A green logo with white text

AI-generated content may be incorrect.

Audit de Project

Rapport de L’exercice de déploiement

|  |  |
| --- | --- |
| Date de livraison | 2025-04-13 |
| Date de réalisation | 2024-04-16 |
| En cadre par | Pr. EL Farouki |
| Réalise par | Benfakir smail |

IAWM

# Table of Contents

[I. Table of Contents 2](#_Toc195960296)

[I. Objective de TP 4](#_Toc195960297)

[II. Configuration de l’environnement 4](#_Toc195960298)

[1. Architecture General 4](#_Toc195960299)

[2. Configuration de serveur Express.js 5](#_Toc195960300)

[A. Initialiser le projet 5](#_Toc195960301)

[B. Development de code exerpress.js 5](#_Toc195960302)

[C. Le lancement du serveur Node.js 7](#_Toc195960303)

[D. Test de fonctionnement 7](#_Toc195960304)

[ Request Get 7](#_Toc195960305)

[ Request Post 7](#_Toc195960306)

[3. Réalisation du Frontend 8](#_Toc195960307)

[A. Création de l’application React 8](#_Toc195960308)

[B. Installer les dépendances 8](#_Toc195960309)

[C. Lancer l'application 9](#_Toc195960310)

[D. UI de l’application 9](#_Toc195960311)

[III. Test d’intégration avec Bruno 10](#_Toc195960312)

[A. Présentation de Bruno 10](#_Toc195960313)

[IV. déploiement GitHub 10](#_Toc195960314)

[✅ 1. Créer un dépôt (repository) sur GitHub 10](#_Toc195960315)

[✅ 2. Initialiser Le projet localement 10](#_Toc195960316)

[✅ 3. Lier ton projet au dépôt GitHub 10](#_Toc195960317)

[✅ 4. Ajouter les fichiers et faire un commit 10](#_Toc195960318)

[✅ 5. Pusher vers GitHub 11](#_Toc195960319)

[✅6. référence de lien GitHub 11](#_Toc195960320)

[V. Conclusion 11](#_Toc195960321)

# Objective de TP

L'objectif de ce TP est de créer une application web Full Stack en utilisant React.js pour le frontend et Express.js pour le backend. L’application doit permettre d’interagir avec une API RESTful pour effectuer des opérations CRUD sur une ressource fictive nommée "tâches". Ce TP vise également à comprendre le fonctionnement d’un serveur Node.js, l’intégration entre client et serveur, ainsi que le déploiement sur GitHub.

# Configuration de l’environnement

## Architecture General

client/

├── public/

│ ├── index.html

├── src/

│ ├── components/

│ │ ├── TaskForm.js

│ │ └── TaskList.js

│ ├── services/

│ │ └── api.js

│ ├── App.js

│ ├── index.js

├── package.json

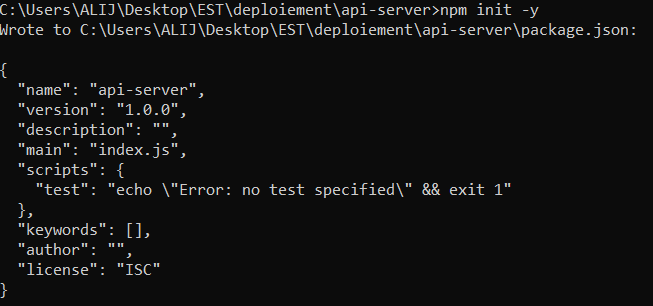
└── ...

## Configuration de serveur Express.js

### Initialiser le projet

Initialisation du projet Node.js avec la commande :

* npm init -y



Installation des dépendances nécessaires pour le backend :

* npm install express cors body-parser

A screen shot of a computer

AI-generated content may be incorrect.

### Development de code exerpress.js

Implémentation de « CRUD » au niveau du serveur

const express = require('express');

const cors = require('cors');

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

const app = express();

const PORT = process.env.PORT || 5000;

*// Middleware*

app.use(cors());

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: true }));

*// ----------------------*

*// Mock data and routes*

*// ----------------------*

let tasks = [

  { id: 1, title: 'Apprendre Express', completed: false },

  { id: 2, title: 'Créer une API REST', completed: false }

];

*// GET all tasks*

app.get('/api/tasks', (*req*, *res*) => {

*res*.json(tasks);

});

*// GET task by ID*

app.get('/api/tasks/:id', (*req*, *res*) => {

  const task = tasks.find(*t* => *t*.id === parseInt(*req*.params.id));

*if* (!task) *return* *res*.status(404).json({ error: 'Tâche non trouvée' });

*res*.json(task);

});

*// POST a new task*

app.post('/api/tasks', (*req*, *res*) => {

  const newTask = {

    id: tasks.length + 1,

    title: *req*.body.title,

    completed: false

  };

  tasks.push(newTask);

