

## Automation Testing Parallelism–Different Approaches

**Agenda for the document:** Achieving parallelism with different Approaches.

**How:** We can achieve parallelism using multiple ways:

1. Through **TestNg**: Using Testng.xml we can achieve parallelism.

```
<suite name="Parallel test runs" parallel="tests" thread-count="2">
```

```
<test name="TestOne">  
  <classes>  
    <class name="com.sample.TestOneExample" ></class>  
  </classes>  
</test>
```

```
<test name="TestTwo">  
  <classes>  
    <class name="com.sample.TestOneExample" ></class>  
  </classes>  
</test>  
</suite>
```

2. Through **Maven** and **JUnit/TestNg**: Using Maven and JUnit 4.7 onwards / TestNg you can run your tests in parallel with the help of surefire Plugin. For achieving this you must set the **parallel parameter**, and change the **threadCount** attribute in pom.xml.

```
<plugins>  
  [...]   
  <plugin>  
    <groupId>org.apache.maven.plugins</groupId>  
    <artifactId>maven-surefire-plugin</artifactId>  
    <version>2.18.1</version>  
    <configuration>  
      <parallel>methods</parallel>  
      <threadCount>10</threadCount>  
    </configuration>  
  </plugin>  
  [...]   
</plugins>
```

### 3. Using Gradle:

Once we apply the Java plugin we can run our tests with the test task, we can also run tests in parallel though by default it runs sequentially.

We need to configure "maxParallelForks" property to more a number more than 1, By default its "1". The "forkEvery" property specifies the maximum number of test classes to execute in a test process, so that your test process will not be using a very large heap, and can be restarted after it has executed specific number of test classes. The default is to execute an unlimited number of tests in each test process.

### 4. using MbUnit for C#-Selenium:

For .net Application while working with C# and Selenium if We use NUnit then we can not achieve Parallel Testing, so we can achieve parallelism using MbUnit, which provides a way to run the Automation Scripts parallelly in the following way:

In the Assembly file we need to mention the below lines:

```
[assembly: DegreeOfParallelism(2)]  
[assembly: Parallelizable(TestScope.All)]
```

Here "2" is the number of instances of browser.

### 5. Using NOSE for Python-Selenium:

In Order to run Test Parallel for Python -Selenium we can use nose:

After installing NOSE we can write the following :

```
nosetests --processes=2
```

this means you are going to run 2 tests at the same time. If you get TimedOutException then you can increase the TimeOut for the Process as:

```
nosetests --processes=2 --process-timeout=10000
```

By this way We can achieve parallel testing using Selenium and with Multiple Unit Test Frameworks.