

Angular 5

Theoretical & Practical deep learning course

40 Hours

Instructor: Yair Shivek (Siwek)

Angular is a JavaScript Client Side Framework. Angular maintain compatibility with web standards and it is easy to learn. Angular will allow you to reuse your code in any deployment target such as web, mobile web, native mobile and native desktop. Angular is using the popular reactive programming library RxJS and allow you to build very fast applications with the maximum speed possible on the Web Platform today. Angular is coming with a great set of tooling helping you to write good code, test it and efficiently move it from development to production.

Prerequisites:

Developers with programming and web development experience.

Module 1 - Introduction to Angular

- Single Page Applications (SPA)
- The need for javascript framework
- Reactive Programming and the future of Web Applications
- Setting up the environment
- “Hello Angular ” - your first app.

Module 2 - TypeScript

- JavaScript, ECMAScript5 , ECMAScript 6 and Type Script
- Understanding the JavaScript Closures
- Types
- Classes
- Inheritance
- Generics Modules
- Tools

Module 3 - Angular behind the hood

- Angular architecture
- How angular works
- The application
- The model
- The components
- The component decorator
- Bootstrapping
- NgModules
- Putting it all together

Module 4 - Angular built in components and directives

- NgFor
- NgIf
- NgSwitch
- NgNonBindable
- NgStyle
- Etc....

Module 5 - Angular template syntax and property binding

- Template Syntax
- Property Binding
- Events
- Two Way Binding
- Change Detection and lifecycle hooks

Module 6 - Services, Observers and the Rxjs library

- Angular Services
- Implementing custom services
- Understanding Reactive Programming
- Reducing the Boilerplate with NgRx
- Working with the RxJS library
- Working with data Observables
- Promises vs. Observables
- NgRx Schematics

Module 7 - Angular Forms

- Forms in angular
- Using controls in forms
- Using Control Groups
- Constructing a Form using a FormBuilder
- Input Validations
- Watchers
- The NgModel
- ReactiveForms Module
- Angular Dynamic Forms
- Ngrx Complex Form

Module 8 - Routing and navigation

- What is Routing
- The need for routing
- Routing options
- Path Location
- Passing parameters
- Guards
- Resolve injectables
- Nested routes
- Lazy Loading and Performance

Module 9 - Views, Directives and responsive design

- Creating views and html templates
- Creating Directives
- Responsive design and your template layout
- Using themes
- Introduction to Google's Material Design
- Material Design vs. Bootstrap
- Working with Angular Material / Angular Bootstrap

Module 10 - communicating with the server

- XHR and AJAX
- Working with Angular http
- Basic get request
- Using REST
- Http and promises
- Http api
- RxJs Mapping
- Best Practices

Module 11 - Data Model

- The importance of a solid data model
- Constructing your Data Models
- Validating data in the model
- Sharing data model with the server

Module 12 - Security and Token authentication

- Introduction to outh
- oauth vs owasp
- Introduction to JWT
- Authentication with NgRx
- Using the routing activation condition.
- Preventing cross-site scripting (XSS)
- Sanitization and security contexts
- Use the offline template compiler
- Trusting safe values
- HTTP-level vulnerabilities
- Cross-site request forgery
- End to End demonstration
 - Using node.js as backend
 - (optional) Using Asp.Net Web API as backend
 - (optional) Using federated Service

Module 13 - Testing

- Testing overview
- End-to-End vs. Unit Testing
- Jasmine testing
- Class testing
- Pipe testing
- Async Service Testing
- The Angular Test Environment
- Component Testing
- Testing tools
- (optional) Advancing Unit Tests
- (optional) Reactive Testing Strategies with NgRx

Appendix (optional)

- IDE
 - plugins in Visual Studio Code
- Performance
 - Best Practices
 - Functions v.s. Pipes influence on speed.
- Mobile
 - Ionic
 - Nativescript
 - Progressive Web Apps (PWAs) with Angular
- WebPack
 - Entry
 - Output
 - Loaders
 - Plugins
 - Configuration
 - Modules
 - Targets
 - The Manifest
 - Redux
 - Redux basics
 - Store
 - Reducers
 - Middlewares
 - Redux Observable