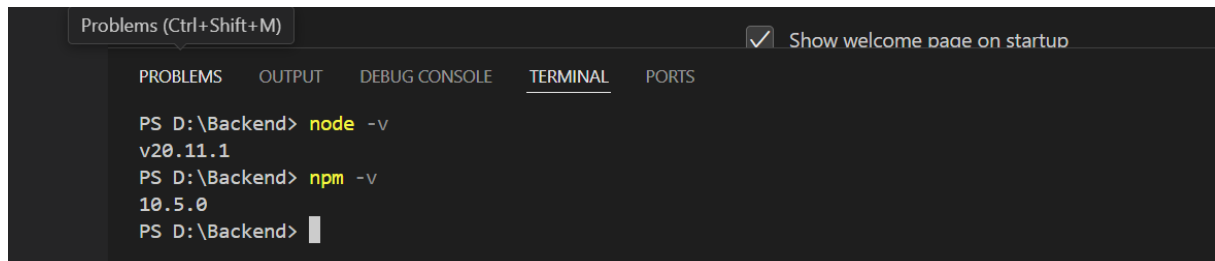


Activité : Créer une API REST de base avec NodeJS et ExpressJS

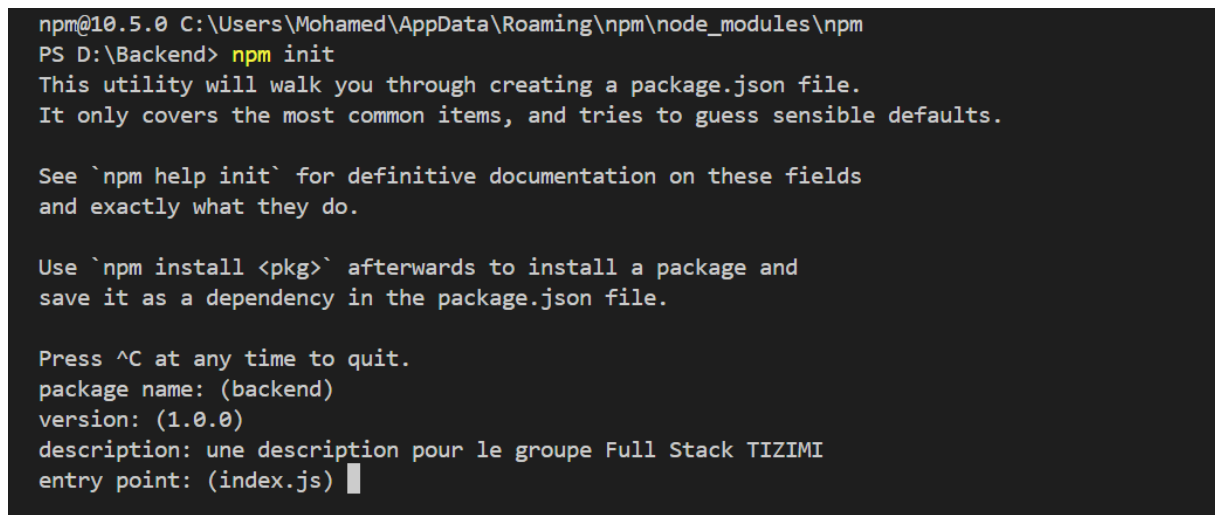
1. Verifier l'installation de node .



The screenshot shows the VS Code interface with the 'TERMINAL' tab active. The terminal output shows the following commands and results:

```
PS D:\Backend> node -v
v20.11.1
PS D:\Backend> npm -v
10.5.0
PS D:\Backend> 
```

2. Créer le fichier package.json



The screenshot shows the VS Code terminal with the output of the 'npm init' command. The output includes instructions on how to use the package.json file and prompts for package details.

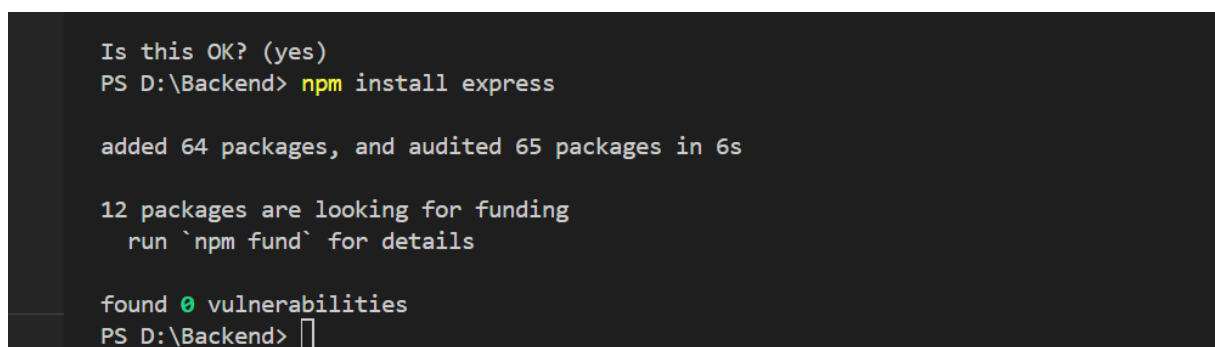
```
npm@10.5.0 C:\Users\Mohamed\AppData\Roaming\npm\node_modules\npm
PS D:\Backend> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (backend)
version: (1.0.0)
description: une description pour le groupe Full Stack TIZIMI
entry point: (index.js) 
```

