

TP : Express js

Realisé par :

Asmae CHADLI & Aya Fakri

Question 1 :

Express.js est un framework minimaliste pour Node.js qui simplifie le développement d'applications web et d'API en offrant des fonctionnalités robustes pour gérer des requêtes HTTP, des routes, des middlewares, etc. Il permet la création d'applications **CRUD**, d'**API RESTful**, et d'applications web évolutives. Express.js est très utilisé pour le développement de serveurs back-end et est apprécié pour sa flexibilité et sa simplicité.

Question 2 :

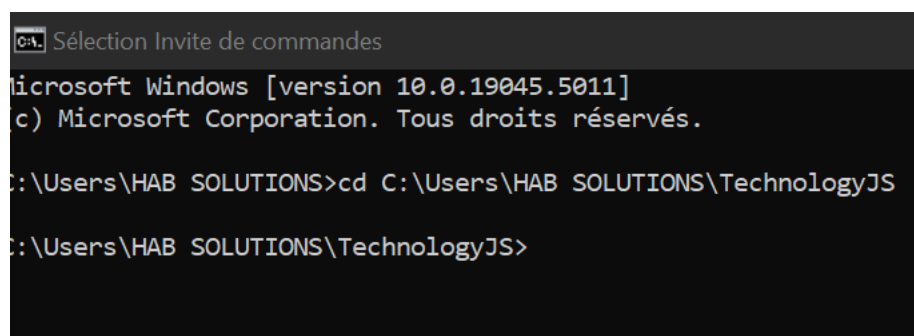
Un **middleware** dans Express.js est une fonction qui intercepte les requêtes HTTP entrantes et leur permet d'être modifiées, loggées, ou traitées avant d'atteindre les routes finales. Les middlewares permettent de réaliser des tâches comme l'authentification, la gestion des erreurs, le parsing des requêtes, etc.

Les deux exemples :

- **Middleware de journalisation (logging)** : Ce middleware est utilisé pour enregistrer des informations sur chaque requête entrante, comme le type de requête (GET, POST, etc.) ou l'URL demandée. Il permet aux développeurs de suivre ce qui se passe sur le serveur, d'analyser le trafic, ou de détecter des anomalies.
- **Middleware d'analyse du corps de la requête (body parser)** : Ce middleware est utilisé pour analyser les données envoyées dans le corps d'une requête, comme des données JSON ou URL-encoded. Il permet de convertir ces données en un format que l'application peut traiter facilement, comme un objet JavaScript.

Creating a simple CRUD application

Question 1 :



```
Selection Invite de commandes
Microsoft Windows [version 10.0.19045.5011]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\HAB SOLUTIONS>cd C:\Users\HAB SOLUTIONS\TechnologyJS
C:\Users\HAB SOLUTIONS\TechnologyJS>
```

Question 2 :

```
PS C:\Users\HAB SOLUTIONS\TechnologyJS> cd TpExpress
PS C:\Users\HAB SOLUTIONS\TechnologyJS\TpExpress> npm init -y
>>
Wrote to C:\Users\HAB SOLUTIONS\TechnologyJS\TpExpress\package.json:
```

```
{
  "name": "tpexpress",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

