Chapitre 4
Gestion des erreurs et sécurité

Gestion des erreurs et sécurité

- Quand on développe une API en Node.js sans framework, c'est à nous de gérer
  - Sécurité (JSON malformé, injections, …)
  - Validation (données incorrectes)
  - Robustesse (crash, erreurs quelconques)

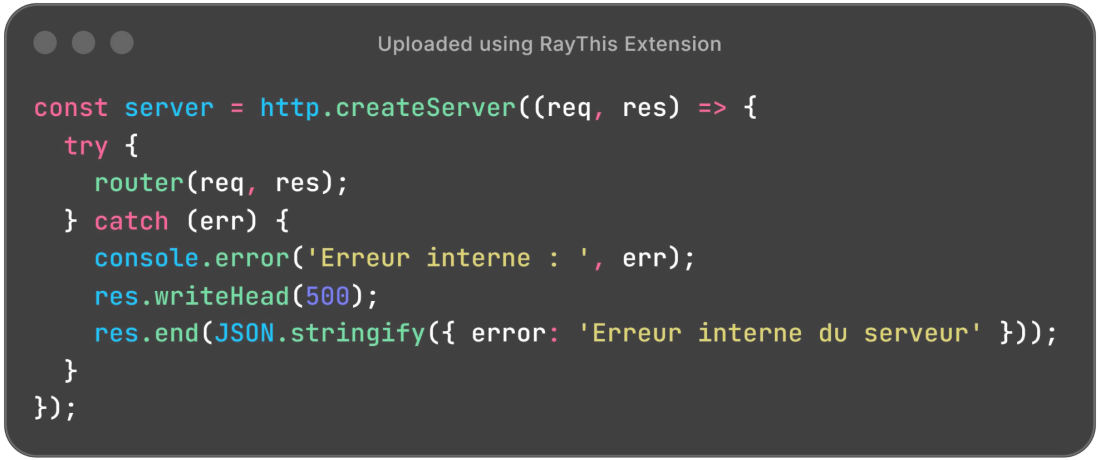
- Ne jamais faire confiance aux données reçues
- Ne jamais laisser un utilisateur faire planter l'application

Validation des entrées

- Exemple : on vérifie qu'un user contient au moins un name
- Protège contre les champs manquants, JSON corrompu

