

# EXPRESS.JS

---

# Express JS

- Module Node.js, minimal et flexible
- Facilite le développement sur Node.js (abstraction code bas niveau)
- Facilite l'organisation de l'application Node.js
- Installation
  - `npm install express`

# Hello World

- Créer un fichier app.js

```
var express = require('express');
var app = express();

app.get('/', function (req, res) {
  res.send('Hello World!');
});

app.listen(3000, function () {
  console.log('Example app listening on port 3000!');
});
```

- Comparer code avec/sans Express.js

# Routage (1)

- Structure définition d'une route

```
app.METHOD(PATH, HANDLER)
```

Où :

- `app` est une instance d'`express`.
- `METHOD` est une `méthode de demande HTTP`.
- `PATH` est un chemin sur le serveur.
- `HANDLER` est la fonction exécutée lorsque la route est mise en correspondance.

- méthode GET sur la page racine

```
app.get('/', function (req, res) {
  res.send('Hello World!');
});
```

# Routage (2)

- demande POST sur la page d'accueil

```
app.post('/', function (req, res) {
  res.send('Got a POST request');
});
```

- demande PUT sur le chemin /user

```
app.put('/user', function (req, res) {
  res.send('Got a PUT request at /user');
});
```

- demande DELETE sur le chemin /user

```
app.delete('/user', function (req, res) {
  res.send('Got a DELETE request at /user');
});
```

# Routage(3)

- Gérer les erreurs 404

```
// ... Tout le code de gestion des routes (app.get) se trouve au-dessus

app.use(function(req, res, next){
  res.setHeader('Content-Type', 'text/plain');
  res.send(404, 'Page introuvable !');
});

app.listen(8080);
```

- Chaîner les appels

```
app.get('/', function(req, res) {

})
.get('/sous-sol', function(req, res) {

})
.get('/etage/1/chambre', function(req, res) {

})
.use(function(req, res, next){

});
```

# Route dynamique

- Ecrire **:nomvariable** dans l'URL
- Paramètre accessible par **req.params.nomvariable**

```
app.get('/etage/:etagenum/chambre', function(req, res) {  
  res.setHeader('Content-Type', 'text/plain');  
  res.end('Vous êtes à la chambre de l\'étage n°' + req.params.etagenum);  
});
```

- Toutes les routes suivantes sont valables
  - /etage/1/chambre
  - /etage/2/chambre
  - /etage/3/chambre
  - /etage/sosisy/chambre

# Route dynamique

- On peut utiliser des expressions régulières

```
app.get('/ab?cd', function(req, res) {  
  res.send('ab?cd');  
});
```

```
app.get('/ab+cd', function(req, res) {  
  res.send('ab+cd');  
});
```

```
app.get('/ab*cd', function(req, res) {  
  res.send('ab*cd');  
});
```

```
app.get('/ab(cd)?e', function(req, res) {  
  res.send('ab(cd)?e');  
});
```

```
app.get(/a/, function(req, res) {  
  res.send('/a/');  
});
```

```
app.get('.*fly$', function(req, res) {  
  res.send('/.*fly$/');  
});
```

# TP 1

- Ré écrire l'application nodebeginner en utilisant Express.js
  - Utiliser les routages ExpressJs
- Installer nodemon avec `npm install -g nodemon`
  - lancer `nodemon index.js`
  - le serveur node redémarre automatiquement à chaque changement de source

# Moteur de template JADE

- npm install jade
- Rajouter code : app.set('view engine', 'jade'); pour spécifier le moteur de templates
- Les vues par défaut se trouve dans le répertoire views
  - fichier views/index.jade

```
html
  head
    title= title
  body
    h1= message
```

- Utiliser res.render pour appeler la vue et spécifier des paramètres

```
app.get('/', function (req, res) {
  res.render('index', { title: 'Hey', message: 'Hello there!'});
});
```

# Jade exemple

```
doctype html
html(lang="en")
  head
    title= pageTitle
    script(type='text/javascript').
      if (foo) {
        bar(1 + 5)
      }
  body
    h1 Jade - node template engine
    #container.col
      if youAreUsingJade
        p You are amazing
      else
        p Get on it!
    p.
      Jade is a terse and simple
      templating language with a
      strong focus on performance
      and powerful features.
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Jade</title>
    <script type="text/javascript">
      if (foo) {
        bar(1 + 5)
      }
    </script>
  </head>
  <body>
    <h1>Jade - node template engine</h1>
    <div id="container" class="col">
      <p>You are amazing</p>
      <p>
        Jade is a terse and simple
        templating language with a
        strong focus on performance
        and powerful features.
      </p>
    </div>
  </body>
</html>
```

# Jade

- Interactive documentation
  - <http://naltatis.github.io/jade-syntax-docs/>

# Générateur Application Express

- Pour créer une squelette d'application
- Installation avec : `$ npm install express-generator -g`
- Générer application avec: `express nomapp`
- install dependencies:
  - `$ cd nomapp&& npm install`
- run the app:
  - `$ DEBUG=nomapp:* npm start`