

# Nodejs REST API

## Instalación y set up

### Instalación de Node:

Desde la página de Node (<https://nodejs.org/en/download/>) descargar la última versión disponible para la plataforma adecuada (para Windows es 8.12) e instalarla.

### Creación del proyecto:

- Crear una carpeta con el nombre deseado del proyecto y hacerla corriente.
- Crear un proyecto con npm : “>npm init “ completando los prompts a medida que se solicitan. Ninguno de los prompts es relevante para el funcionamiento del proyecto.
- Instalar las dependencias del proyecto (paquetes de node):
  - o “npm install mysql --save” (driver de mysql)
  - o “npm install express --save” (express router)
  - o “npm install path --save” (path resolver)
  - o “npm install url --save” (url resolver)
  - o “npm install body-parser --save” (body parser)
  - o “npm install moment --save” (date & time functions)
- Crear una subcarpeta “metadata”
- Crear una subcarpeta “src”

La estructura de carpetas deberá quedar (donde “root” es el nombre asignado a la carpeta del proyecto):

root

..

node\_modules (carpeta)

metadata (carpeta)

src (carpeta)

package.json

### **Instalación del programa:**

El programa consiste de un script principal (e-tangram.js) que deberá copiarse a la carpeta “root” y una serie de módulos javascript que deberán copiarse a la subcarpeta “src”.

La subcarpeta “metadata” contendrá los archivos de metadata descriptores de los recursos.

### **Instalación de pruebas:**

Descomprimir el archivo “metadata.zip” y copiar los .json a la subcarpeta “metadata”.

Deberá crearse una base de datos (schema) con el nombre “testapi”.

Descomprimir el archivo “dbscripts.zip” y ejecutar los scripts desde el MySQL Workbench. El archivo .zip contiene los scripts necesarios para la creación de las tablas de ejemplo que se corresponden con los archivos .json descriptores de la metadata.

El modelo de prueba contiene cinco tablas:

- categorías
- productos
- clientes
- remitos
- remitos\_items

las que deberán crearse en este orden para respetar la integridad referencial.

En este modelo se podrán probar los conceptos de unique, required (not null), primary key, foreign key, cascade delete y version (conurrencia), como también los distintos tipos de dato.

Todas las tablas tienen como PK una ID numérica (integer) autoincrementada.

### **Configuración:**

Configuración del script “e-tangram.js”, (el server).

Editar e-tangram.js y modificar los parámetros de conexión: host, database, user y password de acuerdo a las necesidades:

```
var dbConfig = {  
  host: "localhost",  
  database: "testapi",  
  user: "sa",  
  password: "mysqladmin",  
  multipleStatements: true,  
  supportBigNumbers: true,  
  dateStrings: true,  
  connectionLimit: 30  
}
```

En producción estas variables deberán guardarse en el “environment”.

El “port” por defecto donde “escucha” Node es 1337 y puede cambiarse a voluntad.

El programa e-tangram.js está ampliamente comentado.

### **Cliente:**

Para las pruebas se sugiere el uso de Postman (<https://www.getpostman.com/>).

### **Nota:**

4427 6845