

ESCUELA SUPERIOR DE INGENIERÍA

Programación en Internet

Grado en Ingeniería Informática

Servicios Web & jQuery

Autores:

José Manuel Periñán Freire

Iñaki Urrutia Sanchez

Supervisores:

David Corral Plaza

Cádiz, 26 de 11 de 2018

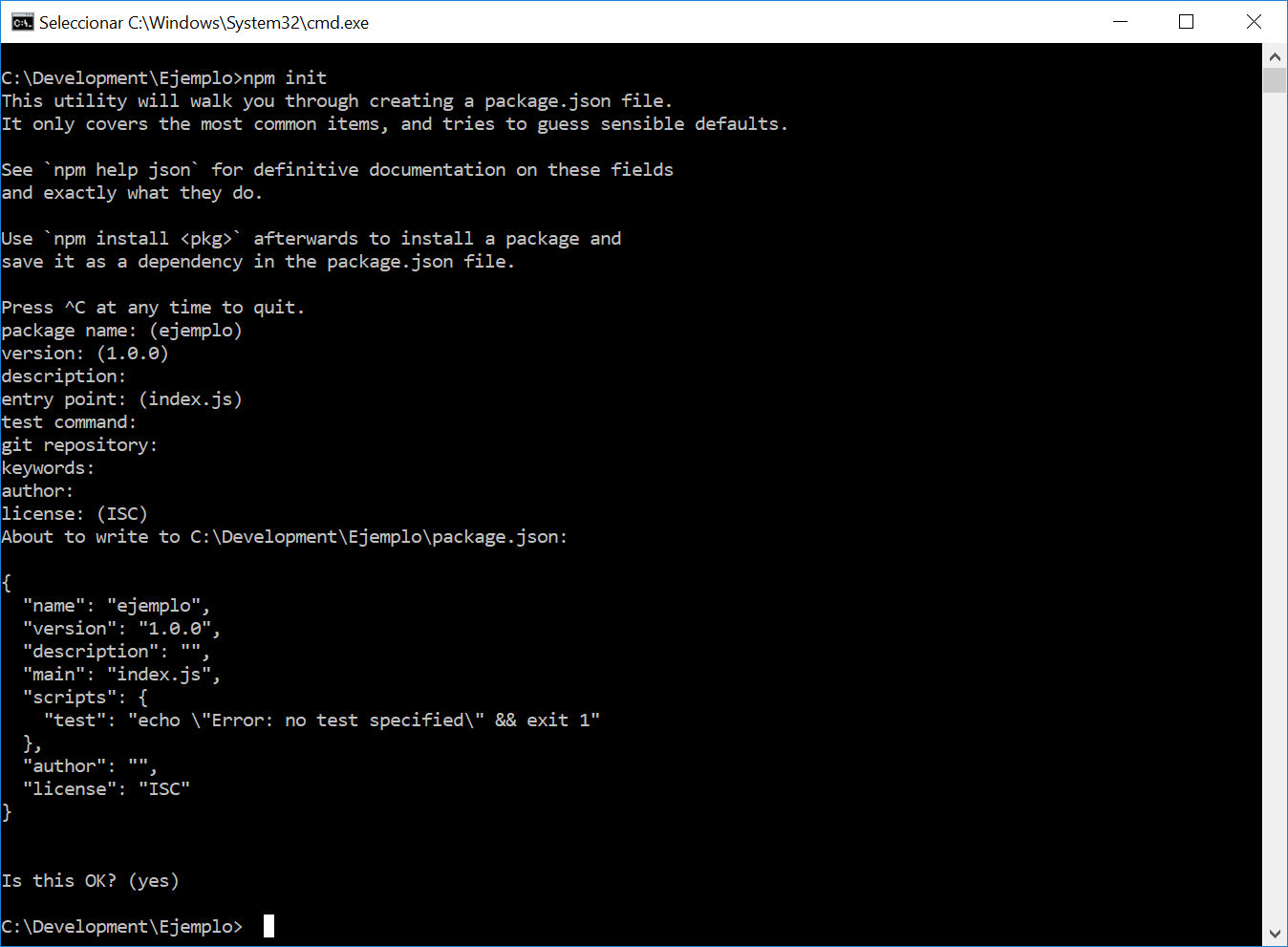
Introducción

Primero vamos a crear una carpeta que contendrá todo el proyecto en nuestro caso:

