Single Page Application Development with AngularJS

Course code: IJ - 30
Course domain: Software Engineering
Number of modules: 1
Duration of the course: 36 astr. (48 study\(^1\)) hours

Sofia, 2016

Copyright © 2003-2016 IPT – Intellectual Products & Technologies Ltd. All rights reserved.

\(^1\) Duration of a study hour is 45 minutes.
Single Page Application Development with AngularJS

STUDY PLAN

<table>
<thead>
<tr>
<th>Module name</th>
<th>1. Single Page Application Development with AngularJS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lectures, astr. hours</td>
<td>17</td>
</tr>
<tr>
<td>Laboratory exercises, astr. hours</td>
<td>17</td>
</tr>
<tr>
<td>Final test and practical problem, astr. hours</td>
<td>2</td>
</tr>
<tr>
<td>Total, astr. hours</td>
<td>36</td>
</tr>
</tbody>
</table>

Lecturer: Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.
E-mail: ttiliev@iproduct.org

Target audience: Medium level JavaScript developers with practical experience in building web applications using HTML 5, CSS 3, JavaScript/ECMAScript 6, and Bootstrap.

Course duration: Duration of the course is 36 astr. (48 study) hours total. Training will be conducted in 9 days – 4 astr. hours each day.

Course Description:

The course provides in-depth study of state-of-the-art JavaScript (ES 5 & 6) and AngularJS 1.5 MV* framework for rapid development of modern, mobile-first, responsive single-page applications that are easy to extend and maintain in long run. The main topics that will be covered during the course include:

1. Single Page Applications (SPA). Horizontal and vertical SPA. Introduction to Model-View-Controller (MVC), Model-View-Presenter (MVC), Model-View-ViewModel (MVVM) – MV* patterns for development of more complex, extensible and easy to maintain web applications using AngularJS framework. Advantages and typical use-cases of AngularJS as a complete solution for client-side MV* SPA development. Developing AngularJS applications with VS Code. Creating and bootstrapping simple AngularJS TODO Application. (3 h.)


4. **AngularJS Services** – built-in and custom services. Providing and consuming services. Dependency Injection (DI) – factory methods, module methods config and run, implicit, inline array and $inject annotations, using ngAnnotate tool to add, remove

5. AngularJS routing – components, routes, outlets, URLs, links, navigation. Three AngularJS router choices: ngRoute module, component router, and UI router (state-based + extra features). UI router as de-facto standard for more complex AngularJS apps. State management and navigation – $stateProvider, ui-view directive. Activating a state – $state.go(), ui-sref directive, and by navigation in the browser. State configuration – name, template, templateUrl, controller, controllerProvider, resolve, and data properties, onEnter and onExit callbacks. State change and view load events. Nested states and nested views – children and parent properties, object (dot) state naming notation, inherited resolved dependencies and custom data, abstract states. Multiple named views and paired states navigation – view reference scheme: viewname@statename, relative and absolute names. URL routing – URL basic and regex parameters, path, query, and state specified parameters, wildcards, using parameters in links and with programmatic navigation. Absolute routes (^). $stateParams service. $urlRouterProvider service – when (redirection), otherwise (page not found handling), rule (custom navigation rules). $urlMatcherFactory, UrlMatchers, and $templateFactory. Building SPA with more complex navigation requirements using UI router.(4 h.)


7. Test Driven Development (TDD) – unit/end-to-end testing AngularJS classes and components using Jasmine, Karma, angular-mocks (ngMock) and Protractor. Separation of concerns and dependency injection. Testing controllers, services, filters and directives. Using beforeEach(). Testing transclusion, external template, and embedded directives – tips and tricks, debugging. Using karma-ng-html2js-
preprocessor to pre-compile HTML templates. Testing promises and asynchronous behavior. Review of sample AngularJS GitHub example projects (bigger ones). (4 h.)


9. **Final test + practical problem** + discussion of additional questions. (2 astr. h.)

The course contains 50% lecture materials and 50% lab exercises. Lectures and exercises will be conducted in parallel and will not be divided in separate sessions in order to achieve immediate reinforcement of new concepts with practical examples and problem solving activities.

During the course participants will get practical experience using *AngularJS 1.5 framework* for building *single page applications* by solving problems and exercises. The learning is conducted in small groups – up to 8 participants using problem-based methodology. During the workshop there will be opportunity for discussion of additional questions the participants are interested in. At the end of the course participants are expected to solve a practical problem – develop simple SPA with AngularJS and routing.