



UNIVERSITÉ SULTAN MOULAY SLIMANE
École Nationale des Sciences Appliquées de Khouribga

Département Mathématiques et Informatique

Filière : Génie informatique

Rapport TP4 : Express.JS

Ahmed BABA

Sous la direction de : Pr. Amal OURDOU

ENSAK 2024/2025

Table des matières

Table des figures	2
1 Objectif	3
2 réalisation du tp	4
2.1 Ajout des items : METHODE POST	4
2.2 Affichage d'items : METHODE GET	4
2.3 Modification d'items : METHODE PUT	6
2.4 Supression d'items : METHODE DELETE	7
3 Conclusion	8

Table des figures

2.1	Ajout d'item livre	4
2.2	Capture des items ajoutés	5
2.3	Capture des items ajoutés	5
2.4	Modification d'item 1	6
2.5	Item 1 modifié	6
2.6	Suppression d'item 1	7
2.7	Item 1 supprimé	7

Chapitre 1

Objectif

Le but de ce TP sur Express.js est de vous familiariser avec les concepts et pratiques de base du développement d'une application web en utilisant Node.js et Express.js, en particulier la gestion des requêtes HTTP pour créer une API CRUD (Create, Read, Update, Delete).

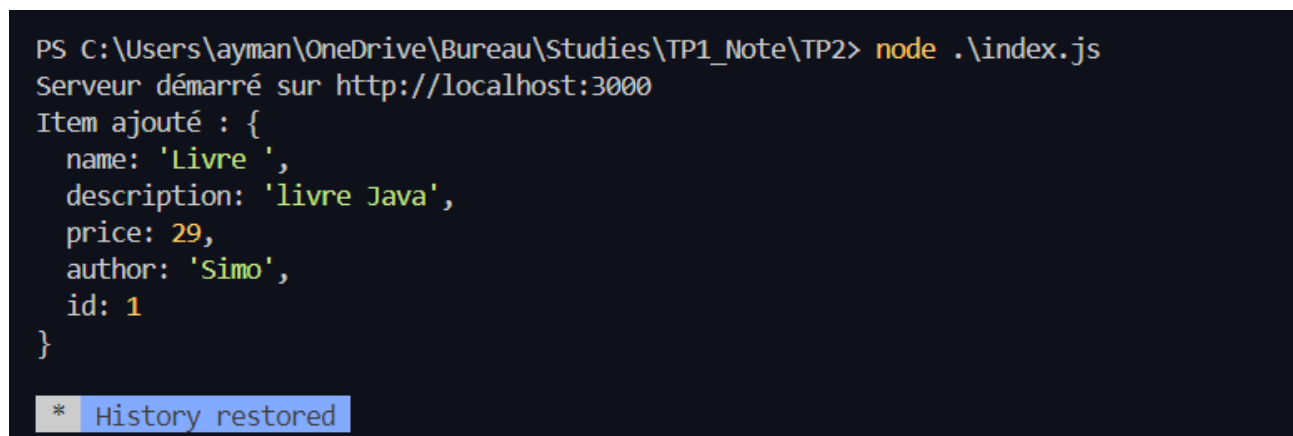
Chapitre 2

réalisation du tp

On peut ajouter les items on utilisant la commande :

Invoke-WebRequest -Uri http://localhost:3000/items/1 -Method Put -ContentType "application/json" -Body '{"name": "Livre", "description": " livre java", "price": 35, "author": "Simo"}

2.1 Ajout des items : METHODE POST



```
PS C:\Users\ayman\OneDrive\Bureau\Studies\TP1_Note\TP2> node .\index.js
Serveur démarré sur http://localhost:3000
Item ajouté : {
  name: 'Livre ',
  description: 'livre Java',
  price: 29,
  author: 'Simo',
  id: 1
}
* History restored
```

