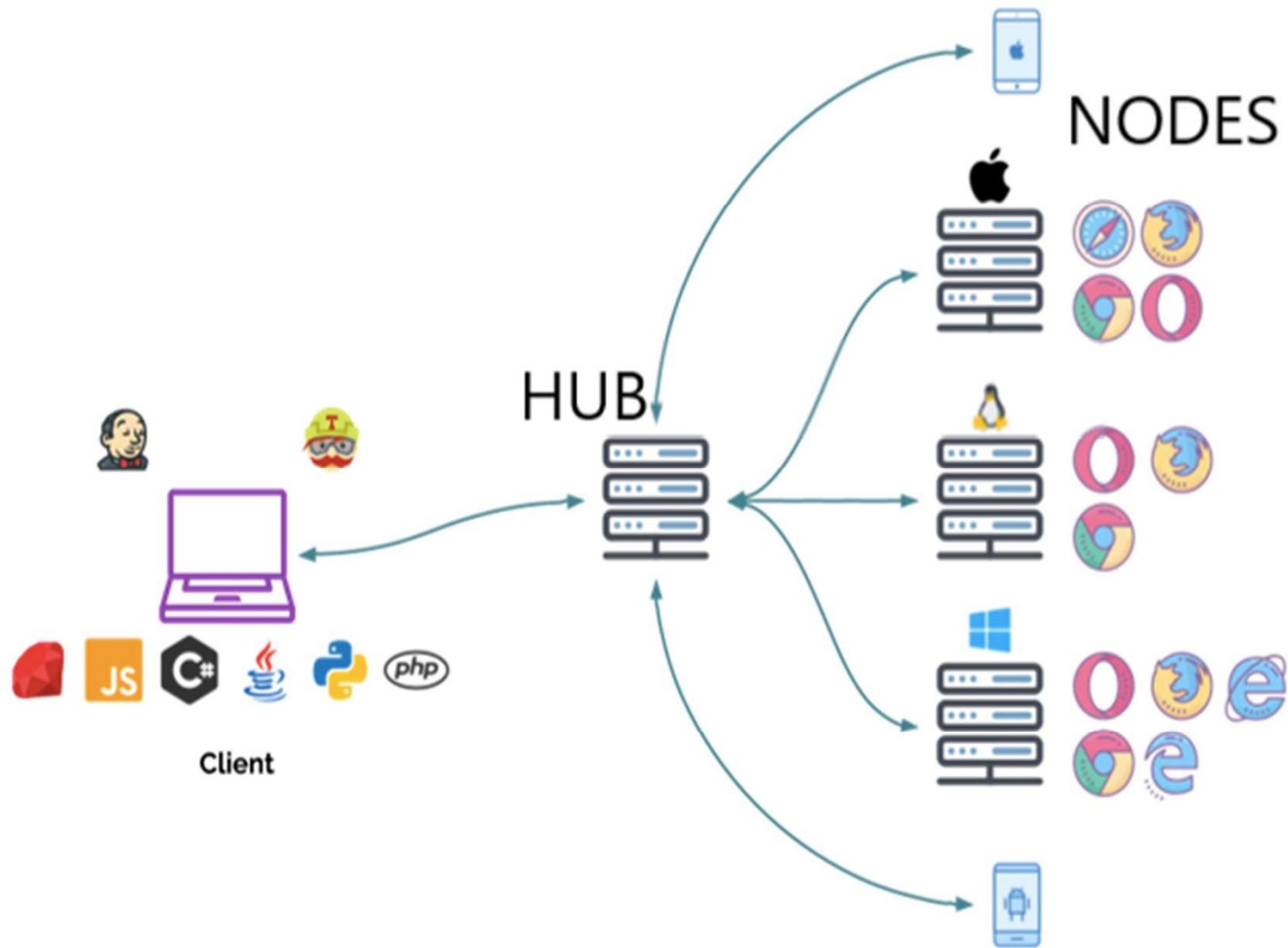
It may seem ideal to performance test in the context of the user but a suite of WebDriver tests are subjected to many points of external and internal fragility which are beyond your control;

for example browser startup speed, speed of HTTP servers, response of third party servers that host JavaScript or CSS, and the instrumentation penalty of the WebDriver implementation itself. Variation at these points will cause variation in your results.

