

# Node.js Test Setup Example

This script illustrates the basic set up of a test that runs in the Sauce Labs virtual browser cloud. It includes using environment variables for authentication credentials, using tags and build numbers for test result management, and reporting the Pass/Fail status of the test to the Sauce Labs dashboard.



## Example Only

The code in this topic is presented as an example only, since your tests and testing environments may require specialized scripting. This information should be taken only as an illustration of how you could set up your tests with Sauce Labs, and is not directly supported by Sauce.

- [Prerequisites](#)
- [Setup Example Script](#)

## Prerequisites

- You should have [a Sauce Labs account](#)
- You should have a node.js Selenium environment already set up on the local machine where you'll launch your tests. If you need help with this, check out the ReadMe in the [JavaScript sample script repository on GitHub](#).
- You should review the [Best Practices for Running Tests](#)

## Setup Example Script

When this script runs, it opens the browser specified in the script, navigates to the saucelabs.demo web app, and then closes the browser. If you want to run this script using your Sauce Labs credentials and have it navigate to your own web app:

1. Follow the instructions in [Best Practice: Use Environment Variables for Authentication Credentials](#) to set up your Sauce Labs account access.
2. Enter the URL for the web app you want to test in the place of [saucedemo.com](#).
3. Once you've been able to run the test against your web app, you can use the [Platform Configurator](#) to experiment with changing the desired capabilities of the test.

Clone this script from the saucelabs-training GitHub repo: <https://github.com/saucelabs-training/demo-js/blob/master/on-boarding-modules/mocha-examples/test/module4-mocha-test.js>

**on-boarding-modules/mocha-examples/test/module4-mocha-test.js** (source from Sauce Labs Training JavaScript)

```
const promise = require('selenium-webdriver');
let expect = require('chai').expect;
let webdriver = require('selenium-webdriver');

promise.USE_PROMISE_MANAGER = false;

let username = process.env.SAUCE_USERNAME,
    accessKey = process.env.SAUCE_ACCESS_KEY,
    /* Change the baseUrl to your application URL */
    baseUrl = "https://www.saucedemo.com",
    tags = ["sauceDemo", "on-boarding", "node", "mocha" ],
    driver;

describe('Instant Sauce Test Module 4', function() {
  this.timeout(40000);

  beforeEach(async function () {
    driver = await new webdriver.Builder().withCapabilities({
      'browserName': 'chrome',
      'platformName': 'Windows 10',
      'browserVersion': 'latest',
      'goog:chromeOptions': { 'w3c' : true },
      'sauce:options': {
        'username': username,
        'accessKey': accessKey,
        'build': 'Onboarding Sample App - NodeJS + Mocha',
        'name': '4-best-practices',
        /* As a best practice, set important test metadata and execution options
        such as build info, tags for reporting, and timeout durations.
        */
        'maxDuration': 3600,
        'idleTimeout': 1000,
        'tags': tags
      }
    }).usingServer("https://ondemand.saucelabs.com/wd/hub").build();

    await driver.getSession().then(function (sessionId) {
      driver.sessionID = sessionId.id_;
    });
  });

  afterEach(async function() {
    await driver.executeScript("sauce:job-result=" + (this.currentTest.state));
    await driver.quit();
  });

  it('should open chrome ', async function() {
    await driver.get(baseUrl);
    const title = await driver.getTitle();
    console.log('Page Title is: ' + title);
    expect(title).equals('Swag Labs');
  });
});
```

