Course : Developing Full JavaScript Web applications

Practical course - 4d - 28h00 - Ref. FUJ
Price : 2580 CHF E.T.

★★★★★ **5 / 5**

You will learn how to develop multi-page or single-page Web applications in JavaScript. The frameworks used will be Angular for the client side and Node.js for the server side. NoSQL databases will be covered, with MongoDB and other useful frameworks such as Express JS.

Teaching objectives

At the end of the training, the participant will be able to:

- Master the concepts of asynchronous, functional and reactive Javascript programming.
- Develop a SPA client application with Angular and Material Design.
- Develop a server application and REST API in Javascript with Node, Express.
- Add a data persistence layer using a NoSQL approach with MongoDB.
- Deploy a Full Javascript application in production using Docker.

Intended audience
Developers and project managers.

Prerequisites
Good knowledge of the fundamentals of the JavaScript language (in particular the notion of callback) as well as strong notions of HTML and CSS.

Practical details

Exercise
A "red thread" application will be built, gradually enriching it with the concepts covered.

Teaching methods
A "red thread" application will be built, gradually enriching it with the concepts covered.

Course schedule

PARTICIPANTS
Developers and project managers.

PREREQUISITES
Good knowledge of the fundamentals of the JavaScript language (in particular the notion of callback) as well as strong notions of HTML and CSS.

TRAINER QUALIFICATIONS
The experts leading the training are specialists in the covered subjects. They have been approved by our instructional teams for both their professional knowledge and their teaching ability, for each course they teach. They have at least five to ten years of experience in their field and hold (or have held) decision-making positions in companies.

ASSESSMENT TERMS
The trainer evaluates each participant's academic progress throughout the training using multiple choice, scenarios, hands-on work and more.
Participants also complete a placement test before and after the course to measure the skills they've developed.

1. A reminder of the basics

- Prototype, hoisting, ES5/6, binding, get/set.
- Asynchronous programming: event loop, blocking call, promise, async-await.
- GUI primitives (DOM access, event management, ajax).
- Reactive programming: observable, subjet, behaviorSubject (RxJS).
- Typescript: typed and modular programming. Decorators. Generics.

Hands-on work
Web application that displays the contents of a BehaviorSubject in real time (e.g. stopwatch).

2. Node.js

- Commands: Node, npm, npx.
- Files: package.json, node_modules, etc.
- Global and local modules, semver versions.
- Asynchronous Node PLC contract.
- Practical tools: Nodemon, ts-node, webpack.

Hands-on work
Installation d'un environnement de développement. Construction d'un site http « hello-world » en JS, sans express.

3. ExpressJS

- RequestListener, Middleware, Middleware chaining.
- Building an application skeleton.
- Express and application configuration.
- Rendering views with EJS.
- Setting up a REST API: CRUD on http.
- Implementation of an express server in Typescript.

Hands-on work
Creation of an express server with EJS templates and a basic JSON web service API, and a REST API.

4. Angular

- Digestion, interest in observables, behaviorSubject.
- Responsive routes and forms.
- http requests.
- Using off-the-shelf layouts and widgets: Angular Material.
- Construction of the production deliverable.
- Starting a project with Angular cli.
- Artifact presentation.
- Databinding concept.

Hands-on work
Creation of an Angular SPA client using Angular Material to perform a CRUD on a resource.

(5) MongoDB

- Mongo installation, normal and via docker.
- Installation of Mongo compass (intuitive client).
- Exploring Mongo concepts: collections, documents, etc.
- CRUD on Mongo from Node.
- REST API with Middleware express exposing a Mongo collection.

Hands-on work
Add a complete persistence layer to the previous application.

(6) Full JavaScript application deployment

- Deployed application architecture: reverse proxy, HTTPS, load balancer.
- List of the main deployment methods: Docker/Kubernetes, PM2, BaaS (Netlify, Firebase, etc.).
- Making a Full Javascript application deployable: using environment variables. dotenv library.
- PM2 production management tools.
- PM2 ecosystem file: ecosystem.config.js.

Hands-on work
Use PM2 to start the previous application in production mode.