

# ACTIVIDAD 11.B3. CRUD DE MOTOS EN NODEJS

## 1. Un post de un artículo

POST ▼ http://localhost:3000/moto/

Params Auth **Headers (10)** Body ● Pre-req. Tests Settings

Headers 8 hidden

	KEY	VALUE	DESCRIP
<input checked="" type="checkbox"/>	Content-type	application/json	

NODEJS / MOTOS / POST Moto Save ... ✎ 💬

POST ▼ http://localhost:3000/moto/ Send

Params Auth Headers (10) Body ● Pre-req. Tests Settings Cookie

raw ▼ JSON ▼ Beauti

```
1 ...
2 ... "marca": "Honda",
3 ... "modelo": "DAW",
4 ... "cilindrada": "2500",
5 ... "precio": "15000",
6 ... "imagen": "imagenHonda.jpg"
7 ...
```

Body Cookies Headers (7) Test Results 🌐 200 OK 7 ms 577 B Save Response

Pretty Raw Preview Visualize JSON ▼ 🔍

```
1
2  "ok": true,
3  "msg": "post API motos: {\n  marca: 'Honda',\n  modelo: 'DAW',\n  cilindrada: '2500',\n  precio: '15000',\n  imagen: 'imagenHonda.jpg',\n  _id: new ObjectId\n    (\n\"62094c02e4f7dabb56da7043\"\n)\n}",
4  "miMoto": {
5    "marca": "Honda",
6    "modelo": "DAW",
7    "cilindrada": "2500",
8    "precio": "15000",
9    "imagen": "imagenHonda.jpg",
10   "_id": "62094c02e4f7dabb56da7043"
11  }
12
```

## 2. Un put ( actualización )

The screenshot displays a REST client interface with a PUT request to the URL `http://localhost:3000/moto/62094c02e4f7dabb56da7043`. The request body is a JSON object with the following fields: `marca` (Honda), `modelo` (PUT), `cilindrada` (2500), `precio` (12995), and `imagen` (imagenHonda.jpg). The `precio` field is highlighted with a red box. The response is a 200 OK status with a JSON body containing `ok: true`, a message, and the updated object, where the `precio` field is also highlighted with a red box.

```
PUT http://localhost:3000/moto/62094c02e4f7dabb56da7043

Params Auth Headers (10) Body ● Pre-req. Tests Settings
raw JSON


1  ...
2  ... "marca": "Honda",
3  ... "modelo": "PUT",
4  ... "cilindrada": "2500",
5  ... "precio": "12995",
6  ... "imagen": "imagenHonda.jpg"
7

Body Cookies Headers (7) Test Results 200 OK
Pretty Raw Preview Visualize JSON

1
2  "ok": true,
3  "msg": "put API Motos: 62094c02e4f7dabb56da7043",
4  "body": {
5    "marca": "Honda",
6    "modelo": "PUT",
7    "cilindrada": "2500",
8    "precio": "12995",
9    "imagen": "imagenHonda.jpg"
10  }
11
```