```
PS C:\Users\HAB SOLUTIONS\TechnologyJS\TpExpress> █
```

Question 3 :

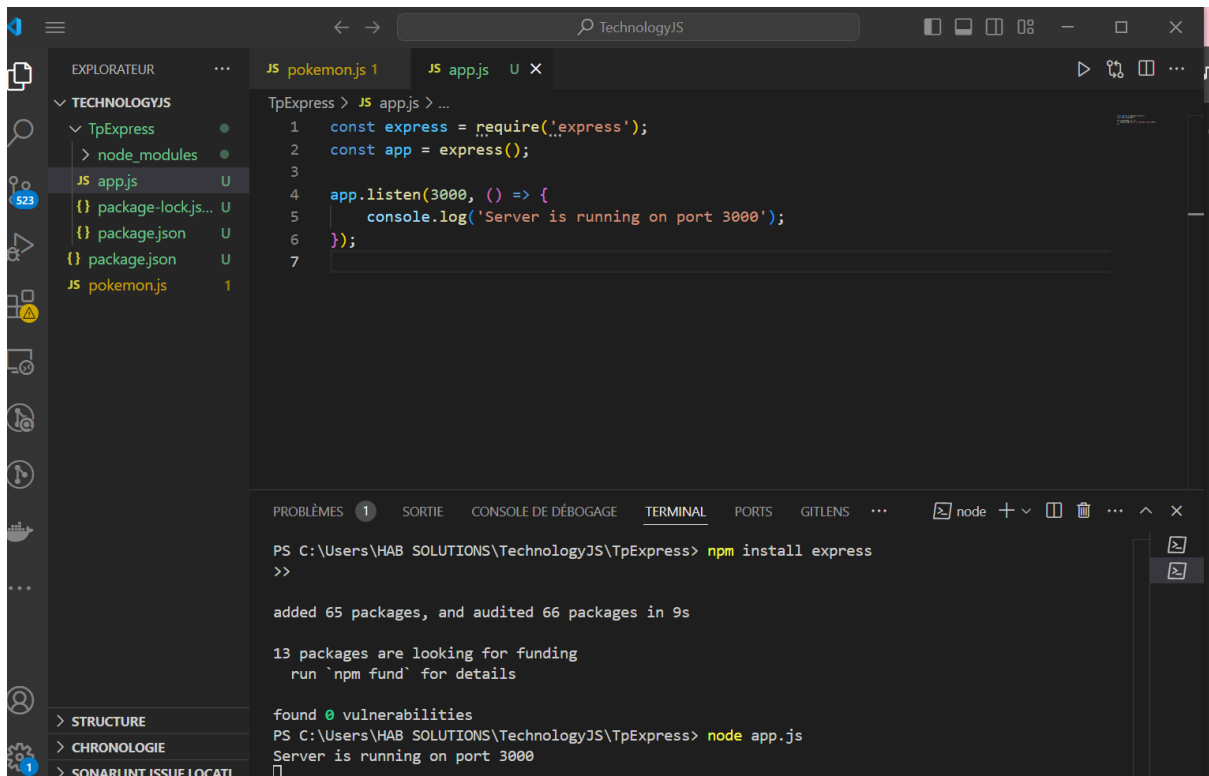
```
PS C:\Users\HAB SOLUTIONS\TechnologyJS\TpExpress> npm install express
>>

added 65 packages, and audited 66 packages in 9s

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

found 0 vulnerabilities
```

Question 4 :



Question 5 :

```
8 let items = [];
9
10 app.post('/items', (req, res) => {
11   const newItem = req.body;
12   items.push(newItem);
13   res.status(201).send('Item added');
14 });
```

Question 6 :

```
app.get('/items', (req, res) => {
  res.json(items);
});
```

Question 7 :

```

20  app.get('/items/:id', (req, res) => {
21      const id = req.params.id;
22      const item = items.find(i => i.id === id);
23      if (item) {
24          res.json(item);
25      } else {
26          res.status(404).send('Item not found');
27      }
28  });

```

Question 8 :

```

30  app.put('/items/:id', (req, res) => {
31      const id = req.params.id;
32      const updatedItem = req.body;
33      const index = items.findIndex(i => i.id === id);
34      if (index !== -1) {
35          items[index] = updatedItem;
36          res.send('Item updated');
37      } else {
38          res.status(404).send('Item not found');
39      }
40  });
41

```

Question 9 :

```

42  app.delete('/items/:id', (req, res) => {
43      const id = req.params.id;
44      const index = items.findIndex(i => i.id === id);
45      if (index !== -1) {
46          items.splice(index, 1);
47          res.send('Item deleted');
48      } else {
49          res.status(404).send('Item not found');
50      }
51  });

```

Question 10 :

```

Server is running on port 3000
PS C:\Users\HAB SOLUTIONS\TechnologyJS\TpExpress> node app.js
Le serveur est démarré sur le port 3000

```

Question 11 :

GET

http://localhost:3000/items

Send

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTests ⚡SettingsCookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

BodyCookiesHeaders (7)Test Results ⚡

200 OK25 ms235 BSave Response ▾

PrettyRawPreviewVisualizeJSON ▾

1

POST

http://localhost:3000/items

Send

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTests ⚡SettingsCookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

BodyCookiesHeaders (7)Test Results ⚡

201 Created11 ms258 BSave Response ▾

PrettyRawPreviewVisualizeHTML ▾

1 Item ajouté avec succès

GET

http://localhost:3000/items/1

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Body

Cookies

Headers (7)

Test Results

404 Not Found

15 ms

248 B

Save Response

Pretty

Raw

Preview

Visualize

HTML

1 Item not found

PUT

http://localhost:3000/items/1

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Body

Cookies

Headers (7)

Test Results

404 Not Found

33 ms

248 B

Save Response

Pretty

Raw

Preview

Visualize

HTML

1 Item not found

DELETE

▼

http://localhost:3000/items/1

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests ♦

Settings

Cookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Body

Cookies

Headers (7)

Test Results ♦

🌐 404 Not Found 9 ms 248 B Save Response ▼

Pretty

Raw

Preview

Visualize

HTML ▼

🔗

📄 🔍

1 Item not found