*C:/Development/Ejemplo*

Dentro de dicha carpeta abrimos la terminal y aplicamos el siguiente comando:

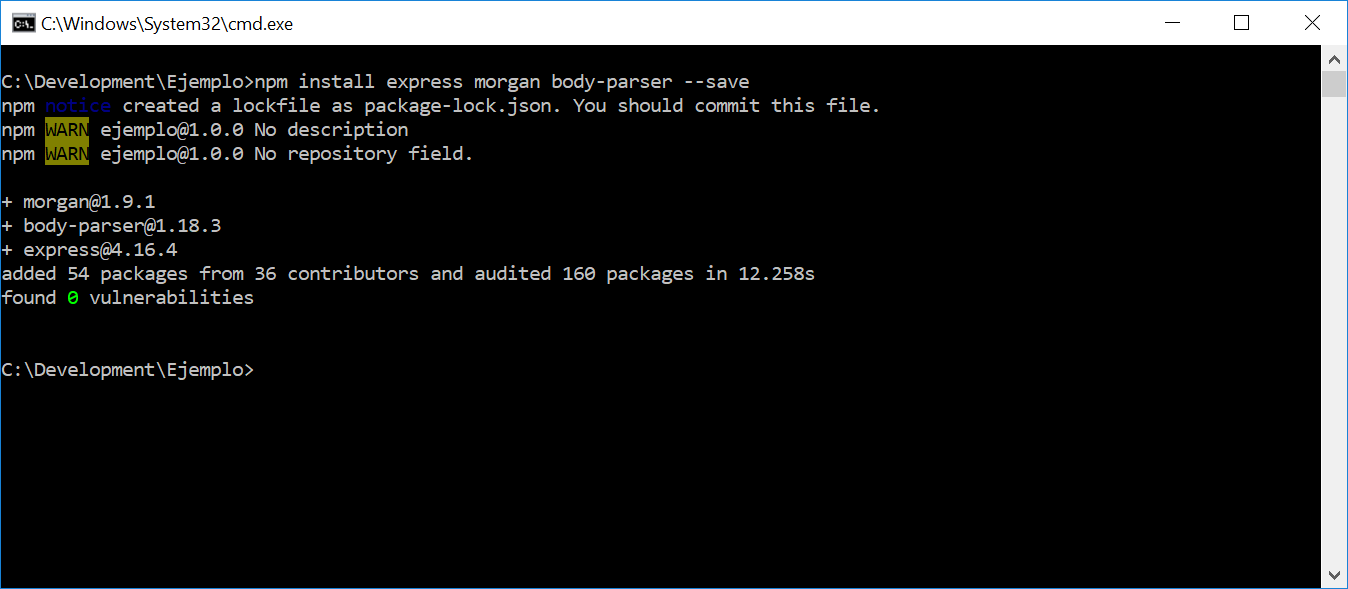
*npm init*



Una vez realizado esto se nos creará en nuestra carpeta “Ejemplo” el archivo *packaje.json*

El siguiente paso será instalar las dependencias con el siguiente comando:

*npm install express morgan body-parser –save*



Ahora creamos el archivo index.js o según el nombre que hayamos indicado en el primer paso en la opción de *entry point*, y lo colocamos en la raíz del proyecto.

Dentro de dicho archivo colocamos las siguientes instrucciones:

Constantes:

const express = require('express');

const app = express();

const logger = require('morgan');

const http = require('http');

const path = require('path');

const PORT = process.env.PORT || 8080;

const bodyParser = require('body-parser');

const baseAPI = '/api/v1';

Configuraciones:

app.use(bodyParser.json());

app.use(logger('dev'));

app.use(bodyParser.urlencoded({

extended: true

}));

Método GET:

app.get('/', function (req, res) {

res.send('Hello World!');

});

Instanciar el servidor y ponerlo a la escucha por el puerto 8080 (por defecto):

