

Angular tutorial

Configure and run the server:

```
cd server
npm install
npm start
```

Mission for Step 1

Install angular cli

```
npm install -g @angular/cli
```

Connect and play with mongodb shell v3.6

```
mongo --host 129.194.184.60 --port 27017
use order
db.orders.find()
```

Create a new angular project

```
ng new orders
```

Mission for Step 2

run the angular project

```
cd orders
ng serve
```


then:

- add a **vendors component** (`ng generate component vendors`),
- use a **title binding property** and declare a list of strings with fake vendors' title,
- in the related template, expose the title as a html title (**h1**) and place the list of strings inside a bullet list (**ul/li**).

Mission for Step 3

- create a repository service helping to access vendors' data,
- use an abstraction of the service and a fake in-memory implementation.

Figure 1 shows an example.



Welcome to Funny WebApp!

hello vendors

- Le petit marché
- le vendeur du dimanche
- Youpie, vous êtes les bienvenus
- Bonne nuit, les petits

Figure 1: Goal for step 3

Mission for Step 4

- do a better abstraction for the order repository by taking into account the latency of such a service (use Observable with rxjs)
- use a Rest implementation for accessing data with the `HttpClientModule` to your Rest Api. (return result from `/vendors` as a list of string)

Figure 2 shows an example.



Welcome to Funny WebApp!

hello vendors

- boulangerie-leonhard-bretzel
- casa-mozzarella
- la-ferme-a-roulettes
- les-fromages-de-gaetan
- messapia-tesori-del-salento
- boucherie-jacky-bula
- crocorient
- les-potagers-de-gaia
- sandrine-suv-nrducteurs

Figure 2: Goal for step 4

Mission for Step 5

- use a `VendorDetail` class using title, categories and products fields
- do routes for `/vendorsdetails` and `/vendorsdetails/:id` in `OrderRepositoryService`
- transform the vendors component to use a set of cards
- create a `VendorCard` component to add more details on a specific vendor. Use a badge for the sum of commands per vendors and a list of products

Figure 3 shows an example.

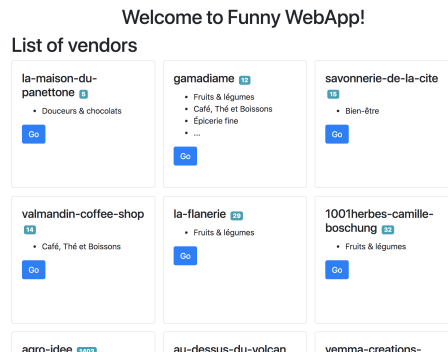


Figure 3: Goal for step 5

Mission for Step 6

- route to a **VendorDetails** component when we select a vendor
- use a **pipe** to transform the vendor id to a capitalized title (this-is-a-shop become This Is A Shop)

Figures 4 and 5 shows an example.

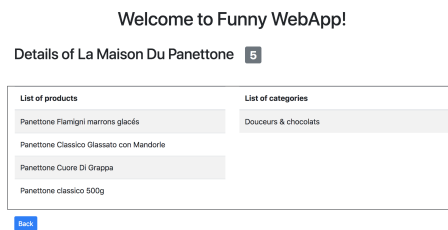


Figure 4: Goal for step 6 (VendorDetails view)

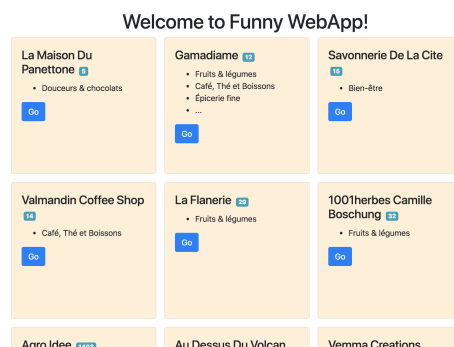


Figure 5: Goal for step 6 (Main view with piped title)