Selenium Grid

Selenium Grid is a smart proxy server that allows Selenium tests to route commands to remote web browser instances. Its aim is to provide an easy way to run tests in parallel on multiple machines.

With Selenium Grid, one server acts as the hub that routes JSON formatted test commands to one or more registered Grid nodes. Tests contact the hub to obtain access to remote browser instances. The hub has a list of registered servers that it provides access to, and allows control of these instances.



Starting Selenium Hub

Navigate to the path where Selenium jar file is placed.

```
java -jar selenium-server-standalone-3.141.59.jar -role hub -  
hubConfig "path"
```

Starting Selenium Node

Navigate to the path where Selenium jar file is placed.

```
java -Dwebdriver.chrome.driver="C:\Users\Saurabh  
Dhingra\workspace\libs\chromedriver_win32_ver_80\chromedriv  
er.exe" -jar selenium-server-standalone-3.141.59.jar -role node  
-hub http://192.168.0.2:4444/wd/hub
```

Agenda

1. Introduction to Katalon Studio
2. Setup
3. Exploring Katalon studio UI and features
4. Web test automation
5. Modes in Katalon Studio
6. Test Suite and collections
7. Logs & Reports

Agenda

1. Script mode and manual mode
2. Working with multiple browsers - Chrome, Edge
3. Identifying web elements - Locators (Test Object)
4. Runtime Test Object
5. Working with form elements - textbox, dropdown(select), button, checkbox, radio buttons
6. Handling multiple windows, alerts, and iframes
7. Mouse operations
8. Running parallel tests
9. Troubleshooting - Debugging