### 3. Un get de un artículo

**NODEJS / MOTOS / GET Motos**



**GET**  http://localhost:3000/moto/62094b13e4f7dabb56da7038

Params Auth Headers (8) Body ● Pre-req. Tests Settings

**Query Params**

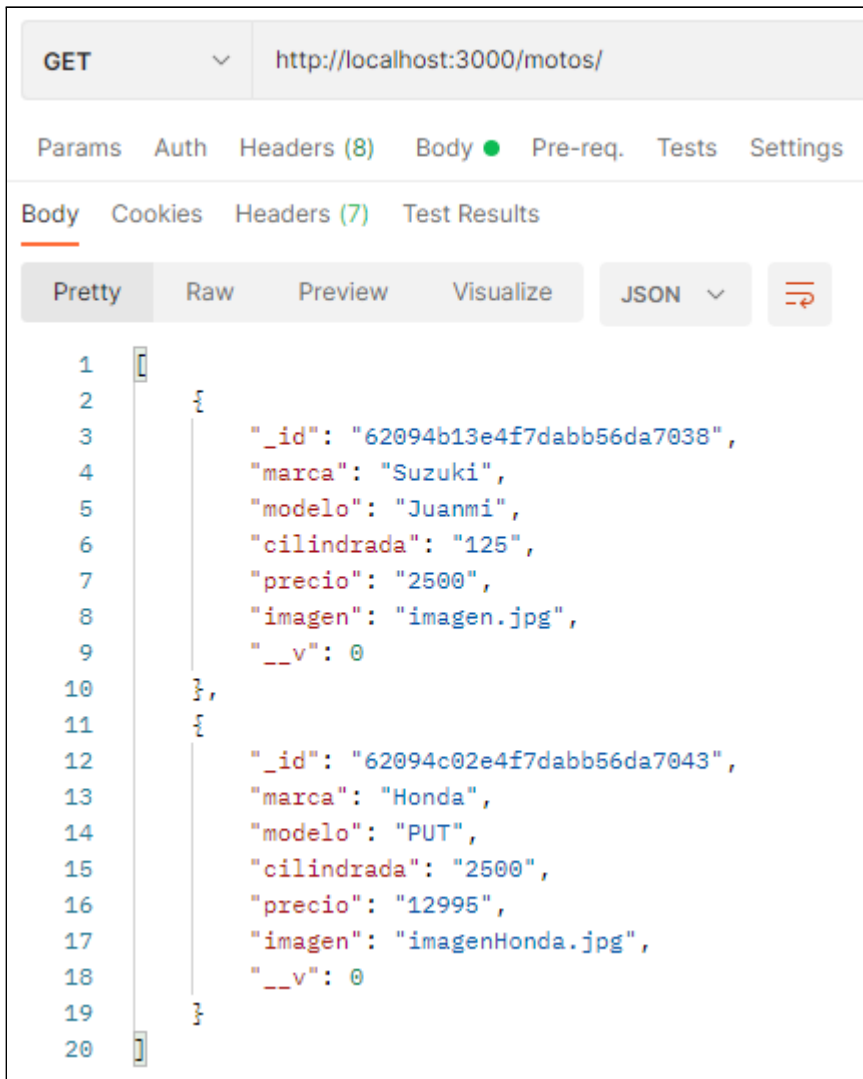
	KEY	VALUE
--	-----	-------

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON  

```
1  {
2    "_id": "62094b13e4f7dabb56da7038",
3    "marca": "Suzuki",
4    "modelo": "Juanmi",
5    "cilindrada": "125",
6    "precio": "2500",
7    "imagen": "imagen.jpg",
8    "--v": 0
9  }
```

## 4. Un get de todos los artículos



The screenshot shows a REST client interface with a GET request to `http://localhost:3000/motos/`. The response is displayed in the 'Body' tab, formatted as JSON. The response contains an array of two motorcycle objects. The first object is for a Suzuki Juanmi with a price of 2500. The second object is for a Honda PUT with a price of 12995. Both objects include an ID, brand, model, cylinder count, price, image path, and a version number.

```
1  [
2    {
3      "_id": "62094b13e4f7dabb56da7038",
4      "marca": "Suzuki",
5      "modelo": "Juanmi",
6      "cilindrada": "125",
7      "precio": "2500",
8      "imagen": "imagen.jpg",
9      "__v": 0
10   },
11   {
12     "_id": "62094c02e4f7dabb56da7043",
13     "marca": "Honda",
14     "modelo": "PUT",
15     "cilindrada": "2500",
16     "precio": "12995",
17     "imagen": "imagenHonda.jpg",
18     "__v": 0
19   }
20 ]
```

## 5. Un delete


**NODEJS / MOTOS / DELETE Moto**

**DELETE** ▾

http://localhost:3000/moto/62094b13e4f7dabb56da7038

ParamsAuthHeaders (8)Body ●Pre-req. TestsSettings

BodyCookiesHeaders (7)Test Results

PrettyRawPreviewVisualizeJSON ▾

```
1 {
2   "ok": true,
3   "msg": "delete API Moto: 62094b13e4f7dabb56da7038"
4 }
```