3. Installer expressJS



The screenshot shows the VS Code terminal with the output of the 'npm install express' command. The output indicates that 64 packages were added and 65 packages were audited in 6 seconds.

```
Is this OK? (yes)
PS D:\Backend> npm install express

added 64 packages, and audited 65 packages in 6s



12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS D:\Backend> 
```

4. Vérifier le fichier package.json

```
{ } package.json > ...
6   "scripts": {
8   },
9   "author": "",
10  "license": "ISC",
11  "dependencies": {
12    "express": "^4.19.2"
13  }
14 }
```

5. Créer le fichier index.js

```
JS index.js >  monApp.listen() callback
1  const express = require('express')
2  const monApp = express()
3  monApp.listen(85, () => {
4     console.log('SALAM API via ExpressJS')
5  }
6  )
```

6. Lancer le serveur

```
run npm fund for details
found 0 vulnerabilities
PS D:\Backend> node index.js
SALAM API via ExpressJS
```

← → ↻ ⓘ localhost:85

Cannot GET /

`{}` equipes.json > ...

```
1  [  
2      {  
3          "id":1,  
4          "name":"CODM",  
5          "country":"Morocco"  
6      },  
7      {  
8          "id":2,  
9          "name":"WAC",  
10         "country":"Morocco"  
11     },  
12     {  
13         "id":3,  
14         "name":"RCA",  
15         "country":"Morocco"  
16     },  
17     {  
18         "id":4,  
19         "name":"FAR",  
20         "country":"Morocco"  
21     },  
22     {  
23         "id":5,  
24         "name":"Real Madrid",  
25         "country":"Spain"
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

PS D:\Backend> `node` index.js

SALAM API via ExpressJS

```
8
9  monApp.get('/equipes',(rer,res) => {
10    |   res.send("Teams :")
11  })
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Backend> node index.js
SALAM API via ExpressJS
```

```
const teams = require('./equipes.json')
monApp.listen(85, () => {
```

GET localhost:85/equipes

localhost:85/equipes

GET localhost:85/equipes

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Retain authorization header when a redirect happens to a different hostname.

Remove referer header on redirect OFF

Remove the referer header when a redirect happens.

Body Cookies Headers (7) Test Results 200 OK 4 ms 504 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 1,
4     "name": "CODM",
5     "country": "Morocco"
6   },
7   {
8     "id": 2,
9     "name": "WAC",
10    "country": "Morocco"
11  },
12  {
13    "id": 3,
```

Windows taskbar: F11, VS Code, Edge, Task View, 13:03, 21/04/2024, FRA

GET localhost:85/equipes/1

localhost:85/equipes/1

GET localhost:85/equipes/1

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Retain authorization header when a redirect happens to a different hostname.

Remove referer header on redirect OFF

Remove the referer header when a redirect happens.

