

Modernization of the mortgage application demo using Angular 2

These steps assume you are familiar enough with your environment to install Docker. www.docker.com will have a download suitable for your platform.

It is also assumed that you have deployed a copy of the mortgage application somewhere, or that you will be using our hosted version at <https://mort.cfapps.io/servlet>

Finally, you should be comfortable with editing code files in your editor of choice (the steps will use Visual Studio code) and using the terminal on your operating system (this guide is based on OSX)

Download the source code files from

here: http://www.elasticcobol.com/downloads/demo/Angular2/source_code.zip

Download the docker configuration files from

here: http://www.elasticcobol.com/downloads/demo/Angular2/docker_config_files.zip

Install Docker on your system:



Open a terminal window and create a directory to store this example in. CD into this directory:

```
Ians-iMac:demos ianw$ clear

Ians-iMac:demos ianw$ mkdir angulardemo
Ians-iMac:demos ianw$ cd angulardemo/
Ians-iMac:angulardemo ianw$
```

Copy the Initial-Dockerfile from the docker configuration files you downloaded to a file called Dockerfile:

```
Ians-iMac:demos ianw$ clear

Ians-iMac:demos ianw$ mkdir angulardemo
Ians-iMac:demos ianw$ cd angulardemo/
Ians-iMac:angulardemo ianw$ docker build -t angular-cli-v1 .
```

Download the initial angular and nodes base image using the following command:
`docker build -t angular-cli-v1 .`

Expect this step to take a few minutes as it downloads a lot of node files:

```
Ians-iMac:demos ianw$ mkdir angulardemo
Ians-iMac:demos ianw$ cd angulardemo/
Ians-iMac:angulardemo ianw$ cp ../docker_config_files/Initial-Dockerfile Dockerfile
Ians-iMac:angulardemo ianw$ docker build -t angular-cli-v1 .
```

When finished you'll see similar to this:

```
angulardemo -- bash -- 92x29
|   +-- camelcase@3.0.0
|   +-- os-locale@1.4.0
|   +-- string-width@1.0.2
|   |   +-- code-point-at@1.1.0
|   |   |   |-- is-fullwidth-code-point@1.0.0
|   |   |   |   |-- number-is-nan@1.0.1
|   |   +-- which-module@1.0.0
|   |   |-- yargs-parser@4.2.1
|   +-- webpack-merge@4.1.0
|   |-- zone.js@0.8.17
|
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@^1.0.0 (node_modules/@angular/cli/node_modules/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
npm info ok
npm info it worked if it ends with ok
npm info using npm@3.10.10
npm info using node@v6.10.3
npm info ok
----> ad913870338a
Removing intermediate container 4940e88ea971
Step 6/6 : EXPOSE 4200
----> Running in 0839edc34d9f
----> 3f85764d4611
Removing intermediate container 0839edc34d9f
Successfully built 3f85764d4611
Successfully tagged angular-cli-v1:latest
Ians-iMac:angulardemo ianw$
```

No we have the standard container we can create a specific container for our application. Enter the following command (you may need to substitute \$PWD based on your OS , the following string works exactly as-is on OSX):

```
docker run -it --rm --name create-angular-project -v "$PWD":/home/app angular-cli-v1 ng new -si mortang
```

You'll see something similar to this:

```
angulardemo -- bash -- 92x29
create mortang/src/environments/environment.ts (387 bytes)
create mortang/src/favicon.ico (5430 bytes)
create mortang/src/index.html (294 bytes)
create mortang/src/main.ts (370 bytes)
create mortang/src/polyfills.ts (2480 bytes)
create mortang/src/styles.css (80 bytes)
create mortang/src/test.ts (1085 bytes)
create mortang/src/tsconfig.app.json (211 bytes)
create mortang/src/tsconfig.spec.json (304 bytes)
create mortang/src/typings.d.ts (104 bytes)
create mortang/e2e/app.e2e-spec.ts (289 bytes)
create mortang/e2e/app.po.ts (208 bytes)
create mortang/e2e/tsconfig.e2e.json (235 bytes)
create mortang/karma.conf.js (923 bytes)
create mortang/package.json (1312 bytes)
create mortang/protractor.conf.js (722 bytes)
create mortang/tsconfig.json (363 bytes)
create mortang/tslint.json (3040 bytes)
create mortang/src/app/app.module.ts (314 bytes)
create mortang/src/app/app.component.css (0 bytes)
create mortang/src/app/app.component.html (1075 bytes)
create mortang/src/app/app.component.spec.ts (986 bytes)
create mortang/src/app/app.component.ts (207 bytes)
create mortang/.angular-cli.json (1125 bytes)
create mortang/src/assets/.gitkeep (0 bytes)
create mortang/.editorconfig (245 bytes)
Successfully initialized git.
Project 'mortang' successfully created.
Ians-iMac:angulardemo ianw$
```

