Testing Angular apps using Protractor in JavaScript

Overview

In this tutorial, we will create some tests in JavaScript using Jasmine and Protractor.

Requirements

- protractor
- jasmine
- jasmine-reporters

Code

conf.js

```
exports.config = {
  directConnect: true,
  // Capabilities to be passed to the webdriver instance.
  capabilities: {
    'browserName': 'chrome'
  },
  // Framework to use. Jasmine is recommended.
  framework: 'jasmine',
  // Spec patterns are relative to the current working directory when
  // protractor is called.
  specs: ['example_spec.js'],
  // Options to be passed to Jasmine.
  jasmineNodeOpts: {
    defaultTimeoutInterval: 30000
  },

  onPrepare: function() {
    var jasmineReporters = require('jasmine-reporters');
    var junitReporter = new jasmineReporters.JUnitXmlReporter({
      // setup the output path for the junit reports
      savePath: 'output/',
      // conslidate all true:
      //   output/junitresults.xml
      //
      // conslidate all set to false:
      //   output/junitresults-example1.xml
      //   output/junitresults-example2.xml
      consolidateAll: false
    });
    jasmine.getEnv().addReporter(junitReporter);
  }
};
```

google_testing.js

```
describe('angularjs homepage', function() {
  it('should greet the named user', function() {
    browser.get('http://www.angularjs.org');
    element(by.model('yourName')).sendKeys('Julie');
    var greeting = element(by.binding('yourName'));
    expect(greeting.getText()).toEqual('Hello Julie!');
  });
  describe('todo list', function() {
    var todoList;
    beforeEach(function() {
      browser.get('http://www.angularjs.org');
      todoList = element.all(by.repeater('todo in todoList.todos'));
    });
    it('should list todos', function() {
      expect(todoList.count()).toEqual(2);
      expect(todoList.get(1).getText()).toEqual('build an angular app');
    });
    it('should add a todo', function() {
      var addTodo = element(by.model('todoList.todoText'));
      var addButton = element(by.css('[value="add"]'));
      addTodo.sendKeys('write a protractor test');
      addButton.click();
      expect(todoList.count()).toEqual(3);
      expect(todoList.get(2).getText()).toEqual('write a protractor test');
    });
  });
});
```

After running the tests and generating the JUnit XML report (e.g., junitresults-angularjshomepage.xml), it can be imported to Xray (either by the REST API or through the **Import Execution Results** action within the Test Execution).

```
protractor  conf.js
```

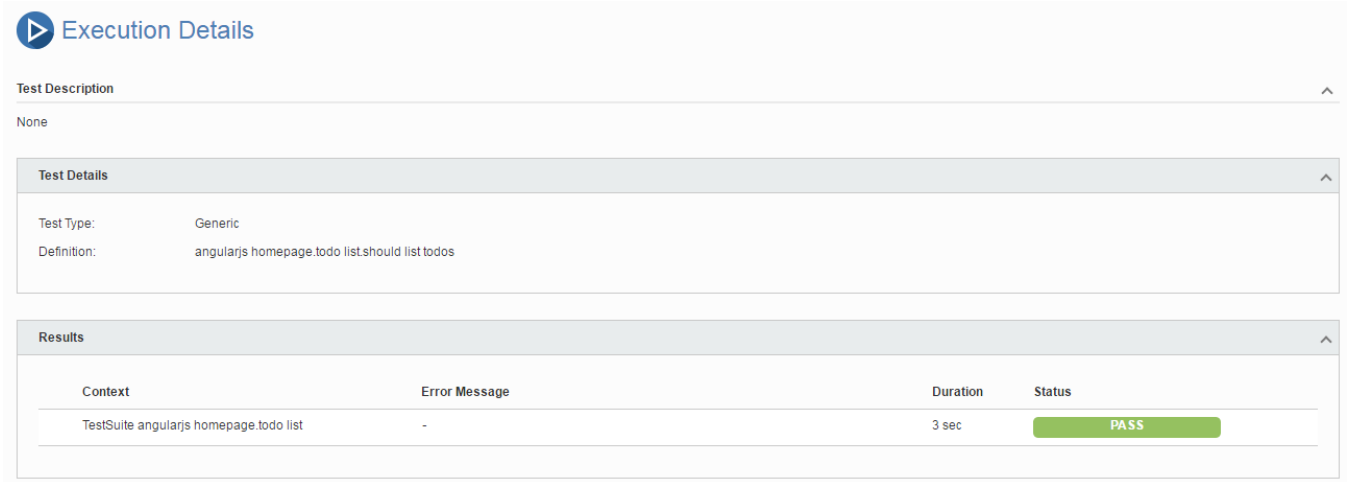
JUnit's Test Case is mapped to a Generic Test in Jira, and the **Generic Test Definition** field contains the name of the test concatenated with the alias of each test case "assert" statement.

	Key	Summary	Test Type	#Req	#Def	Test Sets	Assignee	Status	
1	CALC-122	should list todos	Generic	0	0			PASS	▶ •••
2	CALC-121	should greet the named user	Generic	0	0			PASS	▶ •••
3	CALC-123	should add a todo	Generic	0	0			PASS	▶ •••

Showing 1 to 3 of 3 entries

First Previous 1 Next Last

The Execution Details of the Generic Test contains information about the Test Suite, which in this case corresponds to the description of the first "describe".



References

- http://www.protractortest.org/#/
- https://github.com/angular/protractor/blob/master/example/example_spec.js
- https://github.com/angular/protractor/blob/master/docs/browser-setup.md
- https://github.com/angular/protractor-cookbook/tree/master/jasmine-junit-reports