Express 4.14.0

Fast, unopinionated, minimalist web framework for Node.js

en moins de 10 minutes

Web Applications

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

APIs

With a myriad of HTTP utility methods and middleware at your disposal, creating a robust API is quick and easy.

Performance

Express provides a thin layer of fundamental web application features, without obscuring Node.js features that you know and love.

Frameworks

Many popular frameworks are based on Express.

Documentation translations provided by StrongLoop/IBM: French, German, Spanish, Italian, Japanese, Russian, Chinese, Traditional Chinese, Korean, Portuguese.
Community translation available for: Slovak, Ukrainian and Uzbek.

Une application serveur

```
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!')
})
```

Démarrage

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

On démarre l'application sur la machine sur le port 3000.

Accessible via :

> http://192.168.1.10:3000

Première route

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

Pour chaque requête http GET *req* effecuté à la racine '/',
envoyer la réponse *res* qui contient 'Hello World'.

```
$ curl http://192.168.1.10:3000/
Hello World!
```

Avec ajax et $

```
$.ajax({
  url : 'http://192.168.1.10:3000/',
  method: 'GET'
})
.done(function(res){
        console.log(res);
        console.log('Fini.');
})
.fail(function(){
        console.log('Erreur...');
});
```

On résume

```
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!')
})
```

```
$ curl http://192.168.1.10:3000/
Hello World
```

Routage

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

app.get('/aurevoir', function (req, res) {
  res.send('Bye bye!')
})
```

```
$ curl http://192.168.1.10:3000/
Hello World!

$ curl http://192.168.1.10:3000/aurevoir
Bye bye!
```

POST

```
var bodyParser = require('body-parser');
app.use(bodyParser.json()); // parsing application/json

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

app.post('/newuser', function (req, res) {
  // On récupère les paramètres de la requête
  var body = req.body;
  /*
   body = {
        prenom : 'Jean',
        nom : 'Patrick'
   }
  */
  res.send('Bye bye!')
})
```

Chez notre client

```
$.ajax({
  url : 'http://192.168.1.10:3000/newUser',
  method: 'POST',
  data : {
    prenom: 'Jean',
    nom : 'Patrick'
  }
})
.done(function(){
        console.log('Fini.');
})
.fail(function(){
        console.log('Erreur...');
});
```

Les réponses

Classique

```
app.get('/aurevoir', function (req, res) {
  res.send('Bye bye!')
})
```

Json

```
app.get('/aurevoir', function (req, res) {
  var monObjet = {
        status: 'ok',
        msg: "Tout va bien!"
  };

  res.json(monObjet)
})
```

To sum up

- Express Js : application de seveur web (sous NodeJs)
- des routes pour savoir où écouter (et comment)

```
app.get('/aurevoir', function (req, res) {
  res.send('Bye bye!')
})
```

```
app.post('/newuser', function (req, res) {
  // ... req.body pour accéder au corps de la requete
  res.send('Bye bye!')
})
```

- des réponses

```
app.get('/aurevoir', function (req, res) {
  res.json({
    msg : "Je suis une réponse en json",
    success: true
  })
})
```