cd into the new application directory:

```
Successfully initialized git.
Project 'mortang' successfully created.
Ians-iMac:angulardemo ianw$ ng
-bash: ng: command not found
Ians-iMac:angulardemo ianw$ cd mortang/
Ians-iMac:mortang ianw$
```

Copy the app-Dockerfile from the docker configuration files you downloaded to a file called Dockerfile:

```
Successfully initialized git.
Project 'mortang' successfully created.
Ians-iMac:angulardemo ianw$ ng
-bash: ng: command not found
Ians-iMac:angulardemo ianw$ cd mortang/
Ians-iMac:mortang ianw$ cp ../../docker_config_files/app-Dockerfile Dockerfile
Ians-iMac:mortang ianw$
```

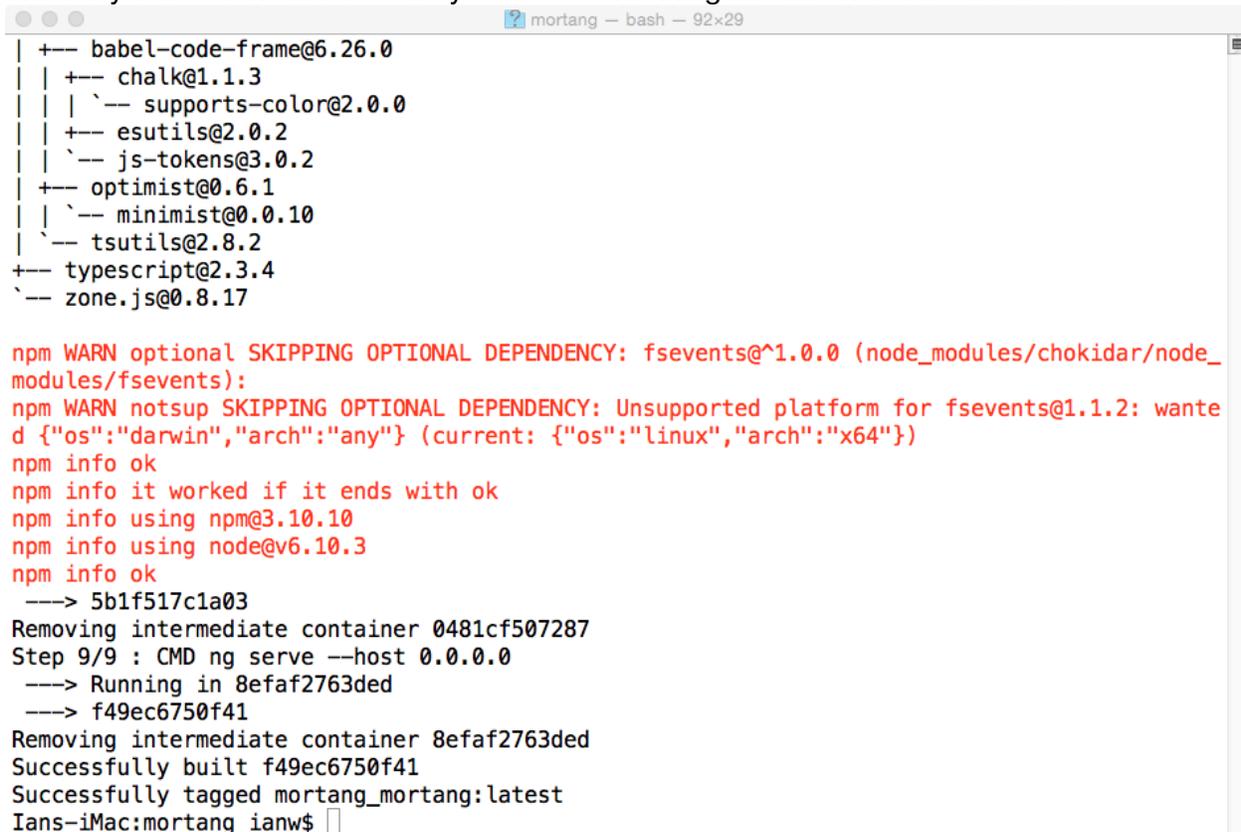
Copy the docker-compose.yml from docker configuration files you downloaded into this directory:

```
Successfully initialized git.
Project 'mortang' successfully created.
Ians-iMac:angulardemo ianw$ ng
-bash: ng: command not found
Ians-iMac:angulardemo ianw$ cd mortang/
Ians-iMac:mortang ianw$ cp ../../docker_config_files/app-Dockerfile Dockerfile
Ians-iMac:mortang ianw$ cp ../../docker_config_files/docker-compose.yml .
Ians-iMac:mortang ianw$
```

