AngularJS and Single Page Applications (code: ANGULARJS)

Overview

AngularJS has achieved tremendous – and unsurprising – success during the last couple of years. It perfectly fits the needs of modern web applications, as the majority of them consist of two distinct modules:
— JavaScript-based front-end that uses HTML5 and communicated with the back-end through REST services
— back-end that serves as a pure data source for the front-end – often simply a CRUD server and authentication-authorization layer.

This training - *AngularJS and Single Page Applications* is focused on the first part: **an effective browser-side application written in JavaScript**.

Like other landmark frameworks and libraries (such as *Ruby on Rails* and JavaScript's own *JQuery*), *AngularJS* introduces its own programming philosophy. To learn AngularJS, you need to learn more than just the library's syntax and services. You need to master the idea behind it, the library's idioms and best practices – and how to avoid the pitfalls that await inexperienced AngularJS programmers.

Many think that *AngularJS* has one flaw: while simple use cases are indeed very simple (declarative syntax and directives allow for creating quite advanced applications without a single line of JavaScript), the more complex ones require exponentially more skill from the programmer. This is why this training is focused on complex, advanced *AngularJS* mechanisms – and on explaining the idea behind some of the less obvious design decisions. We help the participants to learn **more than just practical skills in a single library, but also general knowledge of JavaScript-based client applications**.

Duration

3 days

Agenda

1. AngularJS – application architecture
   — patterns, anti-patterns and controversies: MVC, MVVM, MVP;
   — application components: model, controller, scope, view, directive
   — bidirectional data binding
   — basic directives (including ngApp, ngRepeat, ngSwitch, ngView, ngClass, ngInclude, ngShow, ngHide, ngClick...)
   — injecting dependencies based on names and metadata
2. A more complex front-end
   — organizing application logic with promises, $q services and messages; event propagation
   — details of bidirectional data binding – digest cycle, watch expressions and their syntax, filters (including custom filters) - $watch, $digest, $apply
   — vector and bitmap graphics in the view layer
   — $animate service, element identity and binding a given element to data
3. Complex user interface
   — Handling state and URLs; $router and $location services
   — routing and external templates in practice
   — History API and using fragments

— Forms: FormController, validation, messages, tracking changes
4. Server communication
   — server communication design patterns
   — architecture: loose and tight server - front-end coupling
   — $http and $resource services
   — typical server communication solutions and techniques
5. Code organization
   — modules, injector configuration
   — using requirejs
   — unit testing with Karma and Jasmine
   — details of $injector and $provide services – the uncomfortable truths about registering value, provider, factory and services
   — application initialization
   — custom directives – directive lifecycle, transclusion, isolated and non-isolated scopes

Target audience and prerequisites

The training requires fluency in HTML and in at least one server-side programming language (e.g., PHP, Java, Ruby, Python – including at least basics of object-oriented programming). The participants should also have at least a cursory knowledge of JavaScript and JQuery.

Certificates

Course participants receive completion certificates signed by ALX.

Locations

— Warsaw (English) – Jasna 14/16A
— Online (English) – your home, office or wherever you want
— any other location (London, UK, EU) on request

Price

990 EUR

The price includes:

— course materials,
— snacks, coffee, tea and soft drinks,
— course completion certificate,
— one-time consultation with the instructor after course completion.