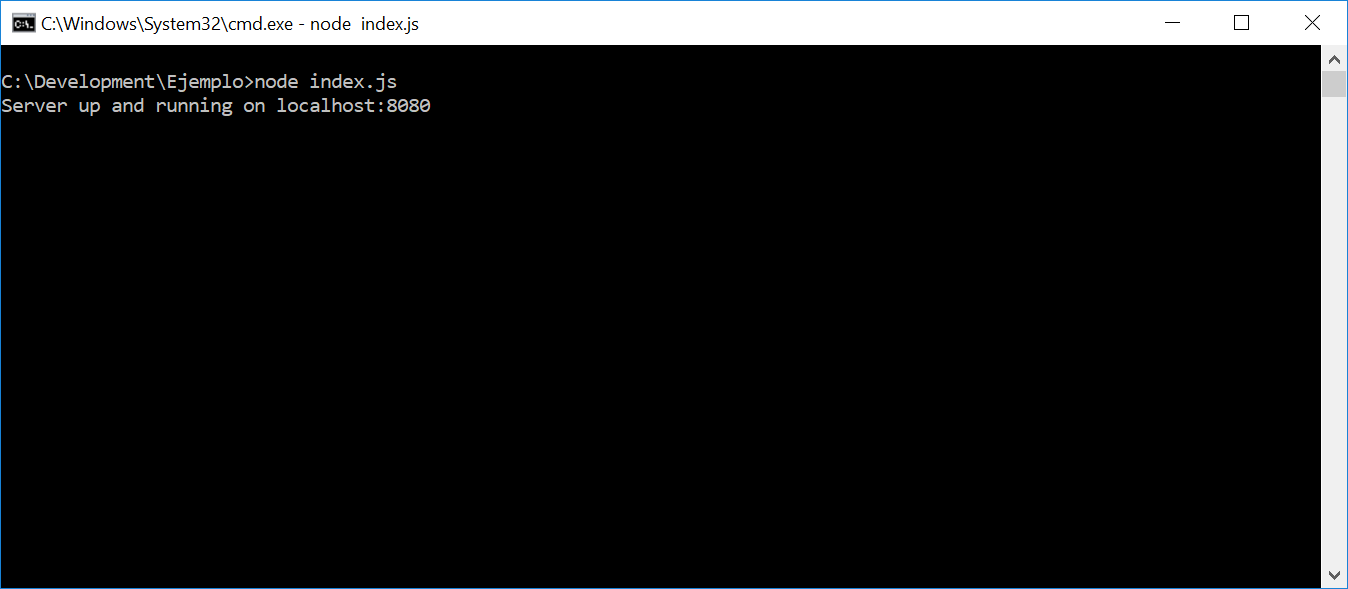
const server = http.createServer(app);

server.listen(PORT, function () {

console.log('Server up and running on localhost:' + PORT);

});

Para inicar la API rest nos colocamos en el directorio raíz del proyecto abrimos la consola y ejecutamos el siguiente comando:



Si todo se ha realizado correctamente el servidor quedaría lanzado y ahora crearemos la API de móviles para usarla como ejemplo para futuras APIs.

Para evitar sobrecargar el index.js vamos a optar por mostrar directamente como realizarlo usando *routes* que nos permite tener organizados los distintos recursos en nuestra API.

El primer paso será crear la carpeta routes en la raíz del proyecto:

*C:/Development/Ejemplo/routes*

Dentro de dicha carpeta creamos el archivo javascript según el recurso que queramos, en este caso, *mobiles.js*

*C:/Development/Ejemplo/routes/mobiles.js*

Dentro de dicho archivo añadimos las siguientes instrucciones:

'use strict';

const express = require('express');

const router = express.Router();

Ahora mismo no estamos usando persistencia así que trabajaremos sobre el siguiente JSON de mobiles, a continuación encontraríamos definidos los métodos GET, POST, DELETE, GET de un móvil que coincida con el criterio y PUT.

- GET devuelve todos los móviles del JSON.

- POST añade un móvil al JSON.

- DELETE borra todos los móviles del JSON.

