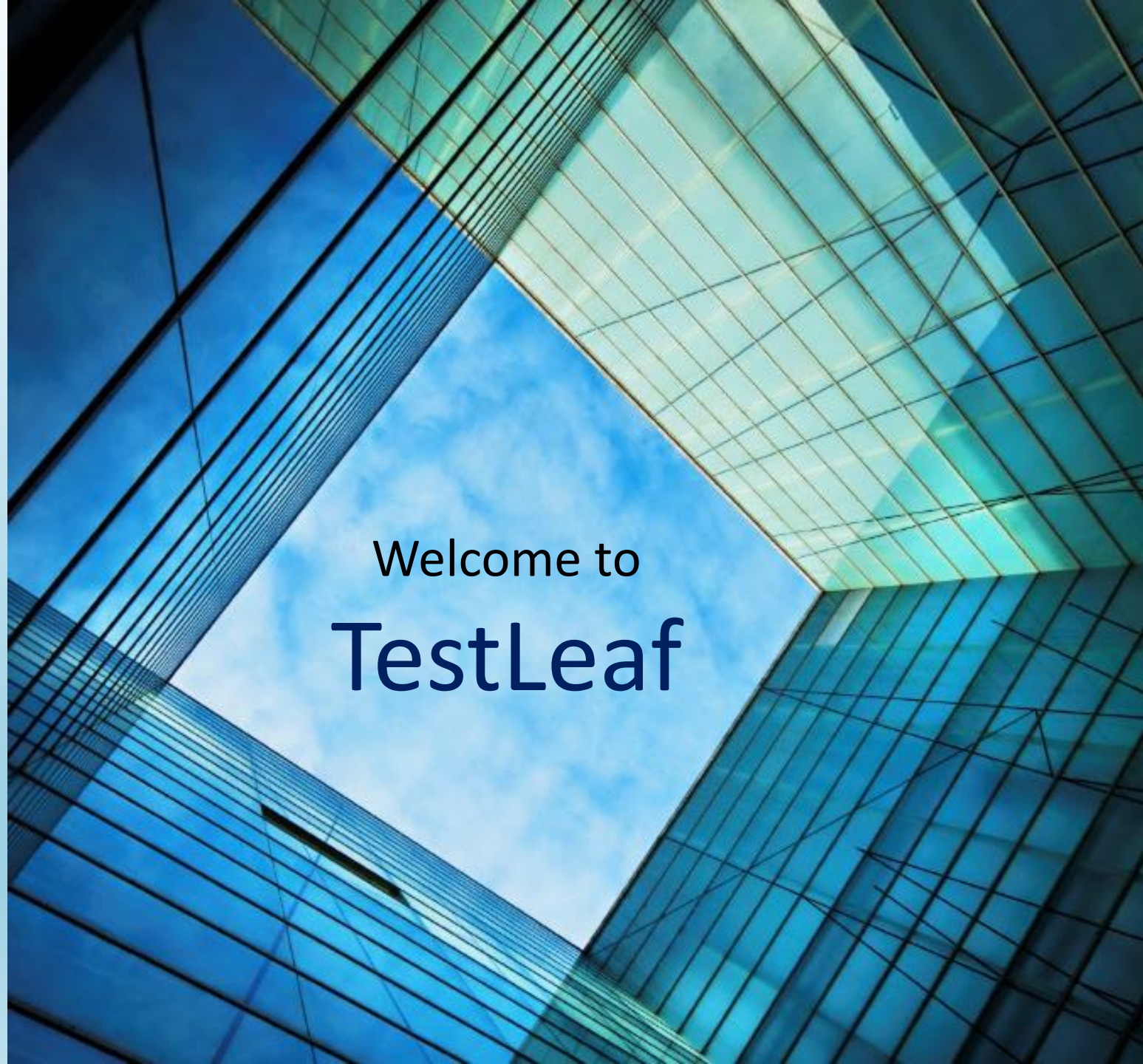
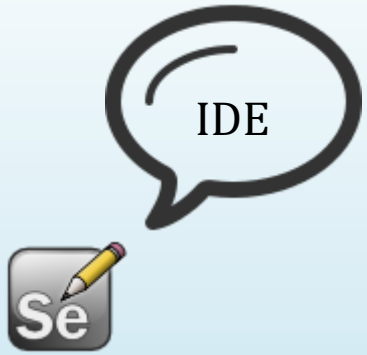
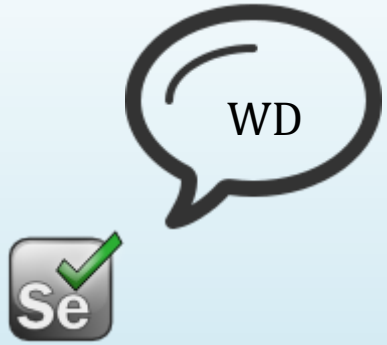
Remote, Parallel Test Execution –