Run the following command to build the application container:

```
docker-compose build
```

This may take a few minutes and you'll see something like this when it's finished:



```
mortang - bash - 92x29
| +-- babel-code-frame@6.26.0
| | +-- chalk@1.1.3
| | | `-- supports-color@2.0.0
| | +-- esutils@2.0.2
| | `-- js-tokens@3.0.2
| +-- optimist@0.6.1
| | `-- minimist@0.0.10
| `-- tsutils@2.8.2
+-- typescript@2.3.4
`-- zone.js@0.8.17

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@^1.0.0 (node_modules/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
npm info ok
npm info it worked if it ends with ok
npm info using npm@3.10.10
npm info using node@v6.10.3
npm info ok
----> 5b1f517c1a03
Removing intermediate container 0481cf507287
Step 9/9 : CMD ng serve --host 0.0.0.0
----> Running in 8efaf2763ded
----> f49ec6750f41
Removing intermediate container 8efaf2763ded
Successfully built f49ec6750f41
Successfully tagged mortang_mortang:latest
Ians-iMac:mortang ianw$
```

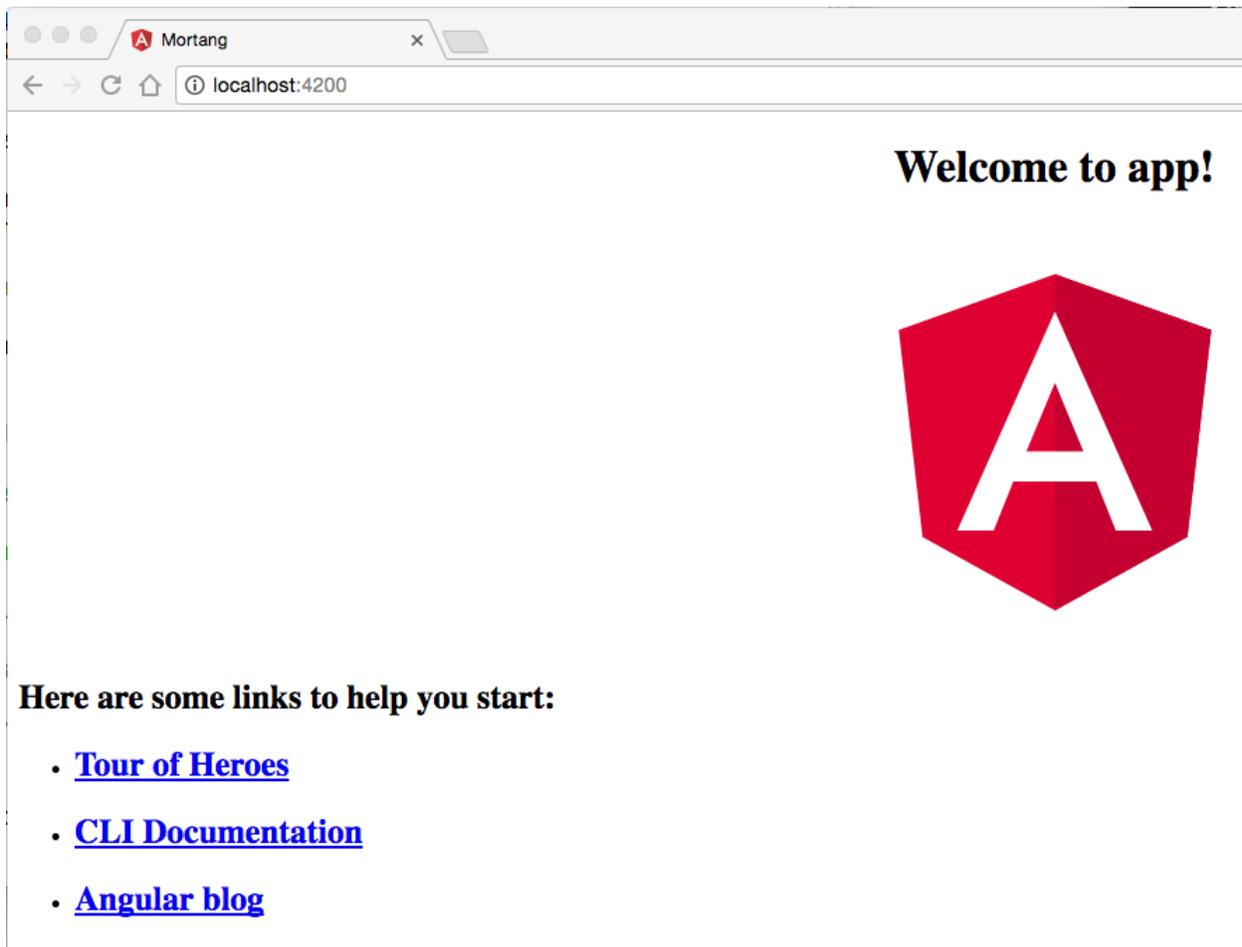
Run the following command to start the container:

```
docker-compose up
```