- PUT modificar un móvil en concreto según la marca que indiquemos.

let mobiles = [

{

"marca":"Xiaomi",

"modelo":"Xiaomi Mi A2 4GB+64GB",

"precio":234

},

{

"marca": "Samsung",

"modelo": "Samsung s9",

"precio": 602

}

];

router.get('/', function (req, res) {

res.send(mobiles);

});

router.post('/', function (req, res) {

mobiles.push(req.body);

res.status(201).send("Mobiles created");

});

router.delete('/', function (req, res) {

mobiles = [];

res.status(200).send("Mobiles deleted");

});

router.get('/:marca', function (req, res) {

let marca = req.params.marca;

res.send(mobiles.filter(m => m.marca === marca));

});

router.put('/:marca', function (req, res) {

let marca = req.params.marca;

mobiles = mobiles.filter(m => m.marca !== marca);

mobiles.push(req.body);

res.status(200).send("Mobiles updated!");

});

Para exportar el router:

module.exports = router;

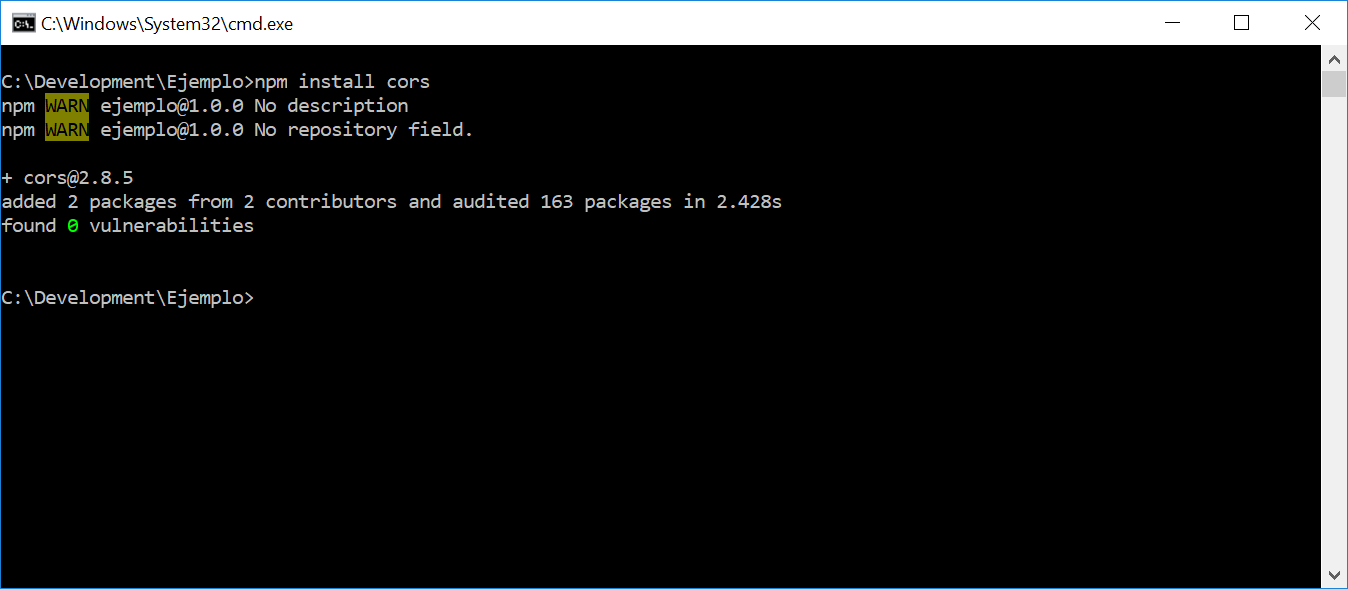
Añadimos al index.js las siguientes líneas para importar el router que acabamos de crear:

const mobiles = require('./routes/mobiles');

app.use('/mobiles', mobiles);

Instalamos después de esto las dependencias de CORS para que no deniegue las solicitudes que provengan de otro servidor:

*npm install cors –save*



Y añadimos las 2 siguientes líneas a nuestro index.js:

