

Testeamos nuestra API Rest

Reporte de resultados

Testeo de la app por medio de AXIOS

Para probar este test se debe ejecutar el script **npm run testaxios**.

Este test ejecuta los siguientes pasos llamando a la API usando la librería AXIOS. Por medio de **console.log()** voy mostrando el resultado de cada paso.

- Crea 4 productos.
- Muestra el listado de productos.
- Modifica el precio del primer producto.
- Elimina el ultimo producto.
- Vuelve a mostrar el listado de productos.
- Limpia la base de datos.

Resultados que muestra la consola

-- INCIO TESTEO AXIOS -----

- Nuevo producto creado: "Bananas"
- Nuevo producto creado: "Peras"
- Nuevo producto creado: "Uvas"
- Nuevo producto creado: "Cerezas"

LISTADO DE PRODUCTOS:

```
[
  {
    "_id": "612ce3e48916204958ddb460",
    "title": "Bananas",
    "price": 90,
    "thumbnail": "10",
    "__v": 0
  },
  {
    "_id": "612ce3e58916204958ddb462",
    "title": "Peras",
    "price": 70,
    "thumbnail": "12",
    "__v": 0
  },
  {
    "_id": "612ce3e58916204958ddb464",
    "title": "Uvas",
    "price": 110,
    "thumbnail": "09",
  }
```

```
"__v": 0
},
{
  "_id": "612ce3e58916204958ddb466",
  "title": "Cerezas",
  "price": 140,
  "thumbnail": "03",
  "__v": 0
}
]
● Precio del primer producto modificado.
✗ Ultimo producto eliminado.
```

LISTADO DE PRODUCTOS ACTUALIZADO:

```
[
  {
    "_id": "612ce3e48916204958ddb460",
    "title": "Bananas",
    "price": 99,
    "thumbnail": "10",
    "__v": 0
  },
  {
    "_id": "612ce3e58916204958ddb462",
    "title": "Peras",
    "price": 70,
    "thumbnail": "12",
    "__v": 0
  },
  {
    "_id": "612ce3e58916204958ddb464",
    "title": "Uvas",
    "price": 110,
    "thumbnail": "09",
    "__v": 0
  }
]
● Base de datos borrada.
-- FIN TESTEO AXIOS -----
```

Testeo de la app por medio de AXIOS

Para probar este test se debe ejecutar el script **npm run test**.

Usando **Mocha**, **Chai** y **Supertest** ejecuto 5 tests.

Los 5 tests pasaron correctamente.

Test 1 Reinicio de base de datos

Comencé borrando la base para empezar de 0.

Test 2 Creación de un nuevo producto (POST)

Luego agregue un producto con estos datos.

```
{"title": "Bananas", "price": 90, "thumbnail": "10"}
```

Luego testee que el producto tenga los siguientes keys: 'title', 'price', 'thumbnail'.

Luego testee que title = 'Bananas', price = 90, Thumbnail = '10'

Test 3 Chequeo que haya 1 producto en la base (GET)

Cargo los productos.

Chequeo que el status sea 200.

Chequeo que el listado tenga 1 producto.

Test 4 Modificación en el precio del primer producto (PUT)

Levanto los productos y selecciono el primero, de ese tomo el ID.

Con ese ID selecciono el producto y cambio el precio de 90 a 105.

Chequeo que el objeto devuelto desde el PUT muestre que 1 item fue modificado
(response.body.updatedProd.nModified = 1)

Por ultimo chequeo que el nuevo precio del ítem sea 105.

Test 5 Eliminación de un item (DEL)

Levanto los productos y selecciono el primero, de ese tomo el ID.

Con ese ID borre el ítem con una llamada a delete.

Chequeo que el status de ese llamado sea 202 (Accepted).

Por ultimo llamo de nuevo al listado de productos y si el status es 404 es que el único producto que habia se borro correctamente.

Resultados que muestra la consola

Testeo de API restfull - Testeo de la API de productos.

- ✓ Reinicio de base de datos **(1341ms)**
- ✓ Creación de un nuevo producto (POST) **(217ms)**
- ✓ Chequeo que haya 1 producto en la base (GET) **(190ms)**
- ✓ Modificación en el precio del primer producto (PUT) **(756ms)**
- ✓ Eliminación de un item (DEL) **(556ms)**