You'll see the following once the container has started:

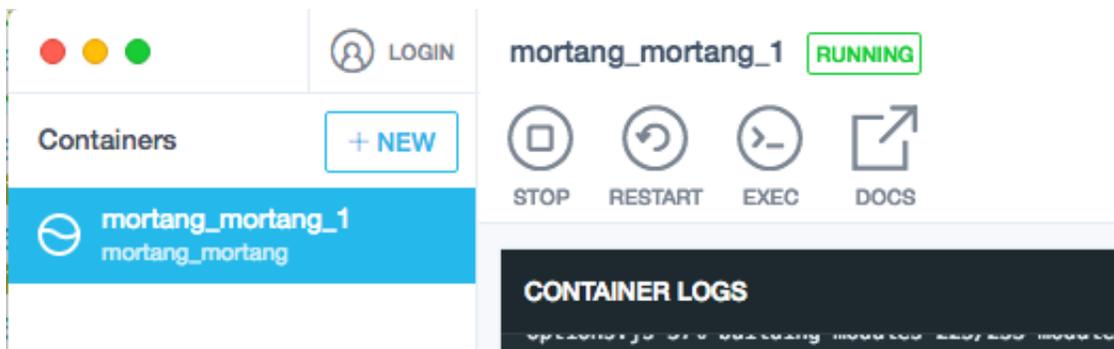
```
mortang -- docker-compose -- 92x29
Removing intermediate container 0481cf507287
Step 9/9 : CMD ng serve --host 0.0.0.0
----> Running in 8efaf2763ded
----> f49ec6750f41
Removing intermediate container 8efaf2763ded
Successfully built f49ec6750f41
Successfully tagged mortang_mortang:latest
Ians-iMac:mortang ianw$ docker-compose up
Creating mortang_mortang_1 ...
Creating mortang_mortang_1 ... done
Attaching to mortang_mortang_1
mortang_1 | ** NG Live Development Server is listening on 0.0.0.0:4200, open your browser o
n http://localhost:4200/ **
morDate: 2017-09-08T20:01:18.458Z
mortang_1 | Hash: eeeaaa28f5226904ece0
mortang_1 | Time: 6891ms
mortang_1 | chunk {inline} inline.bundle.js, inline.bundle.js.map (inline) 5.83 kB [entry]
[rendered]
mortang_1 | chunk {main} main.bundle.js, main.bundle.js.map (main) 8.58 kB {vendor} [initialia
l] [rendered]
mortang_1 | chunk {polyfills} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 209
kB {inline} [initial] [rendered]
mortang_1 | chunk {styles} styles.bundle.js, styles.bundle.js.map (styles) 11.3 kB {inline}
[initial] [rendered]
mortang_1 | chunk {vendor} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.28 MB [initial
] [rendered]
mortang_1 |
mortang_1 | webpack: Compiled successfully.
```

Open a browser tab and navigate to <http://localhost:4200>:



Now we've confirmed the default app works. press Ctrl+C to bring down the container as we'll need that command prompt soon.

