

#pw2025

Power Week 2025

18 - 19 - 20 novembre 2025

IBM Innovation Studio Paris

S15 – Réalisez une API rapidement et documentée avec Fastify

18 novembre 13:30 - 14:30

Sébastien BOURREAU

COOPRL

sebastien.bourreau@cooperl.com



Sébastien BOURREAU

Administrateur IBMi et Concepteur développeur chez Cooperl.

- Cooperl est une société coopérative agricole (SCA)
- Fondée le 19 avril 1966 par Sébastien Coupé
- Spécialisée dans l'élevage porcin
- Présence internationale (Amérique, Asie et Europe)



+6600
Adhérents & Clients

+7400
Salariés

2,92 Mrd
€ Chiffre d'affaire
2023

+160
Salariés à la DSI



Agenda

01

Introduction

02

Comparatif

03

Fonctionnement

04

Documentation

05

Démo

06

Conclusion

Power Week

18 -19 - 20 novembre 2025

01

Introduction

IBM
common
FRANCE

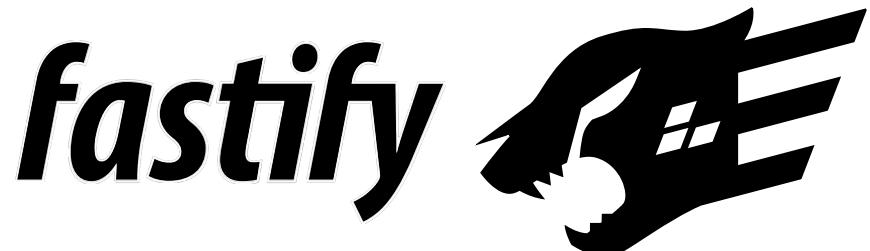
IBM

Introduction

Fastify est un framework web JavaScript qui vise à offrir la meilleure expérience possible aux développeurs et une architecture de plugins puissante.

Il s'inspire de Hapi et Express, il s'agit de l'un des frameworks web les plus rapides du marché.

- Haute performance
- Extensible
- Basé sur des schémas
- Convivial
- Compatible avec TypeScript



Installation

Pré-requis:

- ADDENVVAR ENVVAR (QIBM_MULTI_THREADED) VALUE (Y)
- Installation NodeJS
- OpenSSH Server

Installation avec npm:

```
npm i fastify
```

Installation avec yarn:

```
yarn add fastify
```

Power Week

18 -19 - 20 novembre 2025



common

FRANCE



02

Comparatif

Comparatif - Minimal

Express

```
1. const express =  
require('express')  
2. const app = express()  
  
3. app.get('/', function (req,  
res) {  
  4.   res.send({ hello: 'world' })  
  5. })  
  
6. app.listen(3000)
```

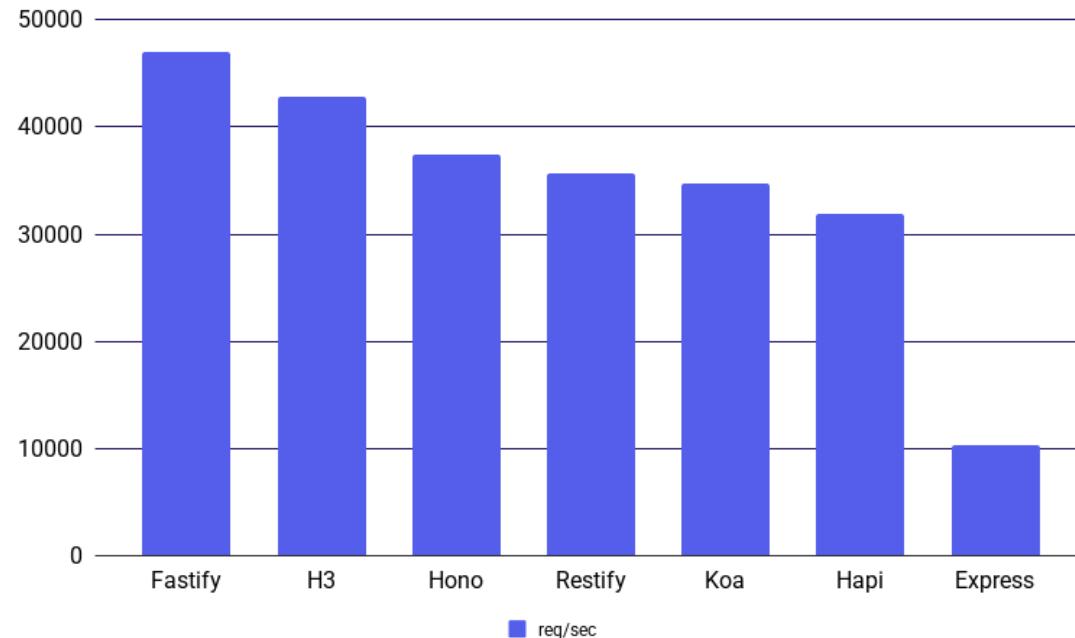
Node core

```
1. const http = require('http')  
  
2. const server =  
http.createServer(handler)  
  
3. function handler (req, res) {  
  4.   res.end(JSON.stringify({ hello:  
'world' }))  
  5. }  
  
6. server.listen(3000)
```