FIGURE 2.1 – Ajout d’item livre

2.2 Affichage d’items : METHODE GET

- Tout les items

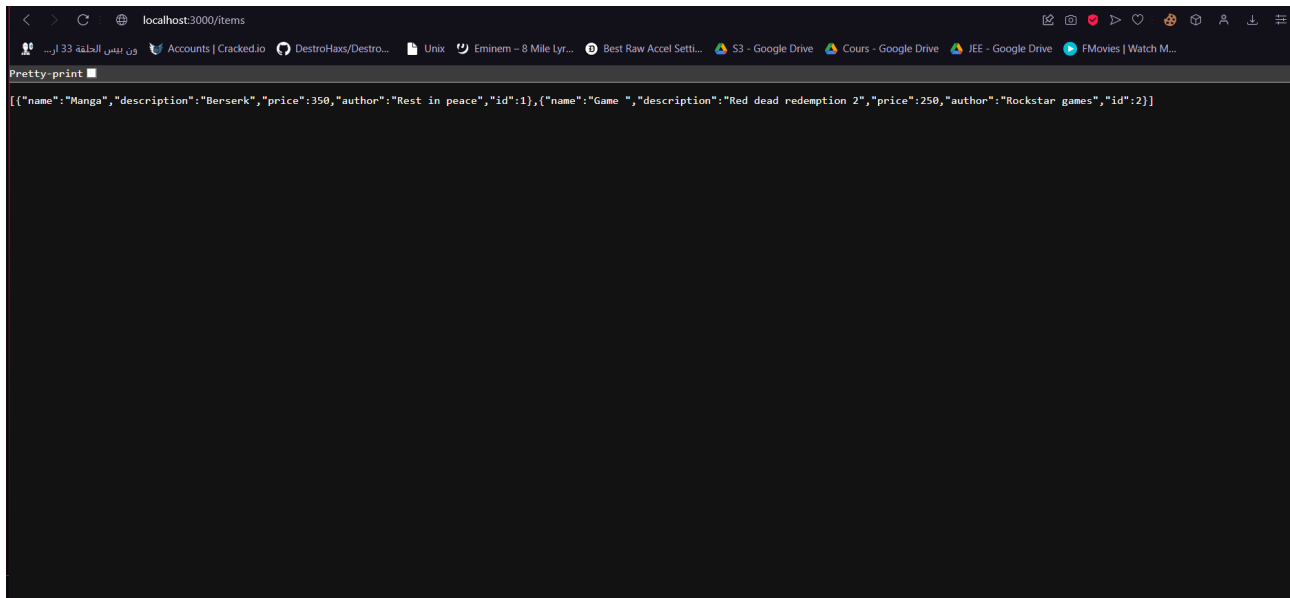


FIGURE 2.2 – Capture des items ajoutés

- Item avec id 1

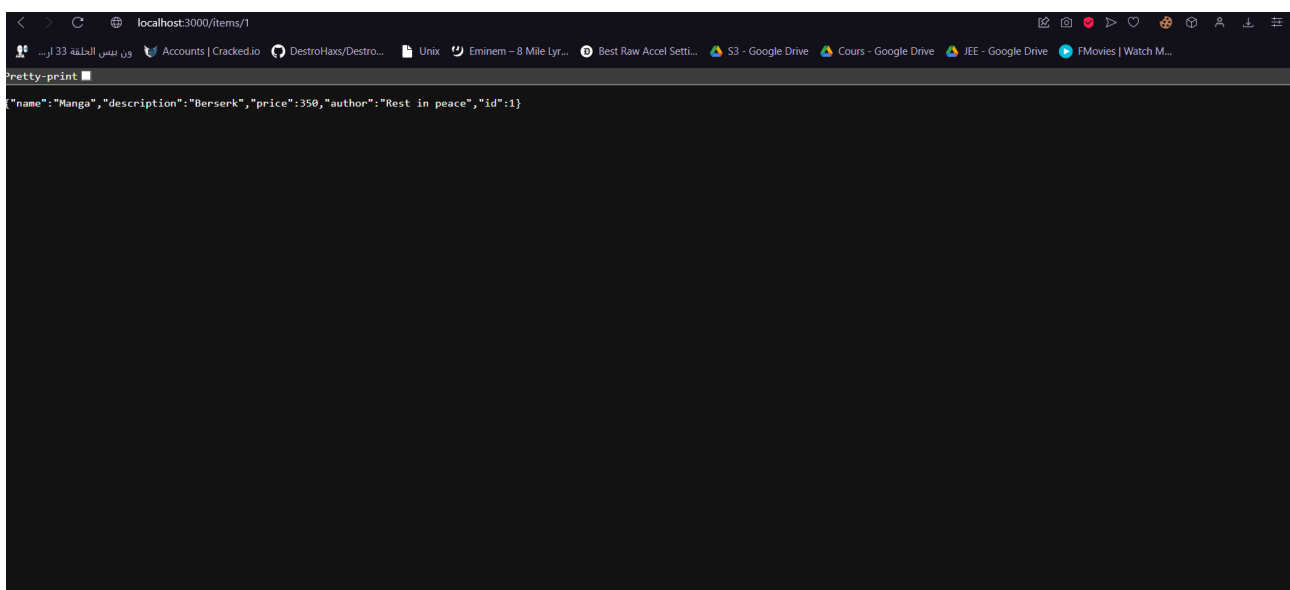


FIGURE 2.3 – Capture des items ajoutés

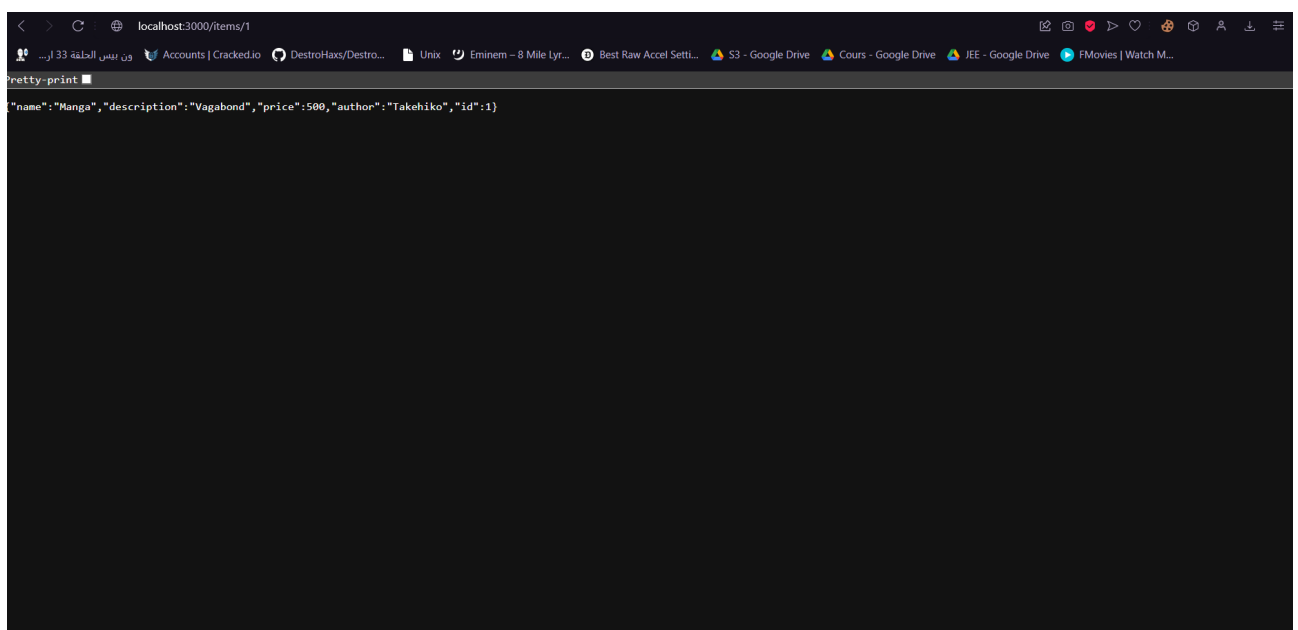
2.3 Modification d'items : METHODE PUT

```
PS C:\Users\ayman\OneDrive\Bureau\Studies\TP1_Note> Invoke-WebRequest -Uri http://localhost:3000/items/1 -Method Put -ContentType "application/json" -Body '{"name": "Manga", "description": "Vagabond", "price": 500, "author": "Takehiko"}'
```

StatusCode : 200
StatusDescription : OK
Content : Item mis à jour avec succès
RawContent : HTTP/1.1 200 OK
Connection: keep-alive
Keep-Alive: timeout=5
Content-Length: 29
Content-Type: text/html; charset=utf-8
Date: Mon, 21 Oct 2024 13:41:42 GMT
ETag: W/"1d-TueD0j1L04jggVBTPTuZff/6M2U...

