

# Práctica Backend Avanzado

Bootcamp Web17 2024







La práctica se hará sobre el código que se creó para la práctica de fundamentos de Node.js.

NodePop

[New product](#) [Pricing](#) [FAQs](#) [About](#)

admin [Logout](#)

## My Products

Name	Price	Image	Tags	Actions
Bicicleta	230.15€		lifestyle, motor	
Iphone	50€		lifestyle, mobile	

total 2 products.

[prev](#) [next](#)

**En caso de no disponer de ese código podéis usar la práctica resuelta que se publicó en Fundamentos como punto de partida.**

Índice de retos:

1. Internacionalización
  2. Creación de un producto con imagen
  3. API REST
  4. [OPCIONAL] Servicio de creación de thumbnails en segundo plano
  5. [OPCIONAL] BONUS TRACK
- 

## Internacionalización

Este reto consiste en convertir el website de la aplicación Nodepop en multi-idioma.

Idiomas disponibles:

- Español
- Inglés

No será necesario internacionalizar el API.

Deberá disponer de un selector de idioma donde el usuario pueda cambiar de inglés a español o viceversa.

## Creación de un producto con imagen

El website debe incluir un link en el menú y una página para crear productos.

En la creación de un producto se podrá subir una imagen.

## API REST

Se le añadirá un API con los siguientes endpoints:

- POST /api/login
  - Permite hacer login en el API retornando un JWT
- GET /api/products
  - **Retorna una lista de productos** con algún filtro (p.e. por nombre), paginación, ordenación y selección de campos. Solo devuelve los productos que son propiedad del usuario logado.
- GET /api/products/<productID>

- **Retorna un producto**
- POST /api/products
  - **Crea un producto**, con subida de imagen
- PUT /api/products/<productID>
  - **Actualiza un producto**
- DELETE /api/products/<productID>
  - **Elimina un producto**

El API debe incluir documentación, al menos en el README, los más valientes pueden probar a hacerlo con [Swagger](#) en [Express](#) (OpenAPI).

El API inicialmente tendrá al menos dos usuarios:

- admin@example.com, clave 1234
- user1@example.com, clave 1234

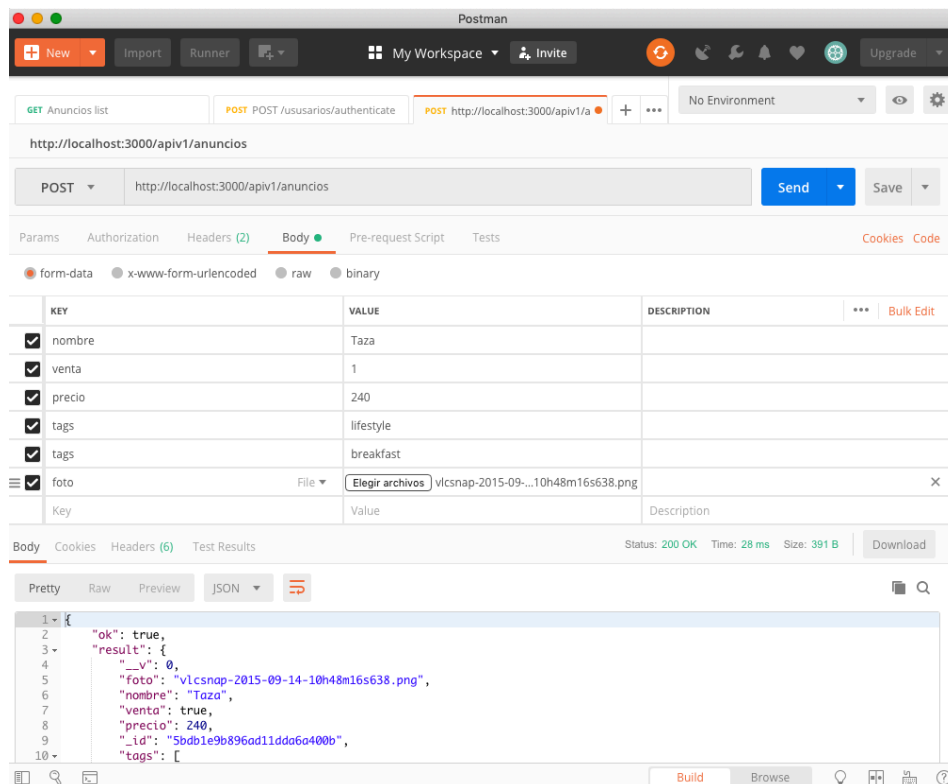
## [OPCIONAL] Servicio de creación de thumbnails en segundo plano

Cada imagen que se suba debe tener un thumbnail. Podemos hacer un microservicio que reciba trabajos (con cote.js o con RabbitMQ), creando dichos thumbnails.

Para crear un thumbnail podemos:

- buscar “node.js image resize” en google y ver cual probáis, por ejemplo [jimp](#)
- instalarla y probarla
- cuando se cree un nuevo producto, se mandará un mensaje (con cote.js o con RabbitMQ) con la ruta del filesystem a la imagen.
- crear un worker que esté suscrito al mensaje/cola y vaya haciendo thumbnails de cada imagen a un tamaño de 100x100 píxeles.

Ejemplo de la llamada de prueba en Postman (fijaros que el tipo del campo ‘foto’ en la petición es ‘File’):



## [OPCIONAL] BONUS TRACK! - Crear un módulo público

Si hemos llegado hasta aquí con ganas... publiquemos un trabajo!

Se propone hacer alguna utilidad de nuestra invención que pueda resultar útil para nosotros mismos o para otros.

Puede ser cualquier tipo de utilidad, a continuación unas sugerencias para evocar vuestra creatividad:

- Dhrase, Frase del día
- RabinstonWabbit, transport de [Winston](#) a [RabbitMQ](#)
- Imandom, que devuelve una URL a una imagen aleatoria online
- Frine, Lector de la primera línea de un fichero de texto
- HipoTcalculator, calculador de hipoteca

Colocar la URL de npm.js del módulo que hayáis publicado al inicio del README.md del repo que entreguéis.