

TD3

Introduction

L'objectif de ce TP est de créer une galerie d'images qui sera déployée par la suite en suivant une architecture micro services.

Partie 1 – « Hello World » Node.js + Express avec Docker

Création du « Hello World »

1)

Nous initialisons un projet nodejs dans un nouveau répertoire : gallery.

On initialise Express :

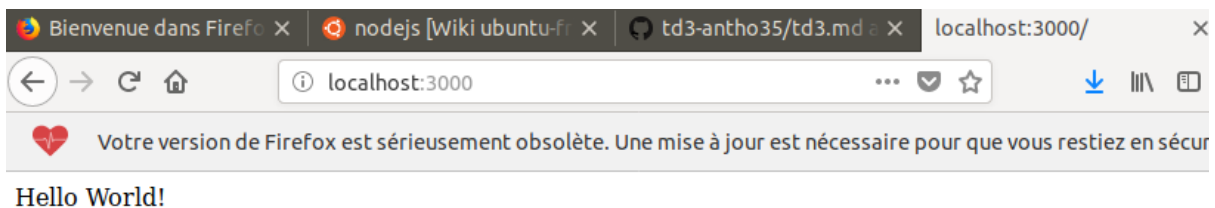
```
ubuntu@ubuntu:~/Documents/Cloud/TD3-ORAIN/gallery$ npm install express
npm WARN saveError ENOENT: no such file or directory, open '/home/ubuntu/Documents/Cloud/TD3-ORAIN/gallery/package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open '/home/ubuntu/Documents/Cloud/TD3-ORAIN/gallery/package.json'
npm WARN gallery No description
npm WARN gallery No repository field.
npm WARN gallery No README data
npm WARN gallery No license field.

+ express@4.16.4
added 48 packages from 36 contributors and audited 121 packages in 4.655s
found 0 vulnerabilities
```

2)

On crée un fichier index.js permettant de créer un « hello word » et écoutant sur le port 3000.

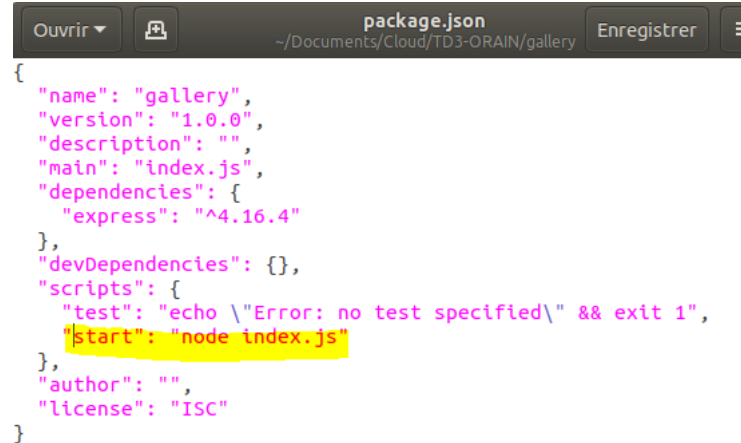
```
ubuntu@ubuntu:~/Documents/Cloud/TD3-ORAIN/gallery$ touch index.js
ubuntu@ubuntu:~/Documents/Cloud/TD3-ORAIN/gallery$ node index.js
App listening on http://localhost:3000!
Client requested /
```



The screenshot shows a Firefox browser window with the address bar set to 'localhost:3000'. The page content displays 'Hello World!'. The browser's status bar at the bottom indicates that the version of Firefox is outdated.

3)

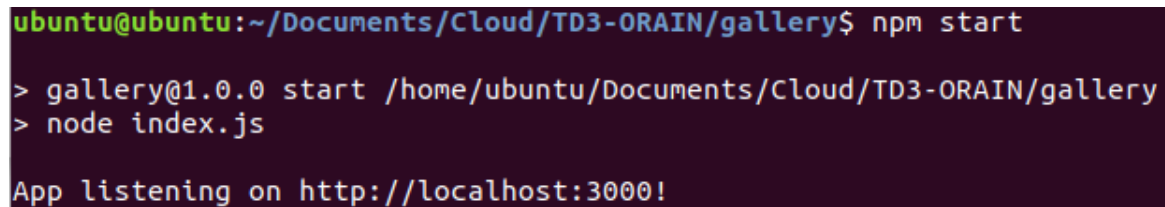
On modifie le fichier package.json pour lancer le script index.js à l'aide de la commande « npm start ».



```
{
  "name": "gallery",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "dependencies": {
    "express": "^4.16.4"
  },
  "devDependencies": {},
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node index.js"
  },
  "author": "",
  "license": "ISC"
}
```

Les modifications du package.json sont surlignées.

On peut ensuite lancer le script :



```
ubuntu@ubuntu:~/Documents/Cloud/TD3-ORAIN/gallery$ npm start

> gallery@1.0.0 start /home/ubuntu/Documents/Cloud/TD3-ORAIN/gallery
> node index.js

App listening on http://localhost:3000!
```

Conteneurisation de notre app

1) La taille de l'image node du TD1 est importante. Pour réduire au maximum ce volume, on peut prendre la version alpine, qui est une version allégée.

