

Testing AngularJS with Jasmine

Course Summary

Description

The **Testing AngularJS with Jasmine** training course teaches developers how to effectively test AngularJS based applications using the Jasmine open-source testing framework. The Testing AngularJS with Jasmine course assumes both experience with JavaScript, Angular, and Jasmine.

If you are looking for a comprehensive AngularJS testing course, consider our [AngularJS Boot Camp](#) .

Objectives

At the end of this course, students will be able to:

- Identify the testing areas associated with AngularJS
- Create Angular Mocks
- Independently test controllers, filters, directives, and services

Topics

- Concepts review
- Angular scenario runner
- Angular mocks (http mocks, etc)
- Writing Effective Tests
- Advanced topics
- Protractor E2E

Audience

This course is designed for developers.

Prerequisites

- Introduction to AngularJS
- Test-driven Development
- Testing JavaScript with Jasmine

Duration

One day

Testing AngularJS with Jasmine

Course Outline

- I. Concepts review**
 - A. Testing concepts
 - B. Jasmine review
 - C. AngularJS Architecture review
- II. Angular scenario runner**
 - A. End-to-End (E2E) testing
 - B. Simulating user interaction
 - C. Navigating pages and routes
 - D. Reloading pages
 - E. Location object
 - F. Setting expectations
 - G. Page content interaction
 - H. Handling Angular Binding
 - I. Form interaction
 - J. Testing loops
- III. Angular mocks (http mocks, etc)**
 - A. Angular Mocks library with 'angular-mocks.js'
 - B. Dependency injection
 - C. Fake HTTP back-end with '\$httpBackend'
 - D. Setting expectations with 'expect'
 - E. Request handling
 - F. Setting back-end definitions with 'when'
 - G. Stub responses and calls
 - H. jasmine spies
- IV. Writing Effective Tests**
 - A. Page content
 - B. controllers, Unit and E2E
 - C. filters, Unit and E2E
 - D. directives, Unit and E2E
 - E. services / factories, Unit and E2E
 - F. templates, Unit and E2E
 - G. events, Unit and E2E
- V. Advanced topics**
 - A. Continuous integration
 - B. Code coverage
 - C. remote/distributed testing
- VI. Protractor E2E**
 - A. Protractor & Karma differences
 - B. WebDriver
 - C. Selenium
 - D. Protractor configuration
 - E. Using 'element' & 'by'
 - F. Page Objects