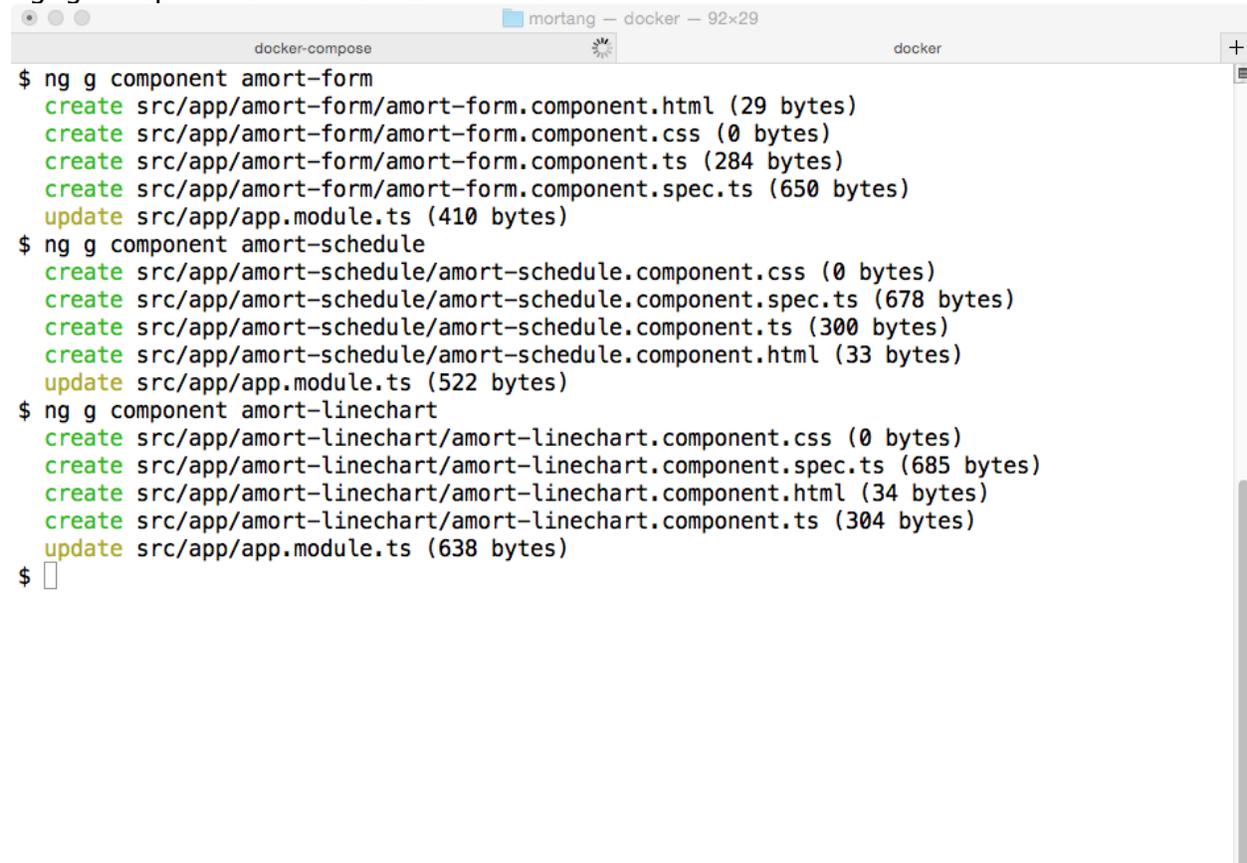
Using your Docker client open up a command line window in the container. For Docker for Mac that involves opening Kinematic ,selecting the container and clicking 'Exec':



Once you have the prompt open you'll need to create some new code to hold the GUI components.

Type the following commands into the container:

```
ng g component amort-form
ng g component amort-schedule
ng g component amort-linechart
```

A screenshot of a terminal window titled "mortang - docker - 92x29". The terminal shows the execution of three Angular CLI commands to generate components. The first command, "ng g component amort-form", results in four files being created (component.html, component.css, component.ts, component.spec.ts) and the app.module.ts file being updated. The second command, "ng g component amort-schedule", results in four files being created (component.css, component.spec.ts, component.ts, component.html) and the app.module.ts file being updated. The third command, "ng g component amort-linechart", results in four files being created (component.css, component.spec.ts, component.html, component.ts) and the app.module.ts file being updated. The terminal prompt "\$" is visible at the beginning and end of the output.

```
$ ng g component amort-form
create src/app/amort-form/amort-form.component.html (29 bytes)
create src/app/amort-form/amort-form.component.css (0 bytes)
create src/app/amort-form/amort-form.component.ts (284 bytes)
create src/app/amort-form/amort-form.component.spec.ts (650 bytes)
update src/app/app.module.ts (410 bytes)
$ ng g component amort-schedule
create src/app/amort-schedule/amort-schedule.component.css (0 bytes)
create src/app/amort-schedule/amort-schedule.component.spec.ts (678 bytes)
create src/app/amort-schedule/amort-schedule.component.ts (300 bytes)
create src/app/amort-schedule/amort-schedule.component.html (33 bytes)
update src/app/app.module.ts (522 bytes)
$ ng g component amort-linechart
create src/app/amort-linechart/amort-linechart.component.css (0 bytes)
create src/app/amort-linechart/amort-linechart.component.spec.ts (685 bytes)
create src/app/amort-linechart/amort-linechart.component.html (34 bytes)
create src/app/amort-linechart/amort-linechart.component.ts (304 bytes)
update src/app/app.module.ts (638 bytes)
$
```

