

# Getting Started with JsTestDriver IntelliJ plugin

## Requirements

- If you aren't aware of JsTestDriver concepts, consider visiting <http://code.google.com/p/js-test-driver/wiki/GettingStarted>. There is also an excellent [tutorial introduction](#) to JavaScript unit testing by Christian Johansen.
- If you don't have JsTestDriver IntelliJ plugin installed, please visit [Installation of JsTestDriver IntelliJ plugin page](#).

## Setup project

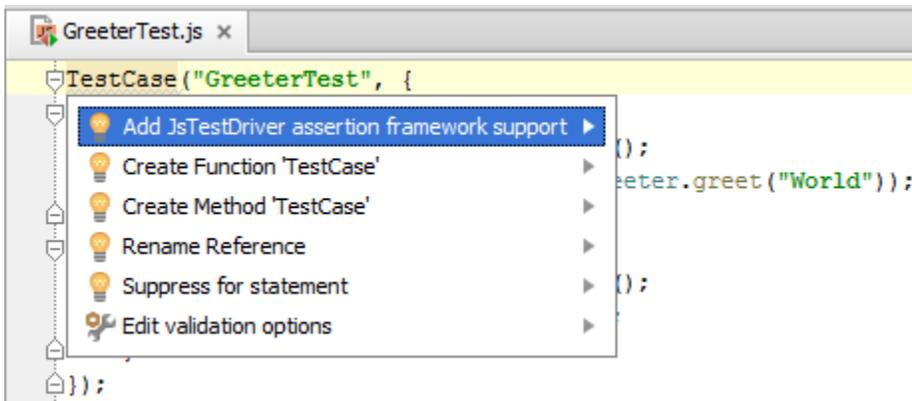
We have created a sample project for you. Please download [greeter-sample example project](#), extract it and open the project in your IDE.

To enable JSTestDriver assistance go to the Settings | Editor | Inspections and turn on JavaScript | JsTestDriver checkbox.

On opening GreeterTest.js you will see highlighting on the TestCase method call expression and tooltip asking about adding JsTestDriver assertion framework support. Press Alt+Enter/Option+Enter and press OK in the opened "Code Assistance For JsTestDriver Assertion framework" dialog.



Or you may position the caret on the highlighted code, press Alt+Enter/Option+Enter and choose "Add JsTestDriver assertion framework support" from the list of available intention actions.



A global JavaScript library named "JsTestDriver Assertion Framework" will be created and associated with a project. To view it, open the Settings dialog (File | Settings) and select JavaScript Libraries.

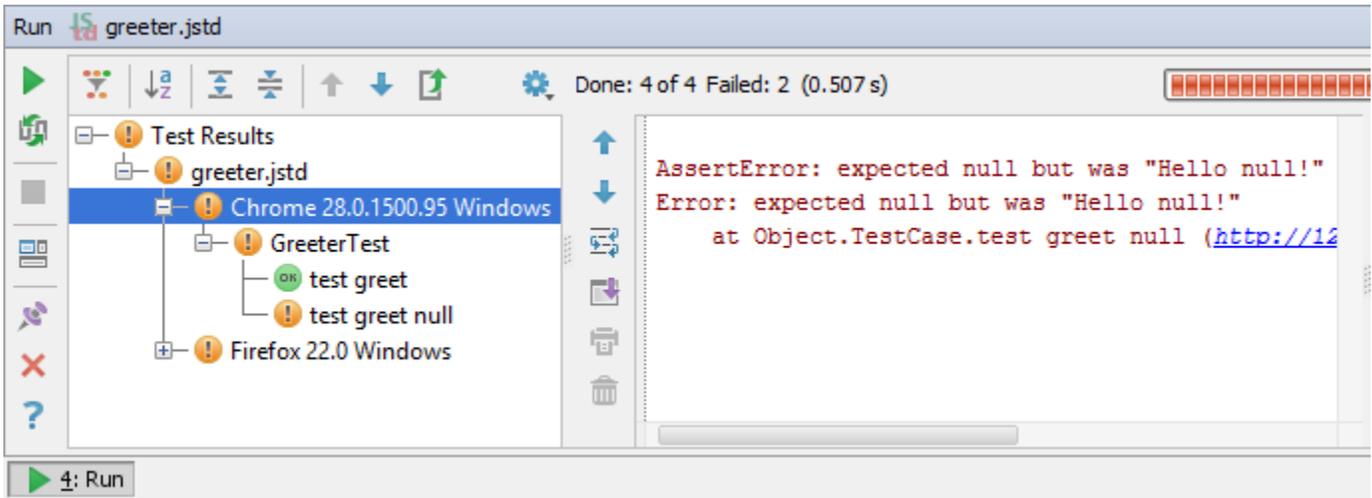
Before you can run any of your tests you need to capture at least one slave browser. The server does not have to reside on the machine where the test runner is, and the browsers themselves can be on different machines as well.

## Running tests

To run tests, right-click greeter.jstd and choose "Run 'greeter.jstd'". Tests will be run against the local JsTestDriver server,

additional JSTestDriver Server tool window will be added at the IDE right bottom corner.

To capture a browser you can click on the suggested link in the Run tool window (like <http://127.0.0.1:9876/capture>) or copy the link and paste it into the desired browser(s).



In our case we have one failed test "GreeterTest test greet null". You can easily navigate from the stack trace to the source code that causes problems. You can also navigate from the test results tree to the test case/test function declaration using the F4 key.

## Generate code

In order to remove duplicated code that sets the greeter local variable, you can refactor the Greeter object creation to a special method called setUp, which is called before each test function is executed. Just invoke the Generate action by pressing Alt+Insert inside of a TestCase method call expression, and then choose "JsTestDriver Setup".



Use the Shift+F10 keyboard shortcut to re-run the tests after changing code.

Test with pleasure!