AngularJS

Duration 3 day(s) (ANGULARJS-03)

Rich application development with AngularJS

Description

AngularJS is a burgeoning JavaScript framework, aimed at structuring and simplifying the development of rich client-side applications. Developed by Google under open-source license, it natively integrates all the features usually obtained by assembling specialized libraries: routing, templates, bindings, REST calls ... Its robustness, versatility and built-in test bench make it a must-have element of any REST-oriented web application! In this tutorial, you will discover two advanced features of AngularJS: how to create your own reusable components through the Guidelines, and how to implement custom filters, and then learn how to handle authentication of your users to a third-party system.

Goals

- Discover AngularJS
- Develop a rich application from a static model
- Integrate into a REST oriented architecture

Public

- Web developer

Prerequisites

- Mastery of the Javascript language
- Knowledge of web technologies

Structure

50% Theory, 50% Practice

A few reminders to get started

- Javascript: prototypes, closures and callbacks
- The Web / Javascript Developer Toolkit
- What is a REST oriented architecture?

AngularJS: positioning and general principles

- A rich ecosystem of specialized bookstores
- JQuery, the Swiss knife of the web developer
- Architecture and features of AngularJS

First steps: from the model to the application

- Integration of the AngularJS library
- Binding concept
- Use of expressions

Controllers and Scopes

- Association of a controller to a template
- Notion of scope, and hierarchy of scopes
- Propagation of binding events

Router: single-page, multi-view applications

- Configuration of the routing, notion of deep-linking
- Management of the browsing history and access to the parameters of the URL
- Prepare the display of a view

Modules and services: a flexible architecture

- Module concept, default module
- Modularization of the application as services
- Dependency injection: benefits and good practices

Forms management

- Validation of the input fields
- Form processing

Integration into a REST architecture

- AJAX calls with $ http service
- Consume REST resources with the $ resource service
- Asynchronous update: notion of promise

Internationalization (I18N)

- Activation of internationalization
- Formatting dates, numbers and currencies

Filters

- Establishment of an existing filter
- Setting filter "filter"
- Develop your own filter

Directives

- Definition and API
- Life cycle of a directive;
- Manipulation of the scope in the directives
- Collaboration with ngModelController
- Transclusion
- Toolbox: $ parse, $ compile, angular.element

Authentication

- Special features for single-page applications
- Good practices and tools for authentication

Unit tests

- Technology approach: Jasmine, KarmaJS
- Unit tests with ngMock
- End to end tests with Angular Scenario

Beyond AngularJS

- Industrialization of development with Yeoman
- Optimization of resources (javascript, images)
- Simplify CSS styles with a preprocessor
- Angular-UI and other modules