Fastify server

```
1. const fastify =  
require('fastify')()  
  
2. fastify.get('/', (req, reply) =>  
{  
  3.   reply.send({ hello: 'world' })  
  4. })  
  
5. fastify.listen(3000, err => {  
  6.   if (err) throw err  
  7. })
```

Comparatif - Performance



Source : <https://fastify.dev>

Comparatif - Fonctionnalités

Feature	Routing	Middleware	Plugins	Decorators	Hooks	Validation	Logging	Async/Await	HTTP2
Express	✓	✓	✗	✗	✗	✗	✗	✗	✗
Hapi	✓	✗	✓	✓	✓	✓	✓	✓	✓
Fastify	✓	✓	✓	✓	✓	✓	✓	✓	✓

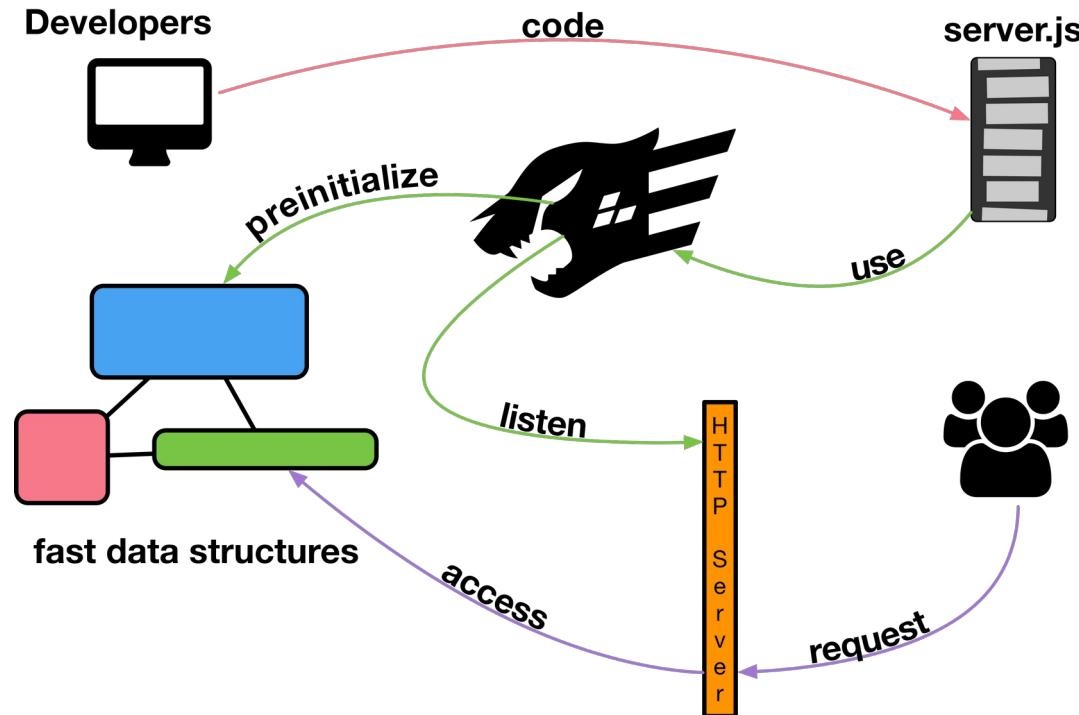
Source : <https://fastify.dev>



