

# Unifié et Universel, le JavaScript avec UnJS

# Estéban Soubiran

Développeur web chez Infomaniak

Élève-ingénieur en cyber-sécurité

Membre d'UnJS



esteban-soubiran.site

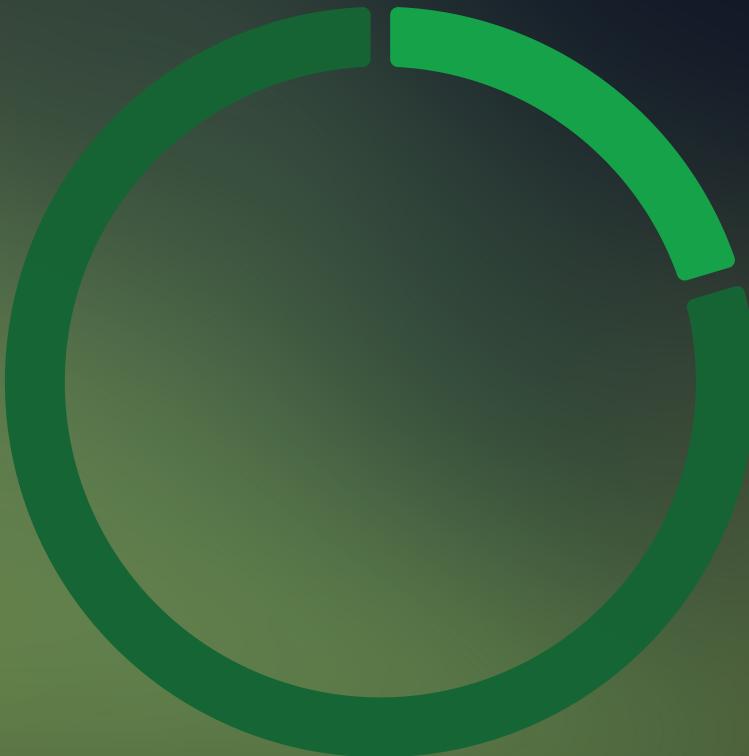


barbapapazes

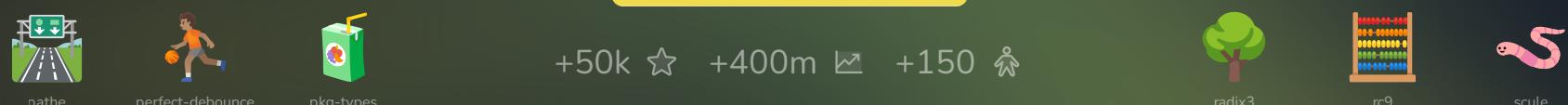


soubiran\_

# Connaissez-vous UnJS ?



- 0 - Je l'utilise régulièrement
- 5 - Oui mais de nom
- 20 - Non





# ofetch

Une meilleure API de `fetch` qui fonctionne avec Node, le navigateur et les workers.

- ☝️ Création d'une instance pré-configurable
- ☝️ Simple et Fonctionnel
- ☝️ Utilisable avec tous les runtimes

```
import { ofetch } from 'ofetch'

const api = ofetch.create({
  baseURL: 'https://api.example.com',
  async onRequestError({ request, options, error }) {
    // Use interceptors to handle errors
  },
})

const data = await api('/api/posts')
/***
 * {
 *   data: [
 *     { id: 1, title: 'Hello' },
 *     { id: 2, title: 'World' },
 *   ],
 * }
 */
```



Le gestionnaire de paquets unifié  
pour Node.js et Bun.

CLI

```
# 🌟 Auto-detect  
npx nypm i my-super-dep
```

Programmatiquement

```
import { addDependency } from 'nypm'
```

⚡ Détection automatique

🔌 Interface pour interagir programmatiquement



# unbuild

Un système de *build* JavaScript unifié.

- ⬢ Inférence du `package.json`
- ⬢ Génération de CJS, ESM et Types
- ⬢ *Bundleless build*
- ⬢ *Passive watcher*

```
{  
  "type": "module",  
  "exports": {  
    ".": {  
      "import": "./dist/index.mjs",  
      "require": "./dist/index.cjs"  
    }  
  },  
  "main": "./dist/index.cjs",  
  "types": "./dist/index.d.ts",  
  "files": ["dist"]  
}
```

```
npx unbuild
```



Magnifique changelog grâce aux commits conventionnels.

- ⚡ Génération du changelog automatisée
- ⚡ Saut de version automatique avec semver
- ⚡ Fonctions exposées pour une utilisation dans d'autres scripts

## Les commits

```
feat: add new feature  
fix: fix a bug  
docs: update documentation
```

## La commande

```
npx changelogen@latest --release
```

## Le résultat

```
🚀 Enhancements  
add new feature (#134)  
📝 Fixes  
fix a bug (#133)  
📘 Docs  
update documentation (#135)  
❤️ Contributors  
Barbabapazines
```



Attribuer des propriétés par défaut de manière récursive.