Now type the following to create the service that handles the communication with the backend service

and some classes to hold the inputs and outputs:

```
ng g service amort
ng g class amort-inputs
ng g class amort-item
```

When you are done your src/app directory should look like this:

```
$ ls -las src/app
total 32
0 drwxr-xr-x 15 app app 510 Sep 12 20:51 .
0 drwxr-xr-x 14 app app 476 Sep  8 19:50 ..
0 drwxr-xr-x  7 app app 238 Sep 12 20:42 amort-form
4 -rw-r--r--  1 app app  29 Sep 12 15:09 amort-inputs.ts
4 -rw-r--r--  1 app app 150 Sep 12 20:42 amort-item.ts
0 drwxr-xr-x  6 app app 204 Sep 12 20:42 amort-line-chart
0 drwxr-xr-x  6 app app 204 Sep  8 20:06 amort-linechart
0 drwxr-xr-x  6 app app 204 Sep  8 20:06 amort-schedule
4 -rw-r--r--  1 app app 368 Sep 12 20:51 amort.service.spec.ts
4 -rw-r--r--  1 app app 111 Sep 12 20:51 amort.service.ts
0 -rw-r--r--  1 app app  0 Sep 12 20:42 app.component.css
4 -rw-r--r--  1 app app  35 Sep 12 20:42 app.component.html
4 -rw-r--r--  1 app app 1000 Sep 12 20:42 app.component.spec.ts
4 -rw-r--r--  1 app app 292 Sep 12 20:42 app.component.ts
4 -rw-r--r--  1 app app 763 Sep 12 20:42 app.module.ts
$
```

Before we continue with copying source code we will ensure the form looks a little better when displayed.

Type the following to install the twitter bootstrap styles

```
npm install bootstrap@next
```

You'll see something like the following:

```

npm info using node@v6.10.3
npm info attempt registry request try #1 at 3:22:58 PM
npm http request GET https://registry.npmjs.org/bootstrap
npm http 200 https://registry.npmjs.org/bootstrap
npm info addNameTag [ 'bootstrap', 'next' ]
npm info retry fetch attempt 1 at 3:23:01 PM
npm info attempt registry request try #1 at 3:23:01 PM
npm http fetch GET https://registry.npmjs.org/bootstrap/-/bootstrap-4.0.0-beta.tgz
npm http fetch 200 https://registry.npmjs.org/bootstrap/-/bootstrap-4.0.0-beta.tgz
npm info attempt registry request try #1 at 3:23:05 PM
npm http request GET https://registry.npmjs.org/fsevents
npm http 200 https://registry.npmjs.org/fsevents
npm info lifecycle bootstrap@4.0.0-beta~preinstall: bootstrap@4.0.0-beta
npm info linkStuff bootstrap@4.0.0-beta
npm info lifecycle bootstrap@4.0.0-beta~install: bootstrap@4.0.0-beta
npm info lifecycle bootstrap@4.0.0-beta~postinstall: bootstrap@4.0.0-beta
mortang@0.0.0 /home/app/mortang
+-- bootstrap@4.0.0-beta
+-- UNMET PEER DEPENDENCY jquery@>=3.0.0
`-- UNMET PEER DEPENDENCY popper.js@^1.11.0

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@^1.0.0 (node_modules/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
npm WARN bootstrap@4.0.0-beta requires a peer of jquery@>=3.0.0 but none was installed.
npm WARN bootstrap@4.0.0-beta requires a peer of popper.js@^1.11.0 but none was installed.
npm info ok
$

```

Switch to the command prompt you used earlier that you brought the container up and down in:

```

^CGracefully stopping... (press Ctrl+C again to force)
Killing mortang_mortang_1 ... done
Ians-iMac:mortang ianw$

