

## Ejercicios Tema 2

En el anterior ejercicio ya dimos los primeros pasos en Node.js, y ahora vamos a ver una introducción a distintos sistemas de plantillas.

### Ejercicio 1

Antes de nada, vamos a necesitar un node con express y algo de contenido para mostrar, por ejemplo una lista de nombres al entrar en una url en concreto. De esta manera veremos cómo confeccionar de manera correcta el manejador de rutas con Express y cómo mostrar los elementos en una lista.

En primer lugar crearemos el proyecto como vimos en teoría:

```
$ sudo express imagina_todo
```

Una vez creado, entramos a la carpeta de nuestro proyecto e instalamos las dependencias con:

```
$ cd imagina_todo
```

```
$ sudo npm install
```

Ya tenemos listo nuestro proyecto. En primer lugar vamos a borrar ficheros que no nos serán necesarios para nuestra aplicación.

Primero borramos el fichero `users.js` de la carpeta `routes` y las siguientes líneas de nuestro fichero `app.js`:

```
var users = require('./routes/users');  
app.use('/users', users);
```

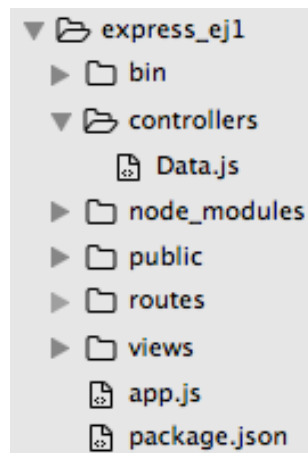
Después de eliminar estos ficheros y líneas debería funcionar correctamente, para comprobarlo:

```
$ npm start
```

El servidor debe ponerse en marcha y acceder a él a través del navegador con la URL `localhost:3000` y visualizar la página de entrada de *Express*.

Ahora es el momento de crear una lista de nombres con una función para devolver estos nombres.

Crearemos la carpeta *controllers* en la raíz del proyecto. Dentro de ésta crearemos el fichero `Data.js`:



Vamos a modificar este fichero Data.js, en él simplemente crearemos un Array y una función que devuelve un Array de la siguiente manera:

```
var nombres = ["Mario", "Patricia", "Manolo", "Santi",  
"David", "Oscar", "Luis", "Saul", "Autor"];  
function getNames() {  
    return nombres;  
}  
module.exports = {  
    getNames: getNames  
}
```

Ya tenemos nuestro “modelo”, ahora vamos a plantear nuestras vistas con SWIG/TWIG y con Handlebars para mostrar el siguiente contenido HTML:

```
<html>  
<head>  
    <meta charset="utf-8">  
    <title>Hello World - Personas</title>  
</head>  
<body>  
    <h1>Hello Swig</h1>  
    <p> Welcome to Hello Swig</p>  
    <p>for:</p>  
    <ol>  
        <li>Mario</li>  
        <li>Patricia</li>  
        <li>Manolo</li>  
        <li>Santi</li>  
        <li>David</li>  
        <li>Oscar</li>  
        <li>Luis</li>  
        <li>Saul</li>  
        <li>Autor</li>  
    </ol>  
</body>  
</html>
```

## Ejercicio2, SWIG:

**Swig** es posiblemente el sistema de plantillas más sencillo que tenemos en nodejs, ya que no nos obliga a aprender nueva sintaxis si ya sabemos HTML, y aunque a día de hoy ya no tiene mantenimiento oficial sigue siendo una opción válida para desarrollos rápidos.

Antes de nada debemos instalar Swig en nuestro proyecto, lo haremos con *npm*:

```
$ npm install swig
```

Después tendremos que redefinir el sistema de plantillas que vamos a utilizar ya que por defecto express viene con jade. Modificaremos las siguientes líneas en *app.js*:

```
// Jade
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');
```

Por las siguientes:

```
//Swig
var swig = require('swig');
swig.setDefaults({ cache: false });
app.engine('html', swig.renderFile);
app.set('views', __dirname + '/views');
app.set('view engine', 'html');
```

En tercer lugar vamos a crear nuestro archivo de plantilla. Para ello crearemos un fichero *index.html* en la carpeta */views* que se corresponderá con nuestro fichero *swig*. Tendrá la siguiente forma:

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>{{ title }} - Tareas</title>
  </head>
  <body>
    <h1>{{ title }}</h1>
    <p> Bienvenido a #{title}</p>
```

```
{#  
  Recorre el array y muestra la lista de valores con loop  
  Puedes hacer uso de loop.first y loop.last  
#}  
</body>  
</html>
```

Por último deberemos de editar el archivo routes/index.js para que nos cargue la información que necesitamos:

```
var express = require('express');  
var router = express.Router();  
var data = require('../controllers/Data');  
  
/* GET home page. */  
router.get('/', function(req, res, next){  
  res.render('index', {  
    title : "Hello Swig", nombres: data.getNames()  
  });  
});  
  
module.exports = router;
```

A estas alturas sólo nos quedaría enlazar los datos con la interfaz cuando se llame a /. Para ello tendremos que modificar el fichero *index.js* de la carpeta routes. Debemos indicar que cargue nuestro fichero *index.html* en con nuestro Array de tareas:

### Ejercicio3, TWIG:

Si vienes de **PHP** tienes la casi obligación de conocer **TWIG**, uno de sus sistemas de plantillas referencia, y que es prácticamente igual que SWIG (se parecen en mucho más que el nombre... de hecho SWIG nació como un fork de TWIG Php, y ahora que ya lo tenemos para node carece un poco de sentido seguir manteniendo los dos proyectos en paralelo).

Antes de nada debemos instalar Twig en nuestro proyecto, lo haremos con *npm*:

```
$ npm install twig
```

Después tendremos que redefinir el sistema de plantillas que vamos a utilizar ya que por defecto express viene con jade. Modificaremos las siguientes líneas en *app.js*:

```
// Jade
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');
```

Por las siguientes:

```
//Twig
var Twig = require("twig");
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'twig');
// This section is optional and used to configure
twig.
app.set("twig options", {
  strict_variables: false
});
```

El único cambio que tendremos que hacer respecto al ejercicio anterior es renombrar la plantilla `index.html` a *`index.twig`* en la carpeta `/views`.