On prend donc la version 8.15.0-alpine

2)

On remplit le fichier Dockerfile de la manière suivante :

Dockerfile	×	index.js	×	package
<pre>FROM node:8.15.0-alpine # Create app directory WORKDIR /usr/src/app # Install app dependencies # A wildcard is used to ensure both package.json AND package-lock.json are copied # where available (npm@5+) COPY package*.json ./ RUN npm install # If you are building your code for production # RUN npm install --only=production # Bundle app source COPY . . EXPOSE 3000 CMD ["npm", "start"]</pre>				

puis pour créer notre image, on lance la commande :

```
orain@orain-TM1701:~/Documents/td3-antho35/td3/gallery$ sudo docker build -t mynodeimage .
Sending build context to Docker daemon 1.915MB
Step 1/7 : FROM node:8.15.0-alpine
8.15.0-alpine: Pulling from library/node
cd784148e348: Pull complete
```

```
Step 7/7 : CMD [ "npm", "start" ]
---> Running in 87eb9afafad6
Removing intermediate container 87eb9afafad6
---> 597ddd228c98
Successfully built 597ddd228c98
Successfully tagged mynodeimage:latest
```

On peut désormais lancer notre image :

```

orain@orain-TM1701:~/Documents/td3-antho35/td3/gallery$ sudo docker run --rm --name mynodeapp -p 3000:3000 mynodeimage
[sudo] Mot de passe de orain :
> gallery@1.0.0 start /usr/src/app
> node index.js

App listening on http://localhost:3000!

```

On obtient bien le même résultat que précédemment lorsque l'on visite <http://localhost:3000>.

Partie 2 – Création d'une galerie d'images statique

J'ai par la suite suivi le tutoriel : <https://stonesoupprogramming.com/2017/07/04/create-node-js-handlebars-express-js-project-from-command-line/> afin de générer un projet nodejs avec Express et le moteur de templates handlebars.

```

orain@orain-TM1701:~/Documents/td3-antho35/td3/gallery/app$ sudo npm install express-generator -g
[sudo] Mot de passe de orain :
/usr/bin/express -> /usr/lib/node_modules/express-generator/bin/express-cli.js
+ express-generator@4.16.0
added 10 packages from 13 contributors in 0.465s

```

```

orain@orain-TM1701:~/Documents/td3-antho35/td3/gallery/app$ express --hbs myhbsapp

warning: option '--hbs' has been renamed to '--view=hbs'

create : myhbsapp/
create : myhbsapp/public/
create : myhbsapp/public/javascripts/
create : myhbsapp/public/images/
create : myhbsapp/public/stylesheets/
create : myhbsapp/public/stylesheets/style.css
create : myhbsapp/routes/
create : myhbsapp/routes/index.js
create : myhbsapp/routes/users.js
create : myhbsapp/views/
create : myhbsapp/views/error.hbs
create : myhbsapp/views/index.hbs
create : myhbsapp/views/layout.hbs
create : myhbsapp/app.js
create : myhbsapp/package.json
create : myhbsapp/bin/
create : myhbsapp/bin/www

```

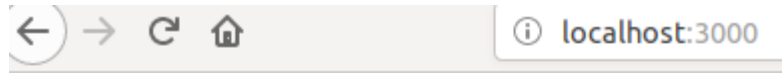
```

orain@orain-TM1701:~/Documents/td3-antho35/td3/gallery/app$ cd myhbsapp
orain@orain-TM1701:~/Documents/td3-antho35/td3/gallery/app/myhbsapp$ npm install
npm notice created a lockfile as package-lock.json. You should commit this file.
added 79 packages from 100 contributors in 11.658s
orain@orain-TM1701:~/Documents/td3-antho35/td3/gallery/app/myhbsapp$ DEBUG=myhbsapp:* npm start

> myhbsapp@0.0.0 start /home/orain/Documents/td3-antho35/td3/gallery/app/myhbsapp
> node ./bin/www

myhbsapp:server Listening on port 3000 +0ms

```

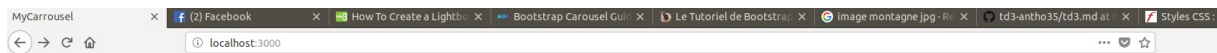


Express

Welcome to Express

J'ai par la suite modifié mon fichier package.json de façon à pouvoir lancer mon app à l'aide de la commande « npm run devstart ». Chaque modification est désormais apparente sans avoir besoin d'avoir à relancer le serveur.

J'ai suivi par la suite un tutoriel bootstrap qui permet de faire un carrousel avec lightbox.



Second slide

On peut à présent créer notre image « carouselimg » :

```
orain@orain-TM1701:~/Documents/td3-antho35/td3/gallery/app/myhbsapp$ sudo docker build -t carouselimg .
Sending build context to Docker daemon 14.44MB
Step 1/7 : FROM node:8.15.0-alpine
--> 5c0c5c94503f
Step 2/7 : WORKDIR /usr/src/app
--> Using cache
--> ba4f2ff94537
Step 3/7 : COPY package*.json ./
--> d1d3939dd367
Step 4/7 : RUN npm install
--> Running in 59174fbb2537
```



```
Step 7/7 : CMD [ "npm", "start" ]
---> Running in 1f187d49ea7d
Removing intermediate container 1f187d49ea7d
---> a43532b975ed
Successfully built a43532b975ed
Successfully tagged carouselimg:latest
```

Puis on la lance :

```
orain@orain-TM1701:~/Documents/td3-antho35/td3/gallery/app/myhbsapp$ sudo docker run --rm --name carouselapp -p 3000:3000 carouselimg
> myhbsapp@0.0.0 start /usr/src/app
> node ./bin/www

GET / 304 49.723 ms - -
GET /stylesheets/style.css 200 3.341 ms - 221
GET / 304 3.276 ms - -
```

On obtient le même résultat que précédemment lorsque l'on visite <http://localhost:3000> :