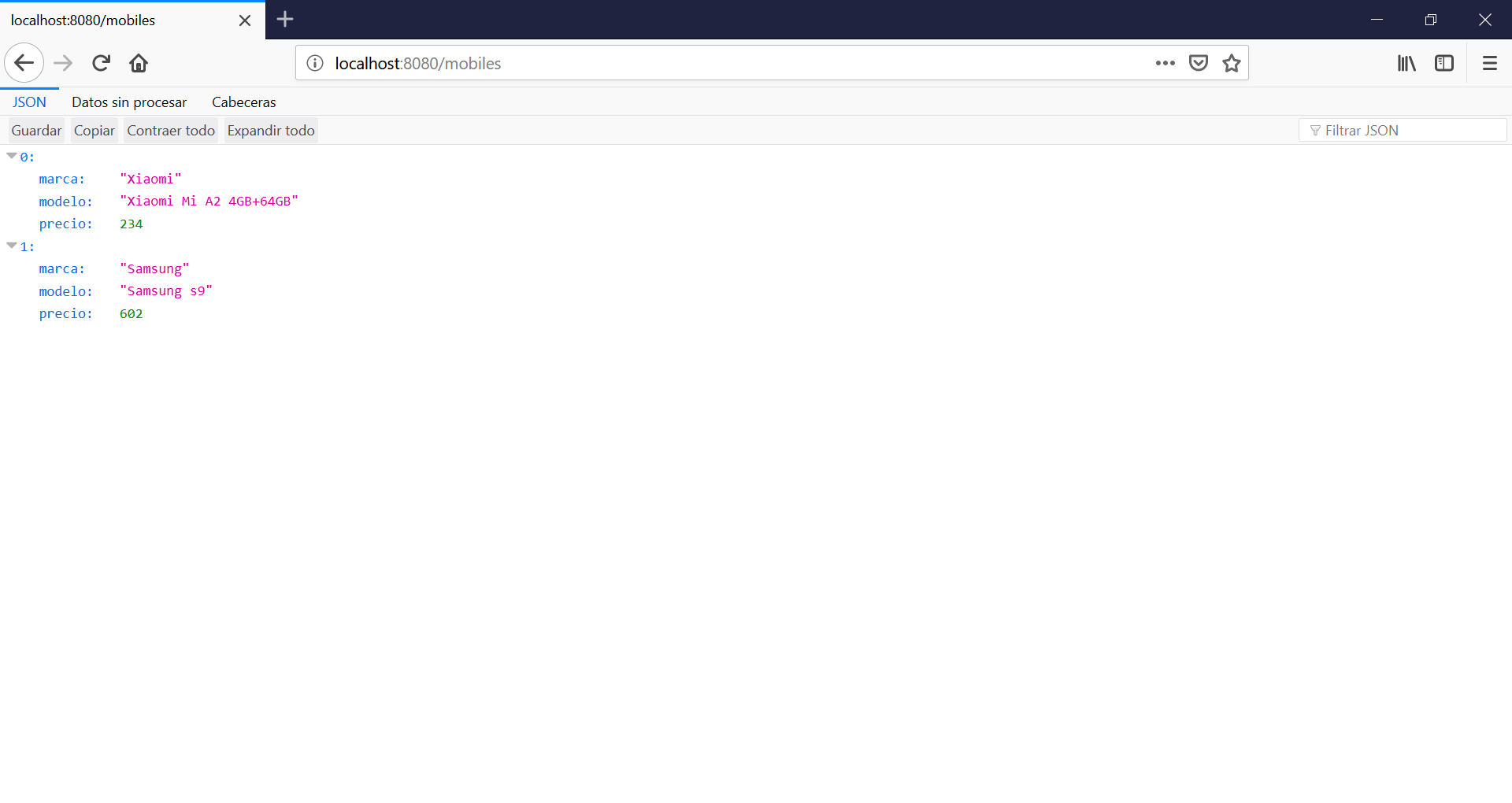
const cors = require('cors');

app.use(cors());

