

# Creació d'una API web amb Node.js

Aquesta pràctica consistirà en crear una petita API REST amb Node.js. S'avaluarà mitjançant un lliurament de la pràctica molt senzill. Servirà per puntuar la UF3 i la UF4.

## Examen de la pràctica: dia 18/5 (5 punts preguntes + 5 punts programar)

Software que heu de tenir instal·lat:

- Visual Studio Code (o l'IDE que preferiu, però jo us recomano aquest)
- Git
- Node.js (des d'[aquest link](#))
- Postman o Insomnia per provar la API

**IDE a utilitzar:** Visual Studio Code amb els plugins de Node.js i Express

Tutorial i documentació que heu de seguir:

<https://docs.microsoft.com/ca-es/learn/modules/build-web-api-nodejs-express/>

Es tracta d'anar seguint aquest tutorial mitjançant les instruccions que hi ha a continuació, però procurant comprendre el codi i què fa cada fitxer, així com les comandes de node.js que emprarem via terminal. Ja que això serà important de cara a l'examen: comprendre què fa cada cosa i no simplement copiar el codi del tutorial.

### PRIMERA PART: Instruccions

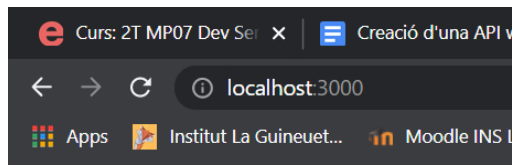
- Llegir i entendre les pàgines 1 i 2 del tutorial, que contenen teoria
- Instalar node.js

[Instalado](#)

- Crear un nou repo a github per aquest projecte

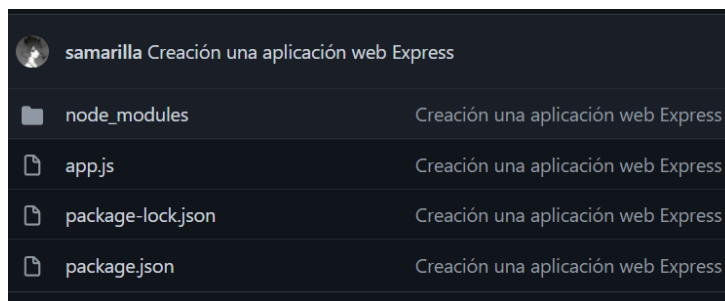
[Hecho](#)

- Seguir les instruccions de la pàgina 3 del tutorial: "Creación de una aplicación web básica con Express"



Hello World!

- Després del punt 4 de les instruccions de la pàgina 3, podeu fer el primer commit i push.
- En aquest moment, haurieu de tenir 3 fitxers: ***app.js***, ***package.json*** i ***package-lock.json***



- Investigueu què fan els fitxers ***package.json*** i ***package-lock.json***.

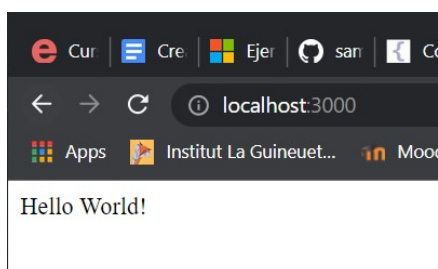
Es un archivo generado automaticamente cuando se instalan paquetes o dependencias en el proyecto, la finalidad es mantener un historial de paquetes instalados y optimizar la forma en que se generan las dependencias del proyecto.

Y package-lock.json se encarga de generar un snapshot del arbol de depenencias que se incluye en el proyecto, con esto se puede resolver problemas especificos que package.json ha dejado sin resolver.

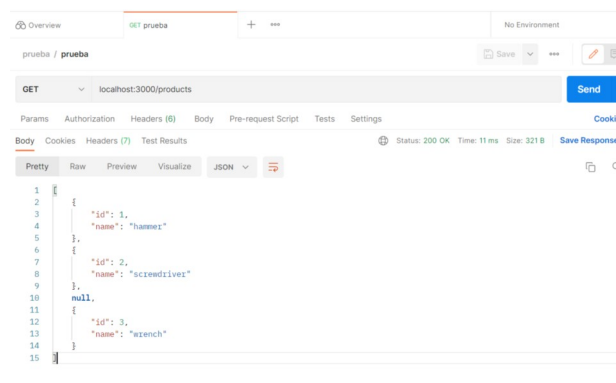
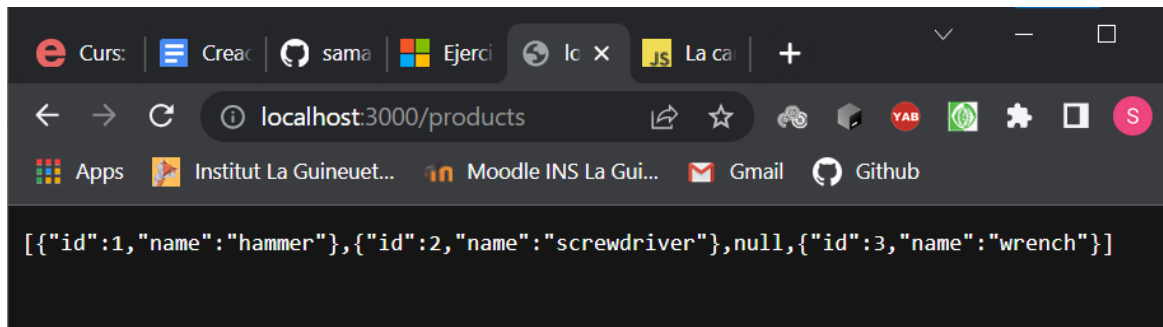
- Investigueu què és la carpeta ***node\_modules***.

