

Introducción a express

Manuel Molino Milla

29 de noviembre de 2017

Índice

1. Introducción	2
1.1. ¿Qué es express?	2
1.2. Instalación	2
1.3. Hello World con express	2
2. Enrutador con express	3
2.1. Ejemplo	3
2.2. Sirviendo páginas estáticas	5
3. Ejercicio	6

1. Introducción

1.1. ¿Qué es express?

- Es un framework para desarrollo web.
- Facilita el desarrollo web con *node.js*
- Entre sus características están:
 - Permite crear rutas en función de peticiones web o métodos HTTP.
 - Configura *middlewares* a las peticiones web.
 - Permite renderizar dinámicamente HTML mediante argumentos pasados a plantillas.

1.2. Instalación

```
npm install express
```

Instala localmente en el directorio *node_modules* Es interesante acompañar express con otros módulos como:

body-parser es un middleware para datos JSON, Raw, Text y URL encoded

cookie-parser para establecer *cookies*

Y muchos mas.

1.3. Hello World con express

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

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

var server = app.listen(8081, function () {
  var host = server.address().address
  var port = server.address().port

  console.log("Example app listening at 8081")
})
```

2. Enrutador con express

2.1. Ejemplo

```
var express = require('express');
var app = express();
var bodyParser = require('body-parser');

// parse application/x-www-form-urlencoded
app.use(bodyParser.urlencoded({ extended: false }));

// parse application/json
app.use(bodyParser.json());
app.set('port', process.env.PORT || 3000);

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

app.get('/users/:userName', function(request, response) {
  var name = request.params.userName;
  response.send('¡Hola, ' + name + '!');
});

app.post('/users', function(request, response) {
  var username = request.body.username;
  response.send('¡Hola, ' + username + '!');
});

//http://localhost:3000/personal/15
//http://localhost:3000/personal/edit/15
app.get(/\//personal\/(\d*)\/?(edit)?(\//(\d*))?/, function (request, response) {
  var message = 'el perfil del empleado #' + request.params[0];
  if (request.params[1] === 'edit') {
    message = 'Editando ' + message;
  } else {
    message = 'Viendo ' + message;
  }
  response.send(message);
});

var server = app.listen(app.get('port'), function () {
  var host = server.address().address
  console.log("Example app listening at http://%s:%s", host, app.get('port'))
});
```

})

Usamos como cliente algún plugin para un navegador como puedes ser rest-client, postman o rested

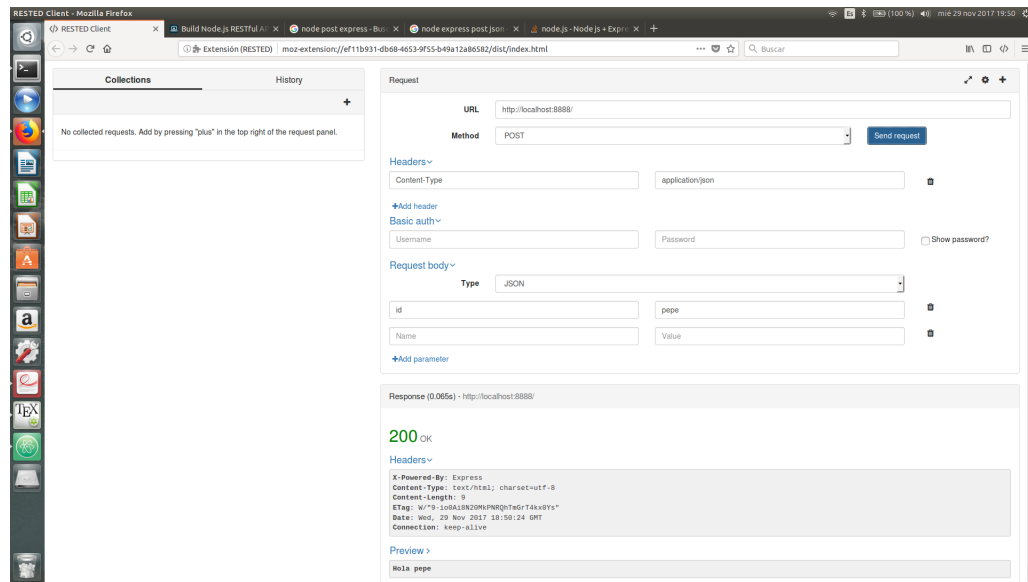


Figura 1: RESTED Cliente

2.2. Sirviendo páginas estáticas

Ejemplo *index.html*

```
<html>
  <body>
    <form action = "http://127.0.0.1:8081/process_get" method = "GET">
      First Name: <input type = "text" name = "first_name"> <br>
      Last Name: <input type = "text" name = "last_name">
      <input type = "submit" value = "Submit">
    </form>
  </body>
</html>
```

Presentando y procesando el formulario:

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

app.use(express.static('public'));
app.get(/\/(index.html)?$/ , function (req, res) {
  res.sendFile( __dirname + "/" + "index.html" );
})

app.get('/process_get', function (req, res) {
  // Prepare output in JSON format
  response = {
    first_name:req.query.first_name,
    last_name:req.query.last_name
  };
  console.log(response);
  // res.end(JSON.stringify(response));
  res.json(response);
})

var server = app.listen(8081, function () {
  var host = server.address().address
  var port = server.address().port
  console.log("Example app listening at http://%s:%s", host, port)
})
```