```

edit `.angular-cli.json` and replace the lines;

```

"defaults": {
  "styleExt": "css",
  "component": {}
}

```

with the follow lines:

```

"defaults": {
  "styleExt": "css",

```

```
    "component": {}
  },

  "scripts": [
    "../node_modules/jquery/dist/jquery.js",
    "../node_modules/tether/dist/js/tether.js",
    "../node_modules/bootstrap/dist/js/bootstrap.js"
  ]
}
```

Line 22 is 'styles.css' :

```
"styles": [
  "styles.css"
],
```

Insert the following:

```
"../node_modules/bootstrap/dist/css/bootstrap.min.css",
```

So it looks like this:

```
"styles": [
  "../node_modules/bootstrap/dist/css/bootstrap.min.css",
  "styles.css"
],
```

Save and exit the file.

Since the other code changes are quite significant we'll copy those in from the downloaded source code folder. You can use a diff tool to see what changes are made, or simply view the sources files.

Copy the source_code folder you downloaded into the root of your application (this should overwrite the src folder):

```

Ians-iMac:mortang ianw$ ls -las
total 88
0 drwxr-xr-x  17 ianw  staff   578 Sep 12 15:37 .
0 drwxr-xr-x   7 ianw  staff   238 Sep 12 15:42 ..
8 -rw-r--r--   1 ianw  staff  1365 Sep 12 10:34 .angular-cli.json
8 -rw-r--r--   1 ianw  staff   245 Sep  8 14:50 .editorconfig
0 drwxr-xr-x  13 ianw  staff   442 Sep 12 10:05 .git
8 -rw-r--r--   1 ianw  staff   516 Sep  8 14:50 .gitignore
8 -rw-r--r--   1 ianw  staff   316 Sep  8 14:56 Dockerfile
8 -rw-r--r--   1 ianw  staff  1098 Sep  8 14:50 README.md
8 -rw-r--r--   1 ianw  staff   128 Sep  8 14:57 docker-compose.yml
0 drwxr-xr-x   5 ianw  staff   170 Sep  8 14:50 e2e
8 -rw-r--r--   1 ianw  staff   923 Sep  8 14:50 karma.conf.js
0 drwxr-xr-x   2 ianw  staff    68 Sep  8 15:01 node_modules
8 -rw-r--r--   1 ianw  staff  1393 Sep 12 15:37 package.json
8 -rw-r--r--   1 ianw  staff   722 Sep  8 14:50 protractor.conf.js
0 drwxr-xr-x  14 ianw  staff   476 Sep  8 14:50 src
8 -rw-r--r--   1 ianw  staff   363 Sep  8 14:50 tsconfig.json
8 -rw-r--r--   1 ianw  staff  3040 Sep  8 14:50 tslint.json
Ians-iMac:mortang ianw$ cp -R ../source_code/ .
Ians-iMac:mortang ianw$ ls -las
total 88
0 drwxr-xr-x  17 ianw  staff   578 Sep 12 15:37 .
0 drwxr-xr-x   7 ianw  staff   238 Sep 12 15:42 ..
8 -rw-r--r--   1 ianw  staff  1365 Sep 12 10:34 .angular-cli.json
8 -rw-r--r--   1 ianw  staff   245 Sep  8 14:50 .editorconfig
0 drwxr-xr-x  13 ianw  staff   442 Sep 12 10:05 .git
8 -rw-r--r--   1 ianw  staff   516 Sep  8 14:50 .gitignore
8 -rw-r--r--   1 ianw  staff   316 Sep  8 14:56 Dockerfile
8 -rw-r--r--   1 ianw  staff  1098 Sep  8 14:50 README.md
8 -rw-r--r--   1 ianw  staff   128 Sep  8 14:57 docker-compose.yml
0 drwxr-xr-x   5 ianw  staff   170 Sep  8 14:50 e2e
8 -rw-r--r--   1 ianw  staff   923 Sep  8 14:50 karma.conf.js
0 drwxr-xr-x   2 ianw  staff    68 Sep  8 15:01 node_modules
8 -rw-r--r--   1 ianw  staff  1393 Sep 12 15:37 package.json
8 -rw-r--r--   1 ianw  staff   722 Sep  8 14:50 protractor.conf.js
0 drwxr-xr-x  14 ianw  staff   476 Sep  8 14:50 src
8 -rw-r--r--   1 ianw  staff   363 Sep  8 14:50 tsconfig.json
8 -rw-r--r--   1 ianw  staff  3040 Sep  8 14:50 tslint.json
Ians-iMac:mortang ianw$ █

```

NOTE: You should NOT have a source_code folder in your application folder!