- ⬢ Léger et rapide
- ⬢ Fusionner avec des fonctions, des tableaux et personnalisable
- ⬢ Types inclus

```
defu({
  vite: {
    plugins: ['custom'],
    ssr: true
  }
}, {
  vite: {
    plugins: ['build-in'],
    outDir: 'dist',
    ssr: false
  }
})
{
  vite: {
    plugins: [ 'custom', 'build-in' ],
    outDir: 'dist',
    ssr: true
  }
}
```

# ⚡ h3

- ⚡ Léger et *tree-shakeable*
- ⚡ Composable et *lazy loaded*
- ⚡ Agnostique du *runtime*

```
export default defineEventHandler(async (event) => {
  assertMethod(event, 'POST')

  const body = await readBody(event)
  const query = getQuery(event)

  if (!body)
    throw createError({ statusCode: 422 })

  if (query)
    return query

  return sendRedirect(event, '/success')
})
```



- Portable et compacts
- Personnalisable et extensible
- KV (unstorage), DB (db0), WebSocket (crossws) intégrés

The screenshot shows the Nitro website homepage. At the top, there's a navigation bar with links for "Guide", "Deploy", "Config", and other icons. Below the header is a large dark banner with abstract red and orange wave patterns. In the center of the banner, the word "Nitro" is written in a large, white, sans-serif font. Underneath "Nitro", the text "Next Generation Server Toolkit" is displayed in a smaller white font. Below the banner, there's a sub-headline "Create web servers with everything you need and deploy them wherever you prefer." followed by three buttons: "Get Started" (pink), "Play Online" (orange), and "View on GitHub" (light blue). At the bottom of the page, the text "A Server for the Modern Web" is centered. There are three callout boxes: "Rapid Development" (with a 🏁 icon), "Deploy Anywhere" (with a 😊 icon), and "Portable and Compact" (with a 📦 icon). Each box contains a brief description of its feature.

[nitro.unjs.io](https://nitro.unjs.io)



# std-env

Utilitaires JavaScript indépendants du runtime.

- ⬢ Détection du *provider* et du *runtime*
- ⬢ CI / TTY / OS / Développement / Version



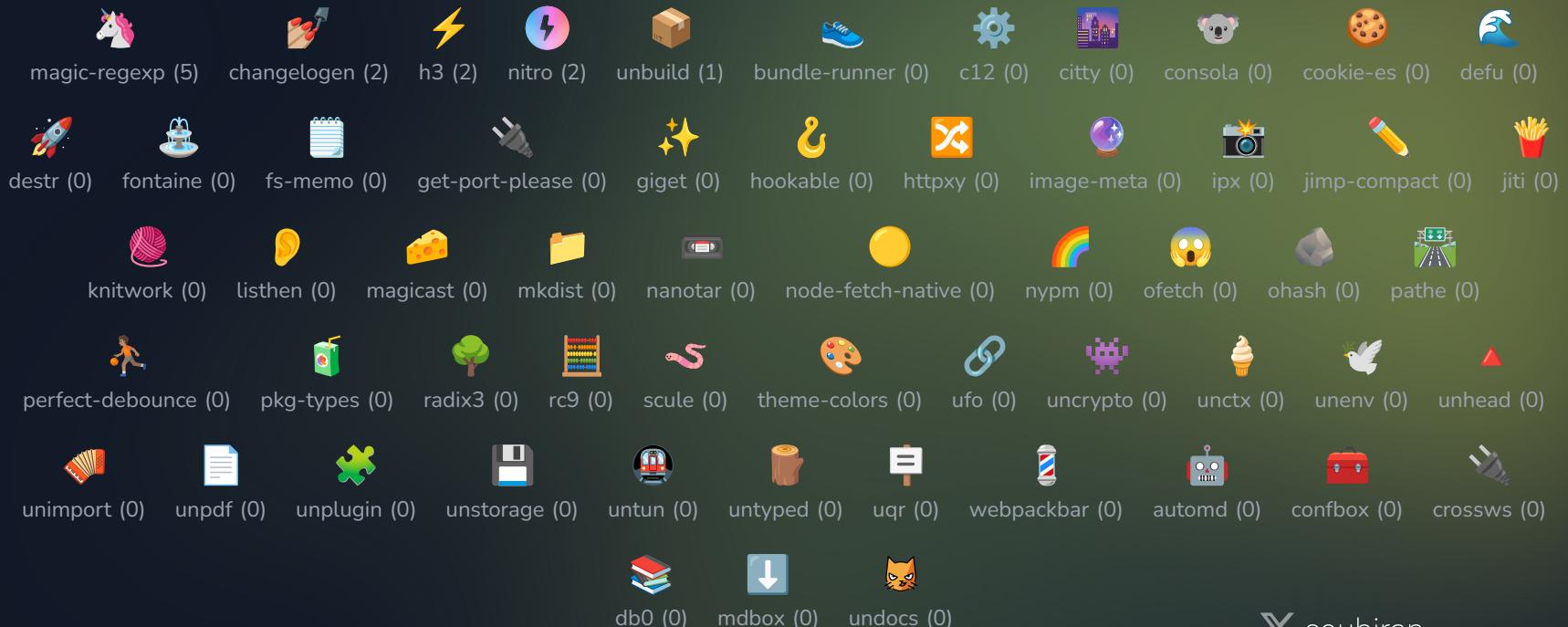
# magic-regexp

Une alternative compilée, sûre et lisible à RegExp.

- ☝ Syntax naturelle et compréhensible
- ☝ Groupes capturés typés automatiquement
- ☝ Compilation en RegExp native

```
// Simple Semver Matcher
createRegExp(
  oneOrMore(digit).groupedAs('major'),
  '.',
  oneOrMore(digit).groupedAs('minor'),
  maybe('. ', oneOrMore(char).groupedAs('patch'))
)
```

# Votre paquet préféré ?



X soubiran\_

🐱 barbapapazes

🔗 esteban-soubiran.site