Introduction to Katalon Studio

NobleProg® Limited 2017
All Rights Reserved

NobleProg

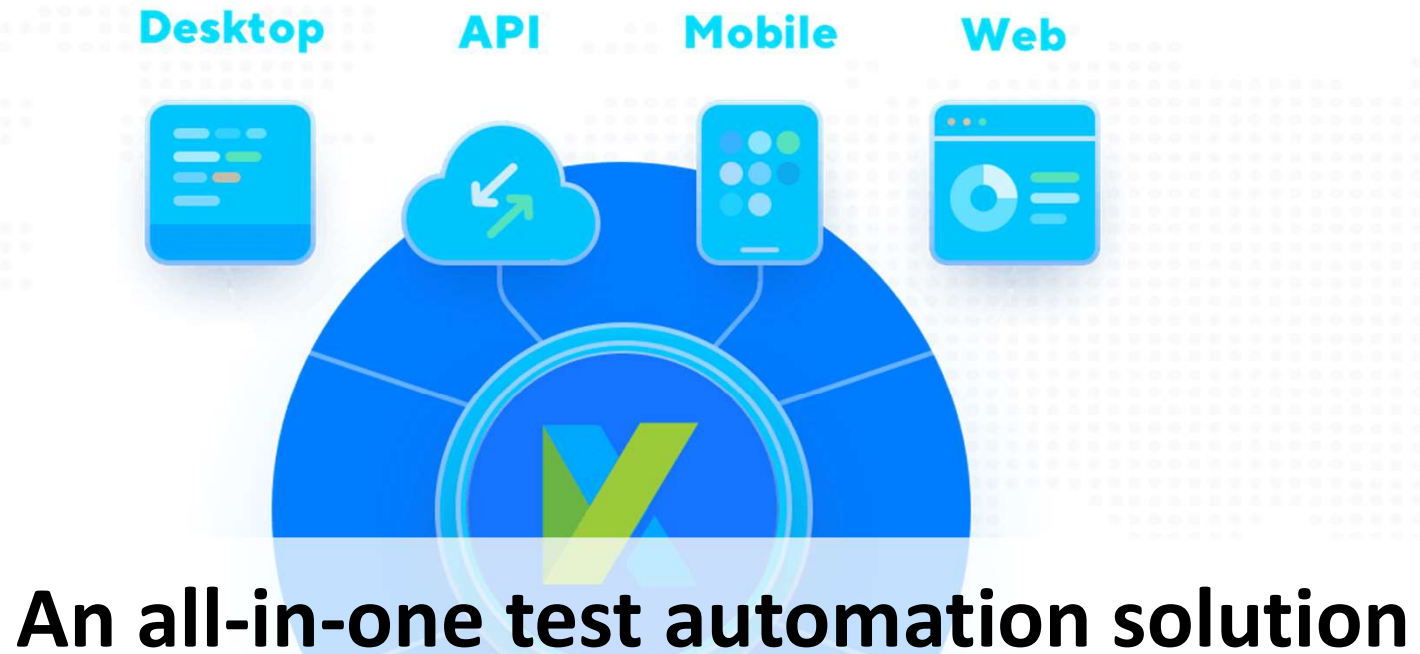
Introduction to Katalon Studio

Katalon Studio is an all in one test automation solution.

Katalon Studio is a smart, robust, and scalable automation solution built for both beginners and expert testers everywhere.

Internally it uses Selenium core engine for UI testing.

Externally it has a user-friendly UI.



Features (Katalon Studio)

1. Build-in reporting
2. Build-in logging.
3. Codeless automation
4. Programming Language Support - Groovy and Java
5. Allow execution on opened sessions
6. Support parallel execution

Katalon Studio - Integrations

Seamlessly integrate with -

1. Source code management tool like Git
2. CI/CD tool like Jenkins, Bamboo, Teamcity, Azure DevOps
3. Test Management tools like JIRA, TestRail, qTest
4. Collaboration tools like Slack, Microsoft Teams
5. Execution platform like Selenium, SauceLabs, BrowserStack, Lambda Test
6. Visual Testing tool like Applitools

Step by step setup

Pre-Requisite -

Install Java on the system

Setup Java path

Installation of Katalon Studio -

Download Katalon Studio

Register

Start the Katalon Studio with katalon.exe file

How Katalon works for UI testing?

1. Generate test case, either by recording or by manually editing.
2. Katalon recognizes objects and controls in tested applications and offers special commands for simulating user actions with them.
3. It also offers specific checkpoints that let you easily verify the application state during the test run.
4. Execute the generated test case.
5. View Execution log
6. Generate execution report

Katalon studio features

1. Recording in Katalon Studio
2. Writing test cases manually
3. Working with Objects and Object Spy
4. Adding Checkpoint

Interacting with..

1. Textbox and text areas
2. Buttons
3. Radio button
4. Checkbox
5. Dropdown (Select)
6. Links
7. Tables
8. Date-picker

Interacting with...

1. Alerts
2. Multiple windows

Performing ...

1. Mouse operations
 1. Mouse hover
 2. Drag and Drop
 3. Double click
 4. Right-click
 5. Scroll down
2. Javascript execution
3. Taking screenshots

Test Suites

1. Creating test suites
2. Creating test suite collection
3. Generate Reports

Data-Driven Testing

1. Data-driven testing via internal tables
2. Data-driven testing via database
3. Data-driven testing via Excel sheet and CSV files



Scripting

NobleProg® Limited 2017
All Rights Reserved

NobleProg

How scripting works?

1. Script a new test case
2. Add or Import new keywords
3. Create a new test listener
4. Build CMD

Groovy Scripting

What is Groovy?

Groovy is an alternative JVM language i.e. language other than Java that run on JVM.

