

Mocha

Mocha is a feature-rich JavaScript test framework running on Node.js and in the browser, making asynchronous testing simple and fun. Mocha is one of the most popular JavaScript testing frameworks, with over 18k stars in GitHub. This section guides you through integrating Mocha and Selenium WebDriver with Perfecto and Perfecto Smart Reporting.

A sample project is available at this GitHub [Mocha-Perfecto sample repository](#).

Prerequisites

Integrating Perfecto and Perfecto Smart Reporting with Mocha requires the following npm packages:

- mocha
- chai
- perfecto-reporting
- selenium-webdriver

1 | Initialize and set up a Mocha project

In this step, we start a new Mocha project from scratch.

1. Install Node.js.
2. To establish the npm project, run the following npm commands:

npm commands

```
npm init
npm install mocha chai perfecto-reporting selenium-webdriver --save-dev
```

3. Create a test folder.
4. Create test suite scripts with the file extension `.spec.js`, for example `tests.spec.js`.
5. Open `package.json` and edit the `script` entry as follows:

package.json sample

```
"scripts": {
  "test": "./node_modules/.bin/mocha"
},
```

This way, when you need to run the test later, you can just use the following simple command:

```
npm test
```

On this page:

- [Prerequisites](#)
- [1 | Initialize and set up a Mocha project](#)
- [2 | Set up and integrate Perfecto reporting via a callback helper](#)

2 | Set up and integrate Perfecto reporting via a callback helper

To launch a Perfecto device session, you must specify a RemoveWebDriver session and device selection capabilities in the helper file.

1. Specify the Perfecto URL and capabilities as follows:

URL and Capabilities

```
var sel      = require('selenium-webdriver');
var perfectoReporting = require('perfecto-reporting');
var capabilities = {
  'platformName' : '<iOS or Android>',
  'deviceName' : '<device_name>', //you can also use other capabilities to specify devices you want
to work with
  'bundleId' : '<app_bundle_id>', //your app bundle id, e.g. com.apple.calculator
  'browserName' : 'mobileOS',
  'securityToken' : '<perfecto_sec_token>'
}

var REMOTE_URL = 'https://<cloud_name>.perfectomobile.com/nexperience/perfectomobile/wd/hub/fast';
```

2. Use Selenium WebDriver to launch the Perfecto RWD session:

perfectoReportHelper.js

```
browser = await new sel.Builder().withCapabilities(capabilities).usingServer(REMOTE_URL).build();
await browser.manage().setTimeouts({implicit:20000});
```

3. Use this browser object to perform any RWD-based activities. For example:

Mocha Sample

```
describe('<Suite Name>', async function() {
  var i = 0;
  while (i++ < REATING_NUM) { // repetitive tests
    it('<Scenario name>', async function() {
      await browser.findElement(sel.By.xpath('<element_xpath>')).click();
      expect(await drv.findElement(sel.By.xpath('<element_xpath>')).getText()).to.be.equal
('3'); //chai assertion
    });
  }
});
```

4. To integrate the automated Perfecto reporting feature on the fly, without changing the existing code structure of each test scenario, set up a customized **describe** function with driver launch callback and reporting and teardown hooks. In the following example, this function is called **describeWithPerfecto**:

describeWithPerfecto function

```
module.exports = function describeWithPerfecto(name, callback, body) {
  let browser;
  let reportingClient;
  describe(name, function() {
    this.timeout(50000);
    // as we using beforeEach here, new browser will be received for every test case
    before(async function() {
      try{
        //launch the remoteWebDriver
        browser = await new sel.Builder().withCapabilities(capabilities).usingServer
(REMOTE_URL).build();

        await browser.manage().setTimeouts({implicit:20000});
        var perfectoExecutionContext =
          await new perfectoReporting.Perfecto.PerfectoExecutionContext({
            webdriver: browser,
            //set the CI job name and CI job number, can be parameterized
            job: {jobName: "MochaPerfectoCI",buildNumber:3},
            //Set the tags
            tags: ['selenium tests']
          }
        );
        reportingClient =
          await new perfectoReporting.Perfecto.PerfectoReportingClient
(perfectoExecutionContext);

        }catch(e)
        {
          console.log(e);
        }
        r={};
        r.drv=browser;
        r.rpt=reportingClient;
        callback(r);
      });
      beforeEach(async function(){
        await reportingClient.testStart(this.currentTest.title); //report test start before each test
scenario
      });
      afterEach(async function(){
        if(this.currentTest.state === "passed")
          await reportingClient.testStop({status: perfectoReporting.Constants.results.passed}); //
report pass in case test scenario is passed
        else if(this.currentTest.state === "failed")
          await reportingClient.testStop( {status: perfectoReporting.Constants.results.failed, //
report fail along with error msg in case test scenario is failed
          message: this.currentTest.err.message
        });
      });

      body(); //user test script body

      // teardown when suite is done
      after(function() {
        return browser.quit();
      });
    });
  });
};
```

5. Use this customized describeWithPerfecto function as follows:

describeWithPerfecto sample

```
describeWithPerfecto('<Suite Name>',
b=>{drv=b.drv;rpt=b.rpt;}, //callback to pick up the Perfecto RWD and PerfectoReporting client from
describeWithPerfecto

function(){ //The real test script body
    var i=0;

    while (i++ < REPEATING_NUM) { // repetitive tests
        it('<Scenario Name>', async function() {
            await rpt.stepStart('Click on element')
            // Optionally if test steps need to be marked out, you can insert
stepStart to mark starting of a logical step
            await drv.findElement(sel.By.xpath('<element xpath>')).click();
            await rpt.stepEnd() // by insert stepEnd to mark the current logical
step end
            await rpt.stepStart('Validate Element Equals 3')
            expect(await drv.findElement(sel.By.xpath('element xpath')).getText()).to.be.equal
('3'); //chai assertion
            await rpt.stepEnd()
        });
    }
});
```