



Sistemas Distribuidos e Internet

Consumo de servicios web desde aplicaciones Node.js

Sesión- 11

Curso 2017/ 2018



Consumo de servicios Web rest en Node.js

Vamos a implementar un cliente REST dentro de la aplicación **tiendaMusica**. Este cliente va a utilizar el servicio web REST comercial <http://fixer.io/> que permite hacer transformaciones de divisas en base a la cotización real

Para realiza peticiones a un servicio REST vamos a utilizar el módulo **request**. Abrimos la consola de comandos, accedemos al directorio principal de la aplicación y ejecutamos el comando **npm install request --save**

```
C:\Users\jordansoy\work\TiendaMusica>npm install request --save
TiendaMusica@0.1.0 C:\Users\jordansoy\work\TiendaMusica
`-- request@2.83.0
   +-- aws-sign2@0.7.0
   +-- aws4@1.6.0
   +-- caseless@0.12.0
   +-- combined-stream@1.0.5
   | `-- delayed-stream@1.0.0
```

Incluimos el require del módulo en el fichero principal de **app.js**, guardamos el módulo en una variable de la aplicación con clave 'rest'.

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

var rest = require('request');
app.set('rest', rest);
```

Para ver el tipo de cambio entre EUR y USD utilizaremos el servicio **GET** <http://api.fixer.io/latest?symbols=USD> según la documentación nos retornará un objeto JSON.

Simplemente a modo de ejemplo (esto no emula un caso real, habría formas mejores de hacerlo como mostrar únicamente los precios en la divisa seleccionada por el usuario), vamos a incluir la posibilidad de poder los precios de una canción en **dólares**, dentro de la vista de detalles de canción.

Accedemos al controlador **rcanciones.js** y buscamos la función que responde a **GET /cancion/:id**

Una vez obtenemos la canción vamos a ofrecer su precio también en dólares.

```
▼ routes
  rapicanciones.js
  > rcanciones.js
  > rusuarios.js
```

Para ello vamos a recuperar el módulo almacenado en **app.get('rest')** y utilizarlo para obtener el cambio EUR -> USD en el servicio de fixer.



Debemos declarar un objeto de configuración especificando la información de la petición a realizar, **url**, **method**, **headers** (en caso de que tenga una header/cabecera, hemos incluido una a modo de ejemplo, pero realmente el servicio de fixer la procesa)

El módulo rest necesita dos parámetros para realizar la petición:

- El objeto de configuración
- La función de callback con parámetros (error, response, body).

Como casi todos los módulos que hemos visto hasta el momento este también funciona de forma **asíncrona**. Por lo tanto, debemos incluir el código que queremos que se ejecute posteriormente dentro de la función de callback que le enviamos como parámetro.

El parámetro **response** de la función de callback contiene la respuesta completa enviada por el servicio web fixer. Utilizando **response** podemos acceder a todos los elementos de la respuesta. El parámetro **body** nos da acceso al cuerpo de la respuesta, en este caso al objeto JSON que nos retorna <http://Fixer.io> aunque sepamos que la respuesta está en formato JSON debemos convertirla a un objeto JSON ya que se procesa inicialmente como un String

El objeto respuesta contiene atributos con claves: **base**, **date**, **rates**, siendo la variable **rates** de tipo **objeto { }** y teniendo una variable con nombre **USD** de tipo numérico.

```
{ "base": "EUR", "date": "2017-10-12", "rates": { "USD": 1.1856 } }
```

Vamos a obtener el cambio de divisas y multiplicar el precio actual por el cambio para obtener el precio en dólares, agregamos una nueva variable **usd** a la **cancion** que se envía a la vista **bcancion**.

```
app.get('/cancion/:id', function (req, res) {
  var criterio = { "_id" : gestorBD.mongo.ObjectId(req.params.id) };

  gestorBD.obtenerCanciones(criterio, function(canciones){
    if ( canciones == null ){
      res.send(respuesta);
    } else {

      var configuracion = {
        url: "http://api.fixer.io/latest?symbols=USD",
        method: "get",
        headers: {
          "token": "ejemplo",
        }
      }
      var rest = app.get("rest");
      rest(configuracion, function (error, response, body) {
        console.log("cod: "+response.statusCode+" Cuerpo :"+body);
        var objetoRespuesta = JSON.parse(body);
        var cambioUSD = objetoRespuesta.rates.USD;
        // nuevo campo "usd"
        canciones[0].usd = cambioUSD * canciones[0].precio;

        var respuesta = swig.renderFile('views/bcancion.html',
        {
          cancion : canciones[0]
        });
        res.send(respuesta);
      });
    }
  });
});
```

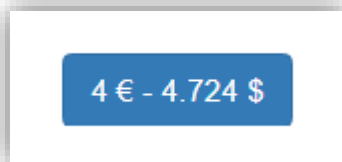


```
    })  
  
    var respuesta = swig.renderFile('views/bcancion.html',  
    {  
        cancion : canciones[0]  
    });  
    res.send(respuesta);  
  
    }  
});  
})
```

Finalmente mostramos el nuevo campo **cancion.usd** en la vista **views/bcancion**. Para mostrarlo de forma rápida lo incluimos a continuación del precio original en el mismo enlace.

```
/{{ cancion._id.toString() }}">{{ cancion.precio }} € - {{ cancion.usd }} $</a>
```

En la vista de detalles de las canciones los precios deberían verse de la siguiente forma:



Otros tipos de peticiones

En el ejemplo anterior solo hemos realizado una petición **get** pero módulo request, nos permite realizar todo tipo de peticiones, simplemente debemos cambiar el objeto configuración. A continuación, se muestra un ejemplo de una petición POST. Este tipo de peticiones contienen un **body** en un formato específico (JSON en el ejemplo).

```
// Insertar un anuncio  
var anuncio = {  
    descripcion : 'Nuevo anuncio',  
    precio : '10'  
};  
var configuracion = {  
    url: "http://ejemplo.ejemplo/anuncio",  
    method: "POST",  
    json: true,  
    headers: {  
        "content-type": "application/json",  
    },  
    body: anuncio  
}  
var rest = app.get("rest");  
rest(configuracion, function (error, response, body) {  
    ...  
})
```