*res*.status(201).json(newTask);

});

*// Base route*

app.get('/', (*req*, *res*) => {

*res*.json({ message: 'API opérationnelle' });

});

*// Start the server*

app.listen(PORT, () => {

  console.log(`Serveur en écoute a le port ${PORT}`);

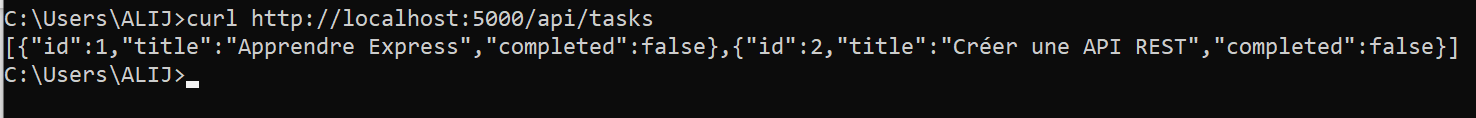
});

### Le lancement du serveur Node.js

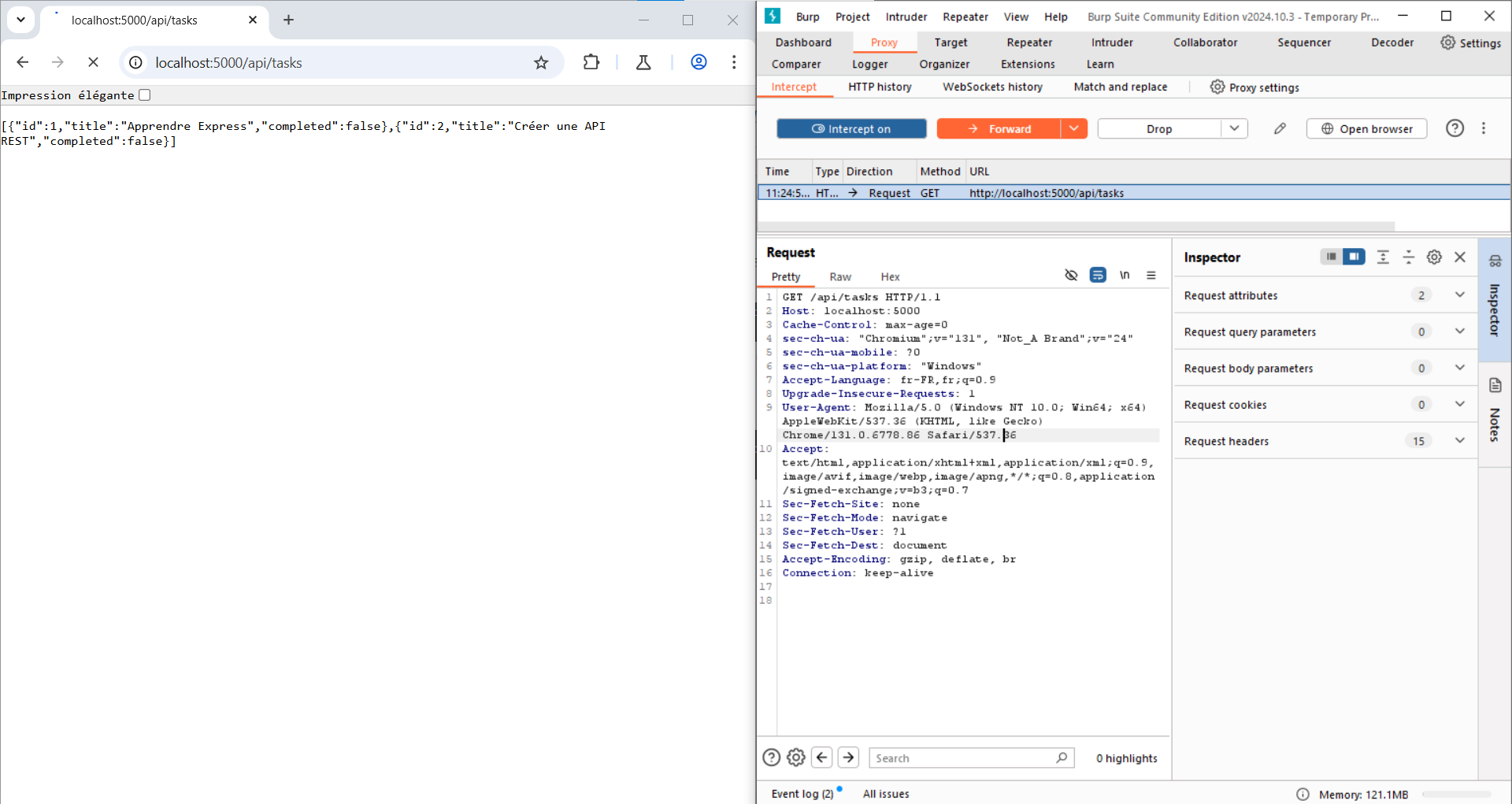
De manière générale, on peut utiliser la commande suivante :  
  
A black background with white text

AI-generated content may be incorrect.