En la carpeta node\_modules se almacena cualquier paquete instalado del proyecto, en una carpeta con el nombre del paquete junto a todos los ficheros necesarios y dependencias en el propio node\_modules.

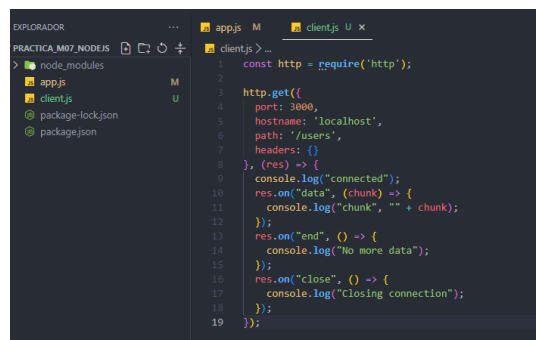
- Executar i veure que funciona. Acte seguit podem fer commit i push



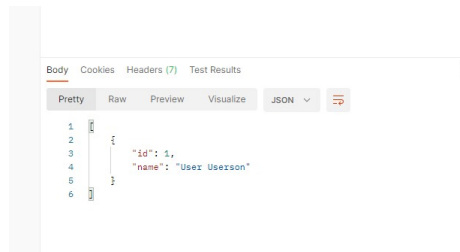
- Seguir les instruccions de la pàgina 3 del tutorial: "Creación de una aplicación web que devuelve datos JSON"
- Executar i veure que funciona tal com es descriu al tutorial. Provar l'endpoint productes amb el navegador i també amb postman.



- Fer commit i push.
- Llegir la pàgina 4. D'aquí és important comprendre què són els paràmetres **req**, **res** i **next**.
- Aneu a la pàgina 5 però no us baixeu el repo que diu. Simplement modifiqueu el vostre **app.js** perquè sigui com el d'aquesta pàgina. També haureu de crear l'arxiu **client.js** i provar-lo tal com diu.



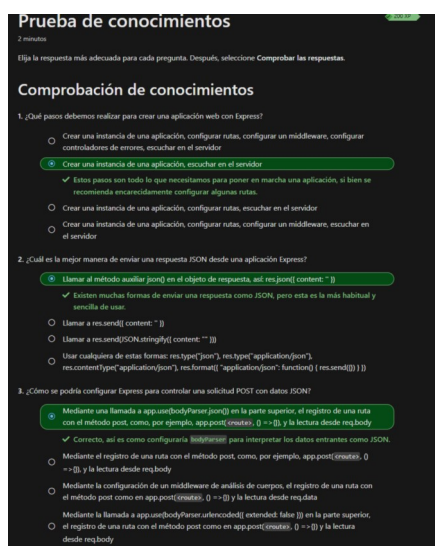
- Un cop hàgiu provat tot el que diu en aquesta pàgina, obriu el postman (o insomnia) i feu una crida al get de **/users**. Investigueu com fer-ho perquè funcioni i retorni el llistat d'usuaris.



- Afegir el middleware **isAuthorized** de la pàgina 5

```
PS F:\Informatica\DAW-2T\M07\clonado\casa\Practica_m07_nodejs> node .\client.js
connected
chunk Not permitted
No more data
Closing connection
PS F:\Informatica\DAW-2T\M07\clonado\casa\Practica_m07_nodejs> node .\client.js
connected
chunk [{"id":1,"name":"User Userson"}]
No more data
Closing connection
PS F:\Informatica\DAW-2T\M07\clonado\casa\Practica_m07_nodejs>
```

- Finalment, contesteu el qüestionari de la pàgina 6



- Un cop hàgiu fet tot això, podeu fer un últim push. Recordeu penjar al moodle el link al vostre repo.

Listo

## SEGONA PART: Instruccions

Volem aconseguir dues coses:

- Que la informació de la resposta de /users surti d'una BD SQL que podeu instanciar amb phpmyadmin (necessitareu el port de la BD). Utilitzarem la llibreria **Sequelize** per express. La podeu instal·lar com a paquet del projecte.

The screenshot shows a REST client interface with the following details:

- URL:** localhost:3000/users
- Method:** GET
- Headers:** Authorization: secretpassword
- Status:** 200 OK
- Time:** 48 ms
- Size:** 878 B
- Response Body (JSON):**

```
{
  "password": "1234",
  "email": "samarilla@ilg.cat",
  "createdAt": null,
  "updatedAt": null
},
{
  "id": 17,
  "username": "cesar",
  "password": "1234",
  "email": "cesar@ilg.cat",
  "createdAt": null,
  "updatedAt": null
}
```