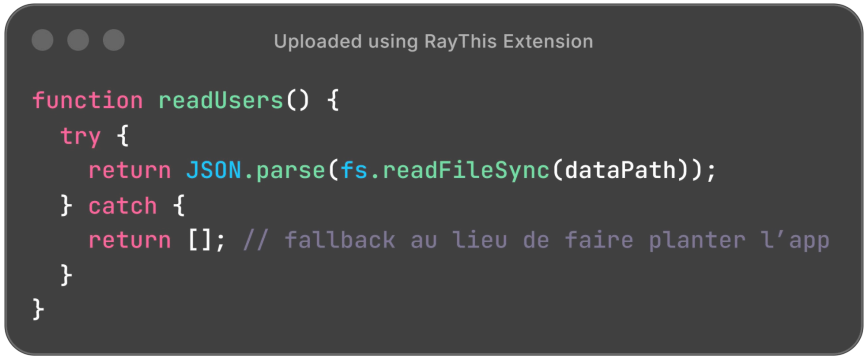
Empêcher les crash

- Dans server.js, on encapsule le routeur pour éviter les erreurs non catchées



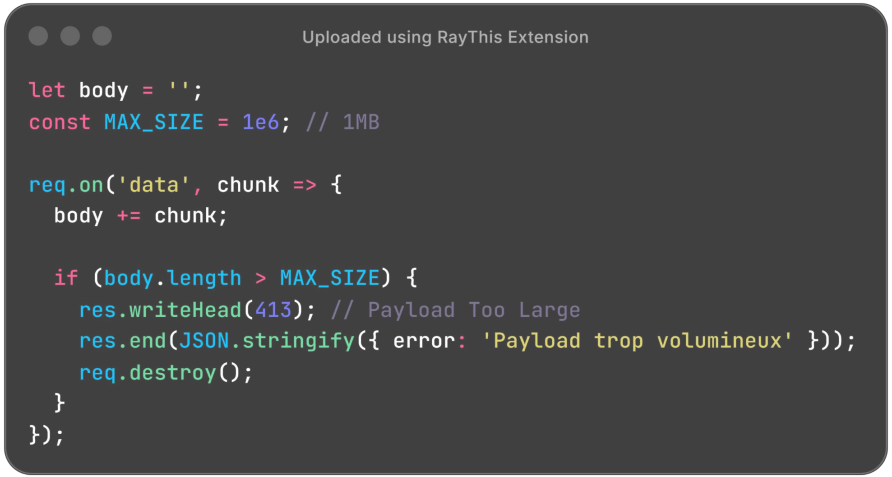
Sécurisation des accès aux fichiers

- Ajout de try/catch dans les fonctions fs



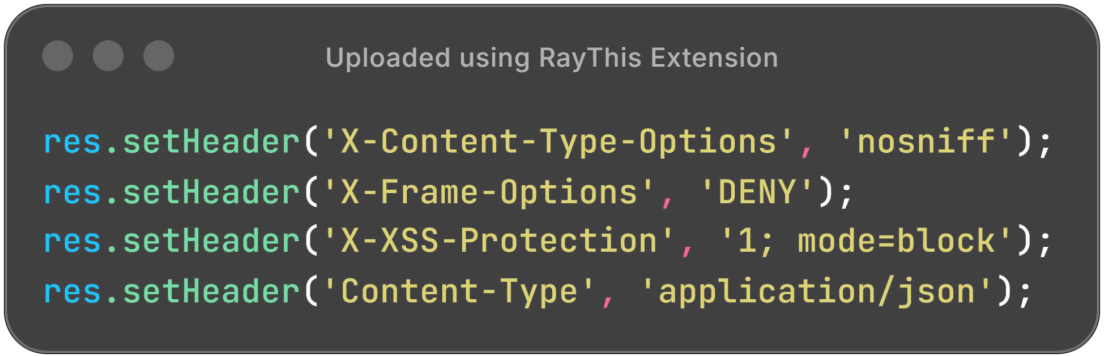
Protection contre les gros payloads (limiter taille JSON)

- Évite une attaque qui fait saturer la RAM par exemple



Ajout des headers “classiques” de sécurité

- Empêcher des attaques navigateurs



Ajout d’un service logger

- A utiliser après chaque catch() au moins

Exercices

Chapitre 5
EJS Templates

Introduction à EJS et configuration

- Un moteur de template utilise un langage différent pour définir un template HTML
- EJS pour Embedded Javascript Templates (côté client et serveur) est un moteur de template
- https://ejs.co/


```
app.set('views', './views');
app.set('view engine', 'ejs');

let message = "Hello World!";

app.get('/', (req, res) => {
  res.render('home', { unMessage:message });
});

// On récupère les fichiers statiques dans le dossier public
app.use(express.static('public'));
```

Une page EJS dans /views/home.ejs

Uploaded using RayThis Extension

```
<h1>Voici un message sur la page home.ejs</h1>
<h2><%= unMessage %></h2>
```

Layout EJS

Uploaded using RayThis Extension

```
<%- include('_header') %>
<body>
  <%-include ("_product_form")%>
</body>
<%-include ("_footer")%>
```

Les différentes balises avec EJS

Uploaded using RayThis Extension

```
<%= 'Scriptlet' %> // Affiche une variable
<%- 'Scriptlet non échappé' %> // Affiche une variable sans échapper les caractères HTML
<% if (condition) { %> // Exécute du code conditionnel
  <p>Condition vraie</p>
<% } else { %> // Exécute du code si la condition est fausse
  <p>Condition fausse</p>
<% } %> // Fin du bloc conditionnel
<% for (let i = 0; i < array.length; i++) { %> // Boucle sur un tableau
  <p>Élément: <%= array[i] %></p>
<% } %> // Fin de la boucle
```

Exercices

TP 3 : Validation des acquis