This guide is in development. If a section is unclear, if you think a step is missing or wrong, if the estimated time seems excessive or scarce, or if you consider the jump from a step to the next too large, please contact me by email or skype

Training owner: [daniel.correatucunduva@cgi.com](mailto:daniel.correatucunduva@cgi.com)

My own live version of the application is available for browsing here (use Chrome)

<https://sprint-ng.herokuapp.com>

Read this full document before starting.

The time estimates take into consideration time to research topics if necessary.

You can develop the frontend and backend sequentially or in parallel, up to personal taste. For complete newcomers I recommend focusing on Angular alone for the first few days. Use placeholder data until you have a backend. If you already have some MEAN or related experience, parallel development is likely more efficient.

# Environment setup (estimated time: 30min)

Install the latest version unless noted otherwise

* NodeJS
* MongoDB and MongoDB Compass
* Angular cli
* Git
* Visual Studio Code (optional, recommended)
* Chrome

# Create an Angular application (estimated time: 30min)

Create an Angular cli application, serve it and browse it. Look at the folder and file structure, and browse the files in the root folder.

# Set up GitHub (estimated time: 15min)

Create a project on GitHub, and link your local environment to it.

# Add a component (estimated time: 15min)

Create a simple ‘Welcome’ component using the Angular cli scaffolding, and nest it inside your root component.

# Implement routing (estimated time: 30min)

Replace the content of your root component with router output, and set up basic routing.

# Plan remaining components (estimated time: 30min)

Take some time to think about how you will organize your components. Browse the live application and take notes on what you think is the optimal folder structure.

# Create scaffolding for remaining components (estimated time: 1h)

Create all the remaining components (with no content) you think will be necessary.

For at least one of them, create the individual folder(s) and files by hand, without using the cli.

# Create data models (estimated time: 30min)

Create all the models you think are adequate. Place them in the appropriate folders.

# Create services (estimated time: 30min)

Create the scaffold for all services you think are adequate. Place them in the appropriate folders.

# Fill out, wire and style components and services (estimated time: 30h)

Develop the components one by one. Consult the official Angular documentation and other resources and learn as you go. Create new / delete components and make any other adjustments you deem necessary. The services can return placeholder data for the time being.

Implement all frontend functionality fully.

*For Ville de Montréal consultants: use Bootstrap for styling, and fully comply with all documentation received.*

# Create an ExpressJS backend API (estimated time: 20h)

Develop the backend with Express. Use a tiered structure:

* server root script (to be executed by Node)
* routes
* controllers
* services
* models

*For Ville de Montréal consultants: fully comply with all documentation received.*

# Connect to Mongo (estimated time: 2h)

Set up a local Mongo database and connect your Express API to it.

Eliminate any placeholder data from source code.

# Deploy (optional, time permitting)

Deploy to the server of your choice.

Recommended: Heroku offers automated deployment directly from GitHub and automated linkage to a partner MongoDB provider. There is a free tier for all necessary services.

# Lint and clean up (estimated time: 3h)

Prepare your source code for reviewing.

*For Ville de Montréal consultants: consult the documentation you received and ensure full compliance.*

# Functional requirements

Develop the backend with Express. Use a tiered structure:

* server root script (to be executed by Node)
* routes
* controllers
* services
* models

*For Ville de Montréal consultants: fully comply with all documentation received.*

# Technical requirements