Using Selenium Grid, TestNG





- Record and Playback tool
- Only Firefox add in
- Create quick bug reproduction scripts



- Create robust java based script
- Run with any browser
- Only in local machine
- Can Parameterize, handle exception, take snapshot

- Can execute remotely
- Scale and distribute scripts across many environments

Selenium Components

Central point that will receive all the test request and distribute them the right nodes

Individual machine (physical / virtual) that registers to the hub for test execution

Grid Components

The Hub is the central point that will receive all the test request and distribute them the right nodes.

- Each node gets itself registered
- Console will show the registered nodes and its capabilities

Hub – What that means?

Console shows –

- The connected nodes and their
  - IP Address
  - Browser and its maximum instances
  - Execution status

Hub Console

Syntax to start in command prompt:

java -jar selenium-server-standalone-2.53.0.jar -role hub

URL to see the console:

http://<Hub IP Address / Server Name>:<port>/grid/console

Let us start a Hub and see the console

Capabilities

- Capabilities - Platform, Browser Name, [Version]
- Types – Desired, Actual

Desired Capabilities	Actual Capabilities
This is what you desire to?	This is what is available really?
You may wish to fly by what you wanted to?	You may get it if there is matching flight else none
You wish to fly in any class on the given destination at schedule time?	If there is match with any class, then you get a flight

What can be capabilities for a train?

- Destination – Madurai
- Class – Third AC
- Scheduled Time – 09.30 PM

Syntax to start in command prompt:

// Default settings

java -jar selenium-server-standalone-2.53.0.jar -role wd –hub http://<Hub IP Address or Server Name>:<port>/grid/register

// Specific capabilities settings

-browser "browserName=<browser>,PLATFORM=<platform>,maxInstances=<number>"

Let us start a Node and see the console

- The capabilities at which - you wish the script should run with – like :
  - Browser as Firefox
  - Platform as WINDOWS
  - Version as “ANY” [Hence it can be optional]

Syntax:

```
DesiredCapabilities dc = new DesiredCapabilities();
dc.setBrowserName("firefox");
dc.setPlatform(Platform.WINDOWS);
```

DesiredCapabilities

- Separates where the tests are running from where the browser is.
- Allows tests to be run with browsers not available on the current OS (because the browser can be elsewhere)

Syntax:

new RemoteWebDriver(java.net.URL remoteAddress,
Capabilities desiredCapabilities)

RemoteWebDriver

Let us run them in Parallel using TestNG

So, when the desired capabilities meets the actual capabilities at the hub, the node is allocated and you are ready to run the script!

You Fly!!

If you find a flight that matches your wish.

So, when the desired capabilities does not have the matching actual capabilities at the hub, it throws the exception.

You wait or change the expectations!!

If you do not find a flight that matches your wish.

So, when the desired capabilities have more than one node with matching actual capabilities, it assigns the first node.

You fly on the first flight!

If there are more than one match?

Let us see a demo in local

- It is always preferred to use Grid

  - ✓ run in remote
  - ✓ run in parallel more than a machine (Using Multithreading)
  - ✓ run for several combinations

- Steps to work

  - Start the Hub, Node
  - Verify using console
  - Change webdriver code to RemoteWebDriver
  - Monitor the execution using console

Summary

Q&A

Thank you

TestLeaf
Always Ahead