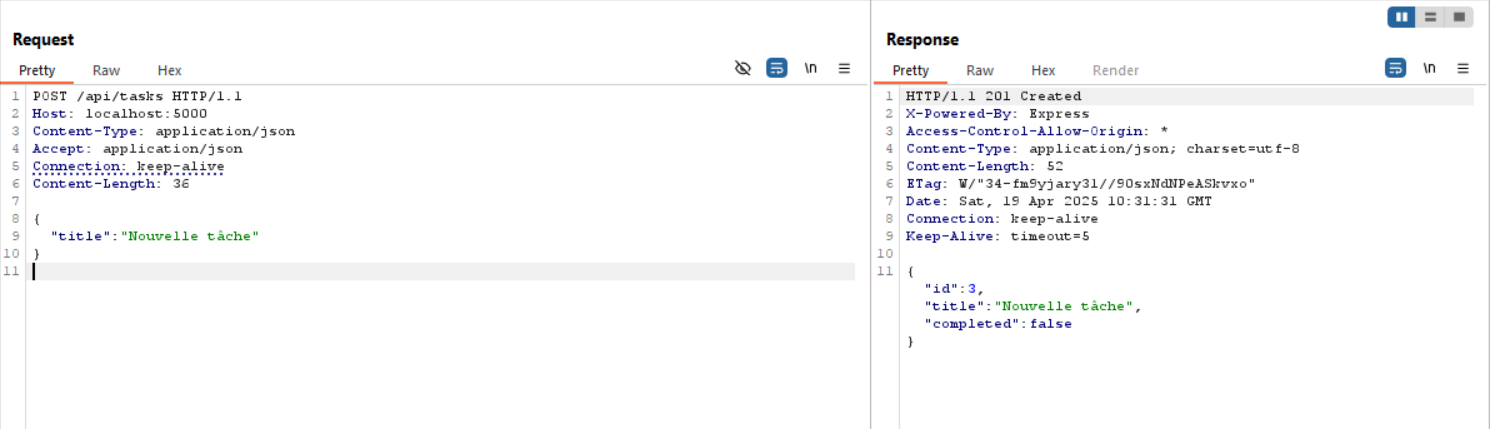
### Test de fonctionnement

* Command curl
* Tool Burp suite

### Request Get



### Request Post



## Réalisation du Frontend

### Création de l’application React

A screenshot of a computer program

AI-generated content may be incorrect.

### Installer les dépendances

A screen shot of a computer

AI-generated content may be incorrect.

### Lancer l'application

A screen shot of a computer code

AI-generated content may be incorrect.

### UI de l’application



# Test d’intégration avec Bruno

### Présentation de Bruno

**Bruno** est une application desktop open-source conçue pour l’envoi de requêtes HTTP. Elle permet aux développeurs de créer, tester et organiser leurs appels API de manière efficace et rapide

Référence :

<api-server-bruno.rar>

# déploiement GitHub

### ✅ 1. Créer un dépôt (repository) sur GitHub

1. Va sur [github.com](https://github.com)
2. Clique sur **"New repository"**
3. Clique sur **"Create repository"**

### ✅ 2. Initialiser Le projet localement

Si tu n’as pas encore de Git dans ton dossier :

* git init

### ✅ 3. ****Lier ton projet au dépôt GitHub****

Copie l’URL que GitHub te donne après la création du repo (ex : https://github.com/tonuser/ton-projet.git)

Puis dans ton terminal :

* git remote

add origin https://github.com/smailkali/API\_project

### ✅ 4. ****Ajouter les fichiers et faire un commit****

* git add .
* git commit -m "Premier commit"

### ✅ 5. Pusher vers GitHub

Si c’est ton premier push :

* git branch -M main

Puis :

* git push -u origin main

### ✅6. référence de lien GitHub

Lien :  
<https://github.com/smailkali/API_Project/tree/master>

# Conclusion

Ce travail pratique a permis d’acquérir une expérience concrète dans la création d’une application web Full Stack en utilisant les technologies modernes comme React.js pour le frontend et Express.js pour le backend. À travers les différentes étapes du projet – de la configuration de l’environnement au développement des fonctionnalités CRUD, en passant par les tests avec Bruno et le déploiement sur GitHub – nous avons pu comprendre de manière approfondie le processus complet de développement et de déploiement d'une application web.