03

Fonctionnement

Fonctionnement



Source : <https://fastify.dev>

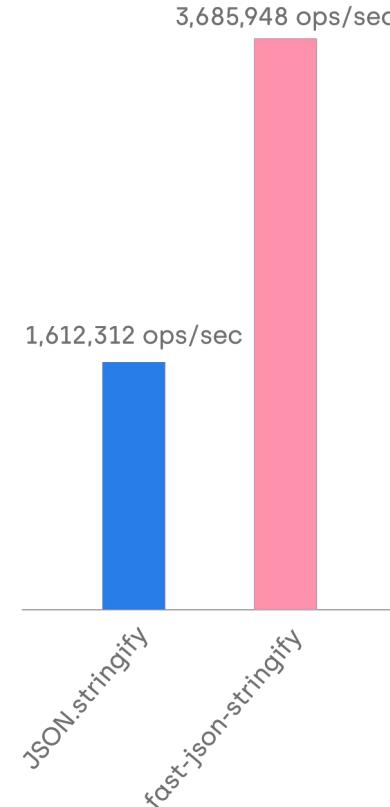
Fonctionnement

Le problème avec `JSON.stringify` :

- Fonction récursive
- Difficile à optimiser
- Code générique
- Ne peut pas être spécifique à un type

La solution : `fast-json-stringify` :

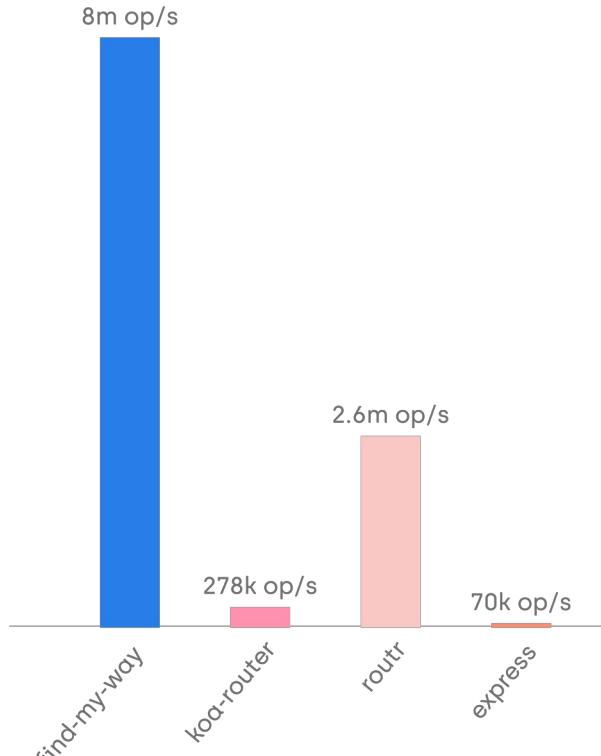
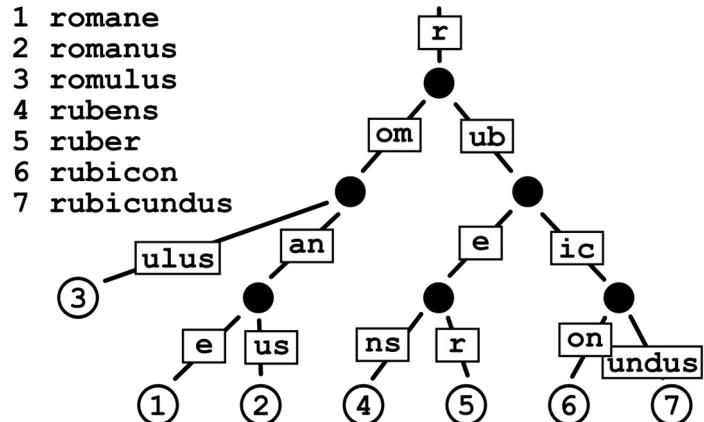
- rendu JSON basé sur un schéma
- génère du code basé sur le schéma
- Empêche les fuites d'informations



Fonctionnement

Le routeur HTTP find-my-way :