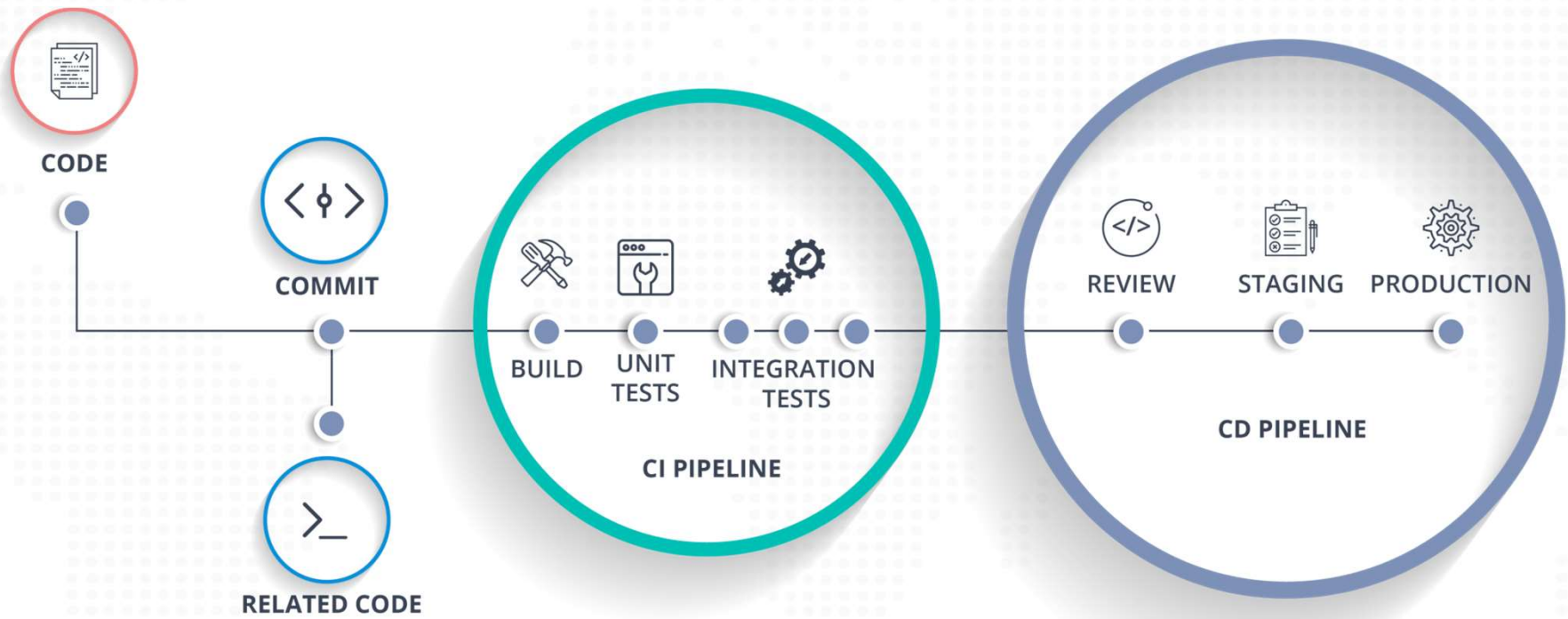
Dynamic language

Compiles to JVM bytecode

Why Groovy?

1. Groovy syntax is very similar to Java programming language
2. You can use existing Java libraries
3. Groovy extends `java.lang.Object`

CI/CD Pipeline



A faint, light gray world map is visible in the background, composed of a grid of small squares.

Integration with CI/CD tool Jenkins

NobleProg® Limited 2017
All Rights Reserved

NobleProg

Agenda -

1. Download and Install Jenkins
2. Running Katalon studio code via CLI
3. Running Katalon studio code via Jenkins pipeline
4. Scheduling the execution of code



Thank You!

NobleProg® Limited 2017
All Rights Reserved

NobleProg