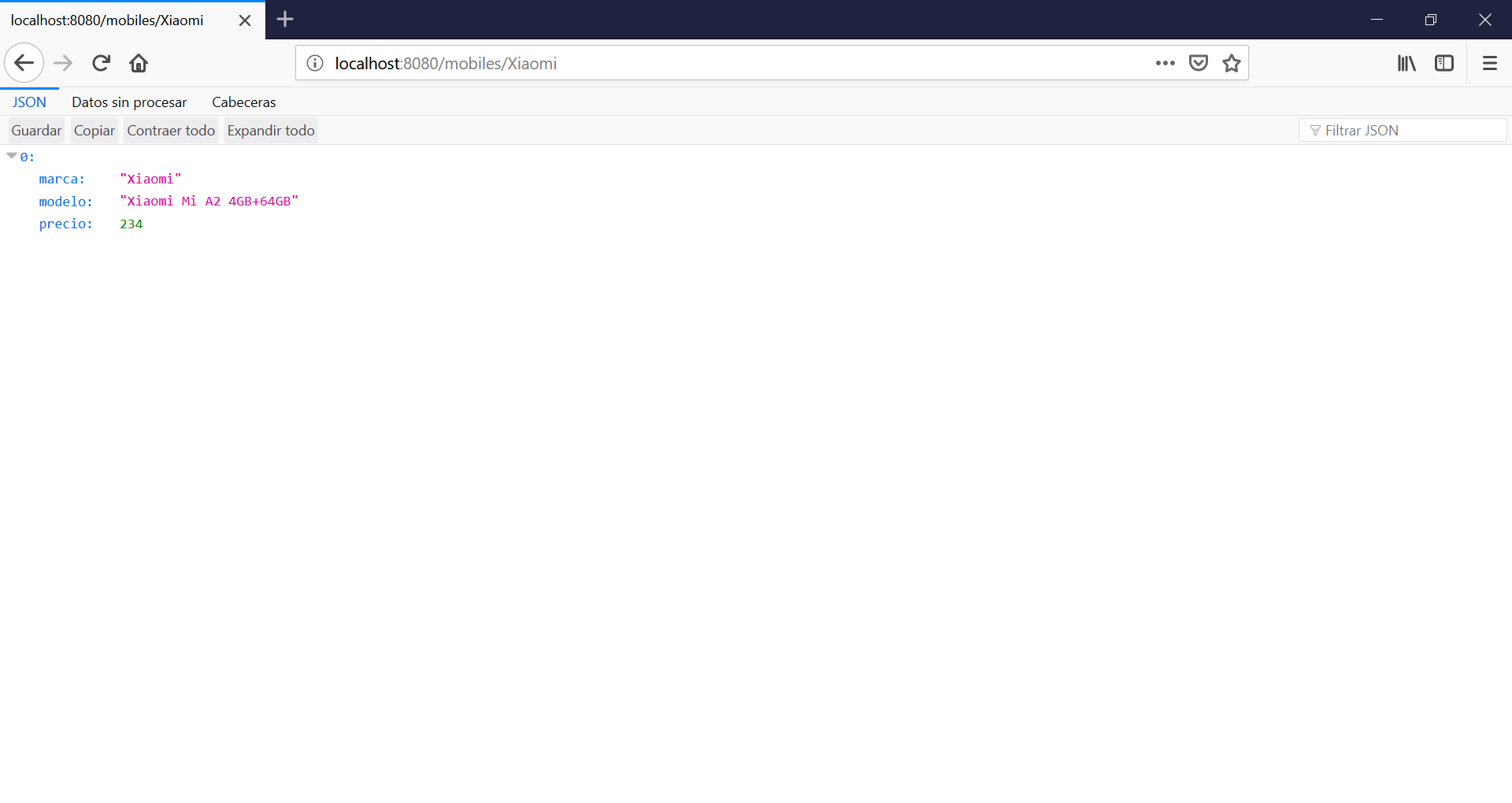
Una vez realizado todo esto empezaremos a probar las distintas peticiones a nuestro servidor:

* GET

Vamos a nuestro navegador y colocamos la siguiente URL: [*http://localhost:8080/mobiles*](http://localhost:8080/mobiles)

