

# Le framework Express.js

## Objectif :

Cet atelier consiste à générer la structure d'un projet Express.js à l'aide **d'express generator**.

## Générateur d'applications Express :

- Utilisez l'outil de générateur d'applications, express, pour créer rapidement un squelette d'application.
- Installez express à l'aide de la commande suivante :

```
$ npm install express-generator -g
```

- Affichez les options de commande à l'aide de l'option **-h** :

```
$ express -h

Usage: express [options][dir]

Options:

  -h, --help            output usage information
  -V, --version          output the version number
  -e, --ejs              add ejs engine support (defaults to jade)
  --hbs                  add handlebars engine support
  -H, --hogan            add hogan.js engine support
  -c, --css <engine>    add stylesheet <engine> support (less|stylus|compass|sass) (defaults to plain css)
  --git                  add .gitignore
  -f, --force            force on non-empty directory
```

- Ce qui suit crée une application Express nommée **myapp** dans le répertoire de travail en cours :

```
$ express myapp

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

- Ensuite, installez les dépendances :

```
$ cd myapp
$ npm install
```

- Sous MacOS ou Linux, exécutez l'application à l'aide de la commande suivante :

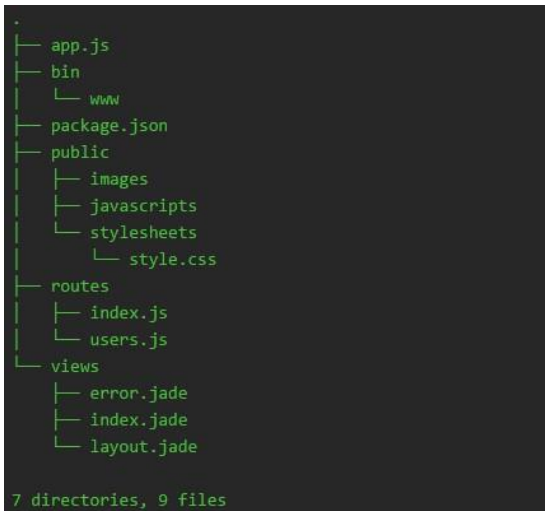
```
$ DEBUG=myapp:* npm start
```

- Sous Windows, utilisez la commande suivante :

```
> set DEBUG=myapp:* & npm start
```

- Ensuite, chargez 'http://localhost:3000/' dans votre navigateur pour accéder à l'application.

L'application générée possède la structure de répertoire suivante :



**NB** : La structure d'application créée par le générateur est l'une des nombreuses manières possibles de structurer les applications Express. Vous avez toute latitude pour l'utiliser ou la modifier en fonction de vos besoins.

## Exercice:

Cet exercice va fournir des données JSON aux différentes routes.

1. Générez le squelette d'une application Express en utilisant [express-generator](#).

Regardez comment fonctionne les routes déjà existantes (folder */routes*).

Regardez dans *app.js* comment ces routes sont renseignées.

2. [Ajoutez une route](#) à votre projet: */os/*

- 2.1 Créez un fichier *os.js* sous */routes*

Pensez à utiliser le module *os* : voir [documentation](#) pour récupérer les données relatives à votre OS.

- 2.2 La réponse à retourner est sous format JSON

```
res.json({
  hostname: os.hostname(),
  type: os.type(),
  platform: os.platform(),
});
```

Affiche les informations sur le serveur.

**Réponse:**

```
{"hostname": "HP-PC", "type": "Windows_NT", "platform": "win32"}
```

3. Ajoutez la route: */os/cpus/* : Affiche la liste des processeurs.

Réponse :

```
[
  {
    "model": "Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz",
    "speed": 775,
    "times": {
      "user": 14505600,
      "nice": 448800,
      "sys": 3623300,
      "idle": 118165900,
      "irq": 600
    }
  },
  {
    "model": "Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz",
    "speed": 775,
    "times": {
      "user": 17399600,
      "nice": 27600,
      "sys": 3071400,
      "idle": 63390400,
      "irq": 0
    }
  },
  {
    "model": "Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz",
    "speed": 775,
    "times": {
      "user": 8651200,
      "nice": 179800,
      "sys": 1996200,
      "idle": 66876600,
      "irq": 0
    }
  },
  {
    "model": "Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz",
    "speed": 775,
    "times": {
      "user": 9568500,
      "nice": 44500,
      "sys": 1791900,
      "idle": 67082400,
      "irq": 0
    }
  }
]
```

4. Ajoutez la route: **/os/cpus/:id** : Affiche les informations relative à un seul processeur.

**Exemple : /os/cpus/0 Réponse :**

```
{
  "model": "Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz",
  "speed": 775,
  "times": {
    "user": 14579800,
    "nice": 448800,
    "sys": 3641000,
    "idle": 118828400,
    "irq": 600
  }
}
```

5. Nous allons maintenant implémenter un autre contrôleur avec Express pour une gestion de stocks : **/products/**

6. Créez la route : **/products/** : Affiche la liste des produits depuis **products.json** au format JSON.

Réponse :

```
{
  "MACBOOKPRO": {
    "name": "Macbook Pro",
    "manufacturer": "Apple",
    "price": 1299,
    "stock": 32,
    "options": ["Intel Core i5", "Retina Display", "Long life battery"]
  },
  "MACBOOKAIR": {
    "name": "Macbook Pro Air",
    "manufacturer": "Apple",
    "price": 1099.99,
    "ultrabook": true,
    "stock": 112,
    "options": ["Intel Core i7", "SSD", "Long life battery"]
  },
  "LENOVOX230": {
    "name": "Thinkpad X230",
    "manufacturer": "Lenovo",
    "price": 1099.99,
    "ultrabook": true,
    "stock": 0,
    "options": ["Intel Core i5", "SSD", "Long life battery"]
  }
}
```

7. Ajoutez la route : **/products/:id** : Affiche les détails du produit associé à id.

Exemple : /products/LENOVOX230

Réponse :

```
{
  "name": "Thinkpad X230",
  "manufacturer": "Lenovo",
  "price": 1099.99,
  "ultrabook": true,
  "stock": 0,
  "options": [
    "Intel Core i5",
    "SSD",
    "Long life battery"
  ]
}
```

8. Ajoutez la route : **/products/:id/:qt** : Affiche le prix total pour **qt** fois le prix du produit **id**.

Exemple : /products/LENOVOX230/10 Réponse :

```
{
  "id": "LENOVOX230",
  "qt": 10,
  "unit_price": 1099,
  "total_price": 10990
}
```

9. Ajoutez la route : **/products/instock/:qt** : Afficher seulement les produits dont le stock est supérieur ou égal à **qt**.

Exemple : /products/instock/100

Réponse :

```
[
  {
    "name": "Macbook Pro Air",
    "manufacturer": "Apple",
    "price": 1099.99,
    "ultrabook": true,
    "stock": 112,
    "options": [
      "Intel Core i7",
      "SSD",
      "Long life battery"
    ]
  }
]
```