Forms : {}
Headers : {[Connection, keep-alive], [Keep-Alive, timeout=5], [Content-Length, 29], [Content-Type, text/html; charset=utf-8]...}
Images : {}
InputFields : {}
Links : {}
ParsedHtml : mshtml.HTMLDocumentClass
RawContentLength : 29

FIGURE 2.4 – Modification d'item 1



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/items/1'. The page content is a JSON object: `{"name": "Manga", "description": "Vagabond", "price": 500, "author": "Takehiko", "id": 1}`. The browser's developer tools are open, showing the 'pretty-print' view of the JSON data.

FIGURE 2.5 – Item 1 modifié

2.4 Supression d'items : METHODE DELETE

```
PS C:\Users\ayman\OneDrive\Bureau\Studies\TP1_Note> Invoke-WebRequest -Uri http://localhost:3000/items/1 -Method Delete

StatusCode      : 200
StatusDescription : OK
Content         : Item supprimé avec succès
RawContent      : HTTP/1.1 200 OK
                  Connection: keep-alive
                  Keep-Alive: timeout=5
                  Content-Length: 27
                  Content-Type: text/html; charset=utf-8
                  Date: Mon, 21 Oct 2024 13:42:37 GMT
                  ETag: W/"1b-UPrdkf1lv6Y8jeZXdv/xNtc1BSA..."
Forms           : {}
Headers         : {[Connection, keep-alive], [Keep-Alive, timeout=5], [Content-Length, 27], [Content-Type, text/html; charset=utf-8]...}
Images          : {}
InputFields     : {}
Links           : {}
ParsedHtml      : mshtml.HTMLDocumentClass
RawContentLength : 27
```

FIGURE 2.6 – Suppression d'item 1

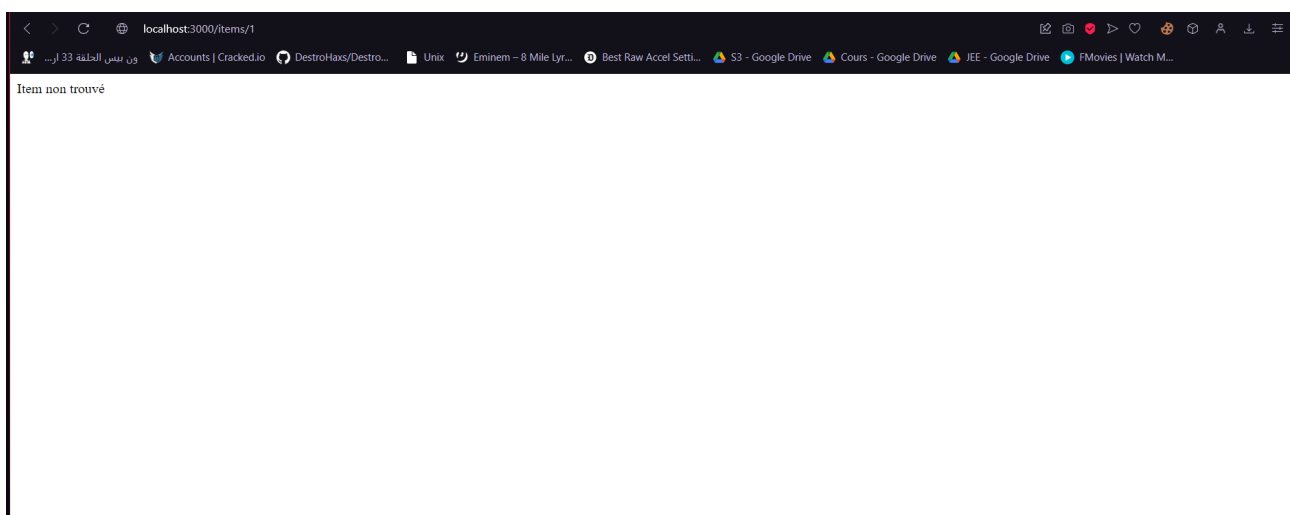


FIGURE 2.7 – Item 1 supprimé

Chapitre 3

Conclusion

Ce TP a permis de découvrir les bases d'Express.js et son utilisation pour créer une API RESTful. En réalisant les opérations CRUD (Create, Read, Update, Delete), nous avons appris à manipuler des données via des requêtes HTTP et à gérer efficacement les routes d'un serveur web. L'intégration de middlewares a montré comment transformer et traiter les requêtes avant leur réponse.

En somme, ce TP a fourni une introduction pratique à la création d'un serveur web avec Node.js et Express.js, permettant ainsi de mieux comprendre les fondements du développement d'applications web.