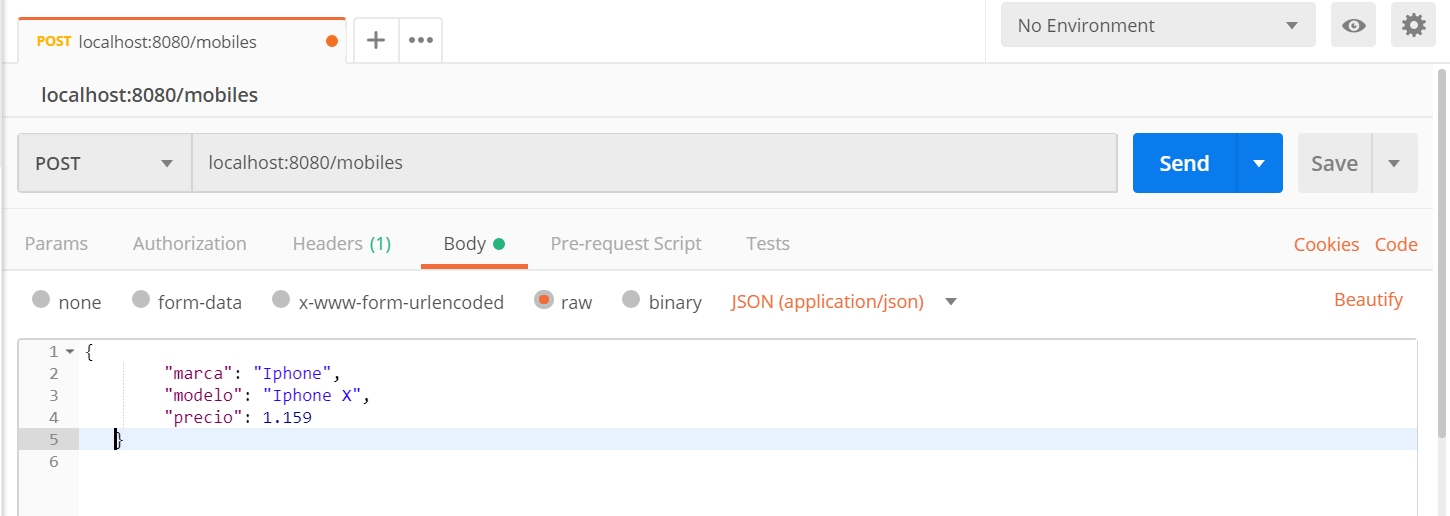


* GET con una marca en concreto:

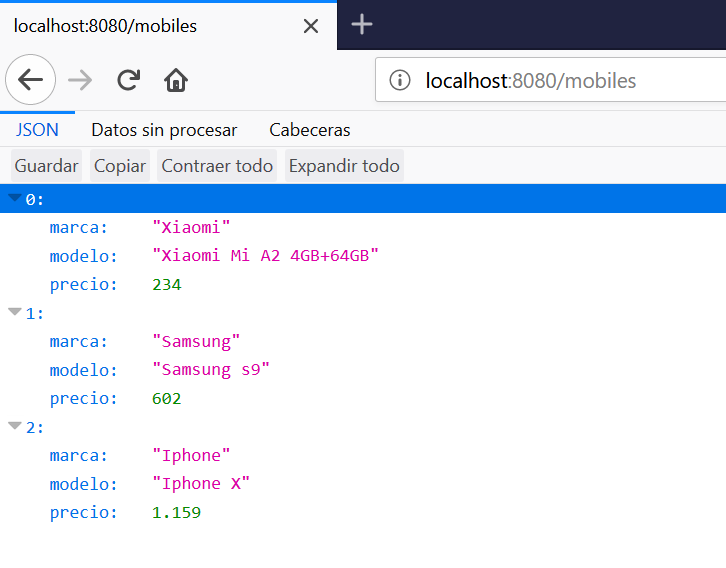


Para las demás operaciones utilizaremos la herramienta Postman:

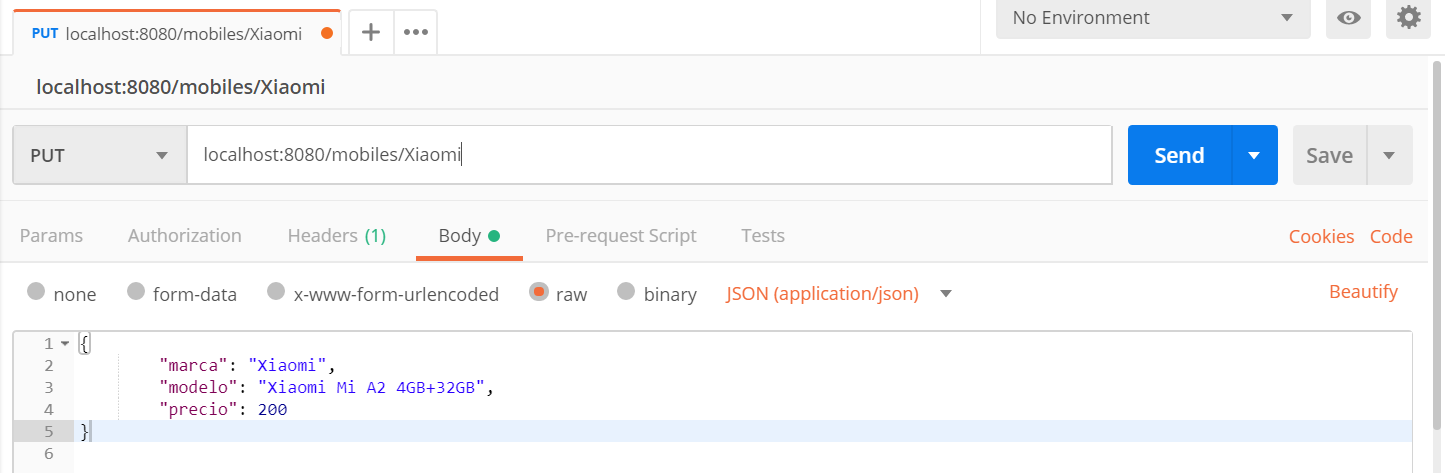
* POST

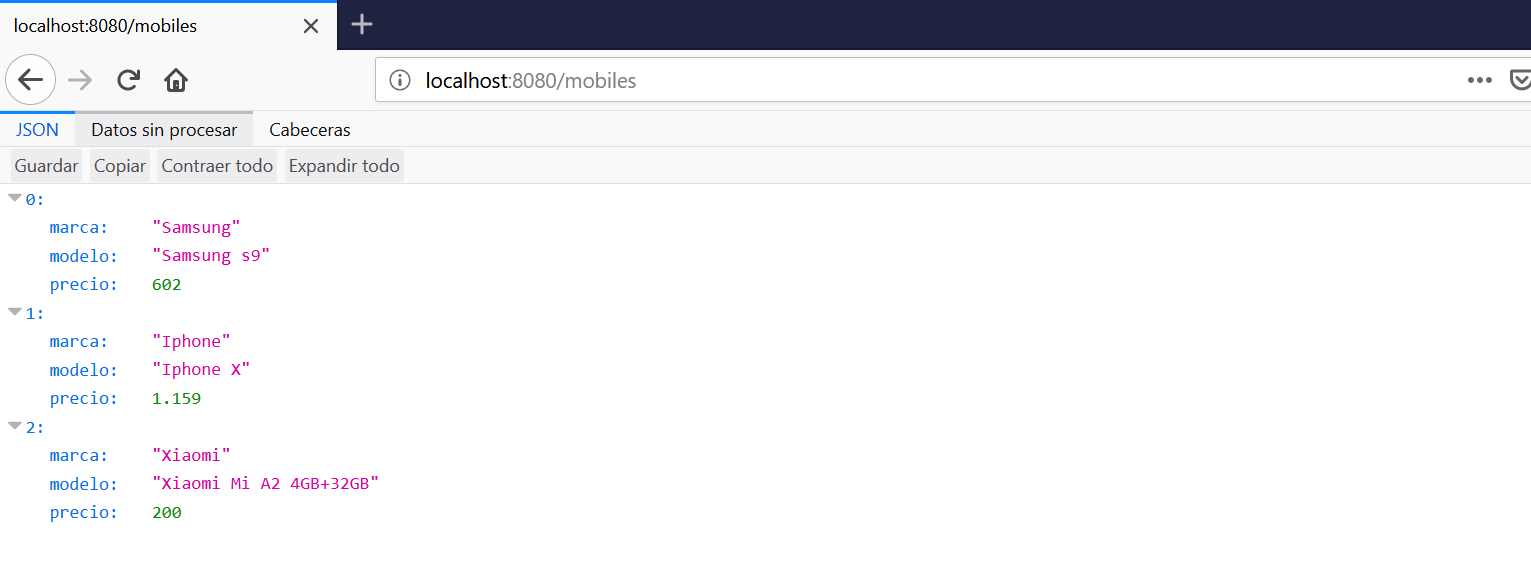


Y quedaría añadido:



* PUT





* DELETE