Edit the file `src/app/amort.service.ts` and ensure line 18 (the private url) points to the correct migrated application (you can use ours or your own if you've followed earlier demos):

```
import { Injectable } from '@angular/core';
import { Http, Headers, Response, URLSearchParams } from
import 'rxjs/add/operator/map';
import { AmortItem } from './amort-item';
import { AmortInputs } from './amort-form/amort-inputs';
```

```
@Injectable()
export class AmortService {

  constructor(private http: Http) {}

  private inputbody = 'transid=inpt&DFH_CURSOR=MONTH&MON
YEARS=30'; //just for testing

  private url = 'https://mort.cfapps.io/servlet'

  private headers = new Headers({ 'Content-Type': 'appli
true });
```

Save the file as required.

Remove the existing container by issuing the following command:

```
docker rm mortang_mortang_1
```

Now rebuild the containing using the following command:

```
docker-compose build
```

And finally bring the container up using the following command:

```
docker-compose up
```

Navigate to a localhost:4200 in your browser and you should see the new look application:

Amortization Form

Start Date

Loan Amount

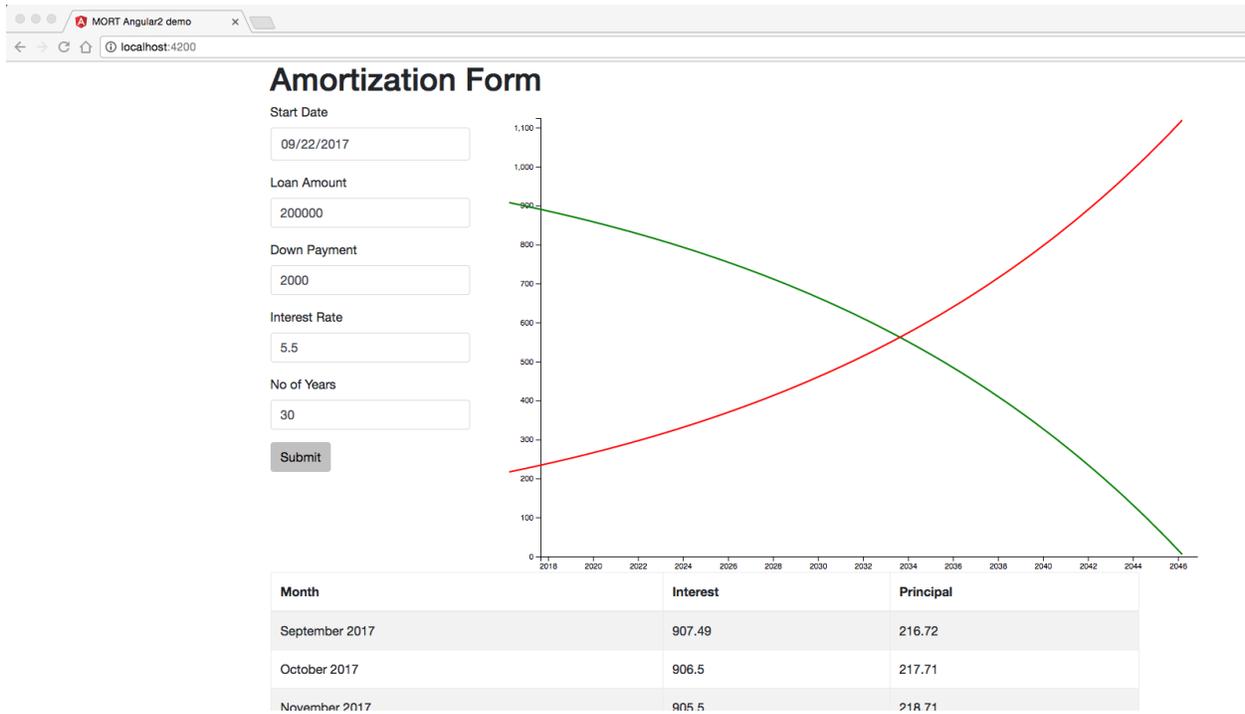
Down Payment

Interest Rate

No of Years

Submit

Select a date and other values and press Submit:



And here you can see a simple but new front end on a migrated application using Node.JS and Angular-2!