



FAST AND LOW OVERHEAD WEB FRAMEWORK, FOR NODE.JS

Fork 2.2k

Star 29.8k

Gestion Headers

Fastify permet une manipulation simple et efficace des en-têtes HTTP à l'aide de méthodes natives de Node.js.

```
app.get('/', async (request, reply) => {  
  // Add a header to the response  
  reply.header('Content-Type', 'application/json');  
});
```

DB Drivers

Fastify est compatible avec divers pilotes de bases de données tels que MongoDB, PostgreSQL, MySQL, MariaDB, etc.

```
fastify.register(require('@fastify/mysql'), {  
  connectionString: 'mysql://root@localhost/mysql'  
});
```

HTTP, Route Handling, Async

Fastify propose une gestion avancée et souple des requêtes HTTP à travers un système de routage efficace. Sa syntaxe intuitive permet de définir aisément des routes et des gestionnaires de requêtes, simplifiant ainsi le développement d'applications web.

En s'appuyant sur l'environnement Node.js, Fastify gère naturellement les opérations asynchrones de manière optimisée, utilisant les fonctions de rappel, les promesses, ainsi que les async/await.



```
// Route handler for GET /  
fastify.get('/', async (request, reply) => {  
  // Example of an asynchronous operation  
  const data = await fetchDataFromDatabase();  
  // Send the fetched data as response  
  reply.send(data);  
});
```

```
// Function simulating fetching data from a database asynchronously  
async function fetchDataFromDatabase() {  
  return new Promise((resolve, reject) => {  
    // Simulate fetching data after a short delay  
    setTimeout(() => {  
      resolve({ message: 'Data fetched from database' });  
    }, 1000);  
  });  
}
```

Type & Validation

Fastify intègre nativement la validation des données via JSON Schema.

```
// JSON Schema to validate user data  
const userSchema = {  
  type: 'object',  
  properties: {  
    username: { type: 'string', minLength: 3 },  
    email: { type: 'string', format: 'email' }  
  },  
  required: ['username', 'email']  
};
```

Ecosystem

Fastify bénéficie d'un écosystème de plugins et de modules tiers qui étendent ses fonctionnalités de base.

63

core plugins

226

community plugins

Environnement

Fastify peut être déployé sur divers serveurs web, y compris les serveurs HTTP natifs de Node.js, ainsi que Nginx ou Apache en utilisant des proxys inversés.

Performances

Fastify est reconnu pour ses performances élevées, ce qui en fait l'un des avantages majeurs du framework.

Bien qu'il ne propose pas directement de fonctionnalités de multi-threading ou de gestion des workers, il peut être intégré avec des outils et des stratégies, tels que le clustering de Node.js, pour améliorer davantage ses performances.

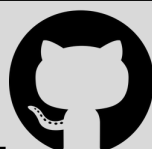
Folder Structure / Workflow

Fastify ne dicte pas de structure de dossier spécifique, mais il est recommandé de suivre les bonnes pratiques générales de structuration des projets Node.js.

```
> config  
> controllers  
> models  
> node_modules  
> routes  
> .gitignore  
package-lock.json  
package.json  
server.js
```

Pour notre projet warmup nous avons nottament pu mettre en place un architecture MVC classique.

Community, Documentation



La communauté de Fastify est principalement présente sur GitHub, offrant un lieu d'échange pour poser des questions, signaler des problèmes et participer au développement du code.

En revanche, le framework étant assez récent (2017) il peut être difficile de trouver des projets fastify ou des retours utilisateurs.

Pour ce qui est de la documentation officielle, elle est plutôt complète et elle propose des guides, des tutoriels et des exemples pour accompagner les utilisateurs dans leur utilisation du framework.

Models, Seeders

Fastify n'offre pas directement de fonctionnalités pour les modèles et les seeders, mais il est compatible avec des ORM tel que Sequelize que nous avons intégré dans notre projet.



Plus :

- Compatibilité ESM;
- Utilisation similaire à Express JS;
- Rapidité des requêtes.

Moins :

- Manque de ressources communautaires (stackoverflow désert);