



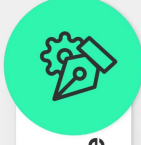
API RESTful

API RESTful

Formato: link a un repositorio en Github con el proyecto cargado.

Sugerencia: no incluir los node_modules

Desafío
entregable



>> Consigna: Realizar un proyecto de servidor basado en node.js y express que ofrezca una API RESTful de productos. En detalle, que incorpore las siguientes rutas:

- GET '/api/productos' -> devuelve todos los productos.
- GET '/api/productos/:id' -> devuelve un producto según su id.
- POST '/api/productos' -> recibe y agrega un producto, y lo devuelve con su id asignado.
- PUT '/api/productos/:id' -> recibe y actualiza un producto según su id.
- DELETE '/api/productos/:id' -> elimina un producto según su id.

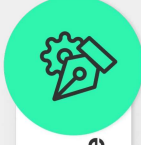
CODER HOUSE

API RESTful

Formato: link a un repositorio en Github con el proyecto cargado.

Sugerencia: no incluir los node_modules

Desafío
entregable



- Cada producto estará representado por un objeto con el siguiente formato:

```
{  
  title: (nombre del producto) ,  
  price: (precio) ,  
  thumbnail: (url al logo o foto del producto)  
}
```

- Cada ítem almacenado dispondrá de un id numérico proporcionado por el backend, comenzando en 1, y que se irá incrementando a medida de que se incorporen productos. Ese id será utilizado para identificar un producto que va a ser listado en forma individual.

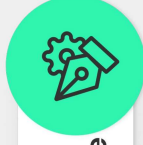
CODER HOUSE

API RESTful

Formato: link a un repositorio en Github con el proyecto cargado.

Sugerencia: no incluir los node_modules

Desafío
entregable



- Para el caso de que un producto no exista, se devolverá el objeto:
`{ error : 'producto no encontrado' }`
- Implementar la API en una clase separada, utilizando un array como soporte de persistencia en memoria.
- Incorporar el Router de express en la url base `'/api/productos'` y configurar todas las subrutinas en base a este.
- Crear un espacio público de servidor que contenga un documento `index.html` con un formulario de ingreso de productos con los datos apropiados.
- El servidor debe estar basado en express y debe implementar los mensajes de conexión al puerto 8080 y en caso de error, representar la descripción del mismo.
- Las respuestas del servidor serán en formato JSON. La funcionalidad será probada a través de Postman y del formulario de ingreso.

CODER HOUSE



¿PREGUNTAS?