- Pas de fermeture
- Pas d'expressions régulières
- Construit sur un arbre radix



Fonctionnement

Validation :

```
const fastify = require('fastify')()

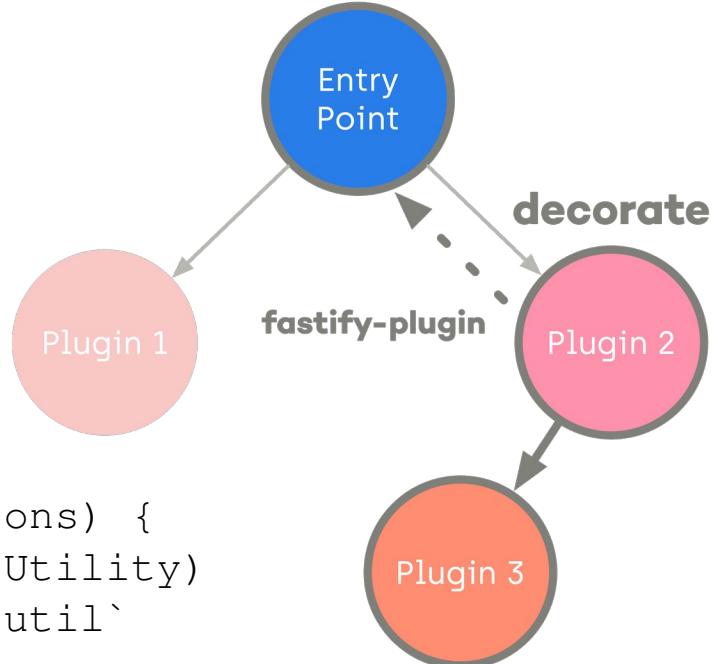
const options = {
  schema: {
    body: {
      type: 'object',
      properties: {
        greeting: { type: 'string' }
      },
      required: ['greeting']
    }
  }
}
```

Fonctionnement

Le système de plugin

```
// register a plugin  
fastify.register(...)
```

```
const fp = require('fastify-plugin')  
async function myPlugin (fastify, options) {  
  fastify.decorate('util', yourAwesomeUtility)  
  // now you can use it with `fastify.util`  
}  
module.exports = fp(myPlugin)
```



Tout ce qui peut être un plugin doit être un plugin.



04

Documentation

Documentation

OpenAPI, connu dans ces précédentes versions sous le nom de Swagger, est une spécification qui permet de décrire l'API d'un service web sous la forme d'un document YAML ou JSON.

Utilisation du plug-in *fastify-swagger-ui*

Documentation

```
await app.register(import('@fastify/swagger-ui'), {
  routePrefix: '/documentation',
  uiConfig: {
    docExpansion: 'list',
    deepLinking: false
  }
})

app.put('/some-route/:id', {
  schema: {
    description: 'post some data',
    tags: ['user', 'code'],
    summary: 'qwerty',
    Params: {
      ...
    },
    body: {
      ...
    },
    Response: {
      ...
    },
  }
}, (req, reply) => {})
```

Power Week

18 -19 - 20 novembre 2025



common

FRANCE



05

Démo

Démo

```
JS server.js  X

JS server.js > ⚡ handler
1 // Import the framework and instantiate it
2 import Fastify from 'fastify'
3 const fastify = Fastify({
4   logger: true
5 })
6
7 // Declare a route
8 fastify.get('/', async function handler [request, reply] {
9   return { hello: 'world' }
10 })
11
12 // Run the server!
13 try {
14   // ATTENTION à bien renseigner le host
15   // sinon écoute seulement sur l'adresse localhost
16   await fastify.listen({ port: 8085, host: '0.0.0.0' })
17 } catch (err) {
18   fastify.log.error(err)
19   process.exit(1)
20 }
```

JSON	Données brutes	En-têtes
{ "hello": "world" }		
<hr/>		

Démo

```
JS server.js  X  
JS server.js > ...  
2 import Fastify from 'fastify'  
3 import firstRoute from './our-first-route.js'  
4  
5 const fastify = Fastify({  
6   logger: true  
7 })  
8  
9 // Register a route  
10 fastify.register(firstRoute)  
11  
12 // Run the server!  
13 try {  
14   // ATTENTION à bien renseigner le host  
15   // sinon écoute seulement sur l'adresse localhost  
16   await fastify.listen({ port: 8085, host: '0.0.0.0' })  
17 } catch (err) {  
18   fastify.log.error(err)  
19   process.exit(1)  
20 }
```

Démo

JS our-first-route.js ×

```
JS our-first-route.js > ...
1  /**
2   * Encapsulates the routes
3   * @param {FastifyInstance} fastify Encapsulated Fastify Instance
4   * @param {Object} options plugin options
5   */
6  async function routes (fastify, options) {
7    fastify.get('/', async (request, reply) => {
8      return { hello: 'world-2' }
9    })
10 }
11
12 export default routes;
```