Body Cookies Headers (7) Test Results 200 OK 8 ms 277 B Save Response

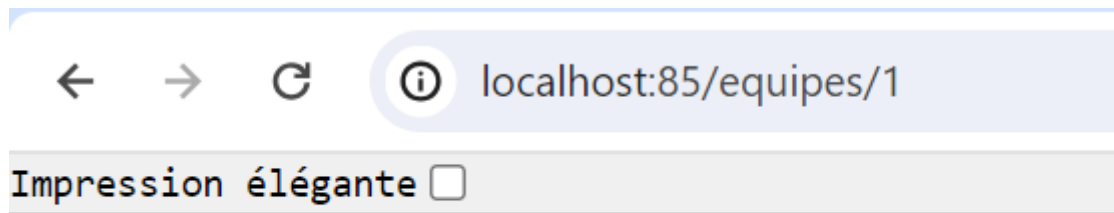
Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "name": "CODM",
4   "country": "Morocco"
5 }
```

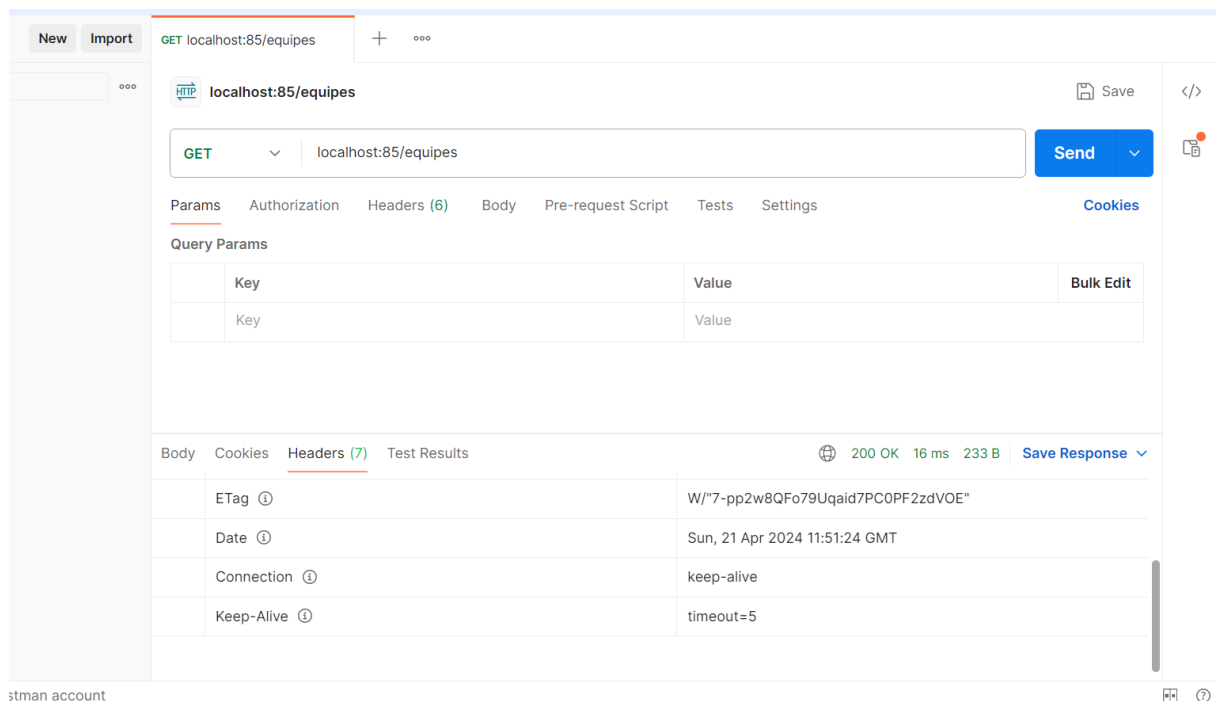
Browser address bar: localhost:85/equipes/

Impression élégante

```
[{"id":1,"name":"CODM","country":"Morocco"}, {"id":2,"name":"WAC","country":"Morocco"}, {"id":3,"name":"RCA","country":"Morocco"}, {"id":4,"name":"FAR","country":"Morocco"}, {"id":5,"name":"Real Madrid","country":"Spain"}, {"id":6,"name":"FC Barcelona","country":"Spain"}]
```



```
{"id":1,"name":"CODM","country":"Morocco"}
```



```
10
11 monApp.get('/equipes',(req,res) => {
12     res.status(200).json(teams)
13 })
14
```

```
15 monApp.get('/equipes/:id',(req,res) => {
16     const id = parseInt(req.params.id)
17     const team = teams.find(team => team.id===id)
18     res.status(200).json(team)
19 })
```

1. ``const express = require('express')``: Cette ligne importe le module Express.js, qui permet de créer des applications web avec Node.js.
2. ``const monApp = express()``: Cette ligne crée une nouvelle instance d'application Express et la stocke dans la variable ``monApp``.
3. ``const teams = require('./equipes.json')``: Ceci importe les données des équipes à partir d'un fichier JSON nommé ``equipes.json`` et les stocke dans la variable ``teams``.
4. ``monApp.listen(85,() => { console.log('SALAM API via ExpressJS') })``: Cette ligne démarre le serveur sur le port 85 et affiche un message dans la console une fois que le serveur est démarré.
5. ``monApp.get('/equipes',(req,res) => { res.status(200).json(teams) })``: Cette ligne définit une route pour l'API qui écoute les requêtes GET sur ``/equipes``. Lorsqu'une telle requête est reçue, elle renvoie les données des équipes stockées dans ``teams`` au format JSON avec un statut HTTP 200 (OK).
6. ``monApp.get('/equipes/:id',(req,res) => { const id = parseInt(req.params.id) const team = teams.find(team => team.id===id) res.status(200).json(team) })``: Cette ligne définit une route pour l'API qui écoute les requêtes GET sur ``/equipes/:id``, où ``:id`` est un paramètre dynamique représentant l'identifiant de l'équipe. Lorsqu'une requête est reçue sur cette route, le code récupère l'identifiant de l'équipe à partir de ``req.params.id``, recherche l'équipe correspondante dans les données ``teams`` à l'aide de ``Array.find()``, puis renvoie les données de cette équipe au format JSON avec un statut HTTP 200.