JSON Données brutes En-têtes

Enregistrer Copier Formater et indenter

{"hello": "world-2"}

Démo

```
JS server.js > ...
JS server.js > ...
2 import Fastify from 'fastify'
3 import swaggerUi from '@fastify/swagger-ui'
4 import swagger from '@fastify/swagger'
5
6 import firstRoute from './our-first-route.js'
7
8 const fastifySwagger = {
9   openapi: {
10     info: { ... },
11     tags: [ ... ],
12   },
13   refResolver: { ... }
14 }
15
16 const fastifySwaggerUiOptions = {
17   routePrefix: '/documentation',
18   uiConfig: { ... },
19   uiHooks: { ... },
20   staticCSP: false,
21   transformStaticCSP: (header) => header,
22   transformSpecification: (swaggerObject, request, reply) => { return swaggerObject },
23   transformSpecificationClone: true
24 }
25
26 const fastify = Fastify({
27   logger: true
28 })
29
30 // Register Swagger
31 await fastify.register(swagger, fastifySwagger)
32 await fastify.register(swaggerUi, fastifySwaggerUiOptions)
33
34 // Register a route
35 fastify.register(firstRoute)
36
37 // Run the server!
38 try {
39   // ATTENTION à bien renseigner le host
40   // sinon écoute seulement sur l'adresse localhost
41   await fastify.listen({ port: 8085, host: '0.0.0.0' })
42 } catch (err) {
43   fastify.log.error(err)
44   process.exit(1)
45 }
```

Démo

JS our-first-route.js X

```
JS our-first-route.js > [e] default
1  /**
2   * Encapsulates the routes
3   * @param {FastifyInstance} fastify Encapsulated Fastify Instance
4   * @param {Object} options plugin options
5   */
6  async function routes (fastify, options) {
7    fastify.get('/', {
8      schema: {
9        description: 'First route',
10       tags: ['First route'],
11       summary: 'Our first route',
12       response: {
13         200: {
14           description: 'Successful response',
15           type: 'object',
16           properties: {
17             hello: { type: 'string' }
18           }
19         },
20         500: {
21           description: 'Internal server error',
22         }
23       }
24     }
25   }, async (request, reply) => {
26     return { hello: 'world-3' }
27   })
28 }
29
30 export default routes;
```

JSON

Données brutes

En-têtes

Enregistrer

Copier

Formater et indenter

```
{"hello":"world-3"}
```

Démo

IBM PowerWeek <http://gigtest:8085/documentation/json>

Demonstrates Fastify documentation

First route The first route **GET** / Our first route

First route

Parameters[Try it out](#)

No parameters

Responses**Code****Description****Links**

200 Successful response

No links

Media type

[application/json](#) 

Controls Accept header.

[Example Value](#) | [Schema](#)

```
{  
  "hello": "string"  
}
```

500 Internal server error

No links

Media type

[application/json](#) [Example Value](#) | [Schema](#)

```
"string"
```

Démo

GET / Our first route Cancel

First route

Parameters

No parameters

Responses

Curl

```
curl -X 'GET' \
'http://gigctest:8085/' \
-H 'accept: application/json'
```

Request URL

```
http://gigctest:8085
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "hello": "world-3" }</pre> <p>Download</p> <p>Response headers</p> <pre>connection: keep-alive content-length: 19 content-type: application/json; charset=utf-8 date: Sun, 16 Nov 2025 08:56:41 GMT keep-alive: timeout=72</pre>

Démo

Authorize 

login Login endpoints

POST `/login` Return a token when success 

POST `/logout` To implement 

customers Customers-related endpoints

GET `/api/customer/` Return all customers  

POST `/api/customer/` Return a customer  

GET `/api/customer/{cusnum}` Return a customer  

PUT `/api/customer/{cusnum}` Update a customer  

DELETE `/api/customer/{cusnum}` Delete a customer  

Démo

Code	Description	Links
200	Successful response	No links
	<p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>[{ "number": 0, "last_name": "string", "initials": "string", "street": "string", "city": "string", "state": "string", "zip_code": 0, "credit_limit": 0, "CICID": 0, "BALDUE": 0, "COTDUE": 0 }]</pre>	
400	Bad request	No links
	<p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div> <p>Example Value Schema</p> <pre>{}</pre>	
403	Forbidden	No links
	<p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div> <p>Example Value Schema</p> <pre>{}</pre>	
500	Internal server error	No links
	<p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div> <p>Example Value Schema</p> <pre>{ "statusCode": 0, "error": "string", "message": "string" }</pre>	



06

Conclusion

Conclusion

A vous de l'essayer ...

Pour aller plus loin...

Fastify: <https://fastify.dev/>

Module Swagger: <https://github.com/fastify/fastify-swagger>

Module Swagger-ui: <https://github.com/fastify/fastify-swagger-ui>

PM2: <https://pm2.keymetrics.io/>

Github de la démo: <https://github.com/sebCIL/api-with-fastify>

MERCI