

文脈の多いこのご時世に
ユーザーフレンドリーなモジュールを
npm-publish

[Kyoto.js 16](#)

このスライドの方針

- 今後を見据えてモジュールを作りたいが
 - がんばりたくもない
 - なんか現状いい感じにしとく方法を模索している

CAVEAT

- ベストプラクティスではない
- いろんな文脈のこまごまとした話
 - しません
- experimental な内容が含まれる
 - なんか間違ってるかも
 - 事情が変わるかも

CAVEAT

- ケース・バイ・ケースなので
 - 諦めるという選択もあり
- 個々のケースの詳細なコード説明
 - あんまりしません

文脈

形式

- CommonJS
- ESModule
- AMD
- UMD
- etc...

バンドラ

- Browserify
- Webpack
- Rollup
- ...etc

拡張子

- .js
- .ts
- .mjs

文脈が多い

誰がどこでどういう風に使うのかが多岐にわたる

提供者として

どう吐き出しておけばよい？

おさらい

package.jsonのフィールド指定

Package.jsonの一部

```
{  
  "typings": "dist/index.d.ts", // 型定義  
  "main": "dist/index.js", // CommonJS  
  "module": "dist/index.es.js", // ESMODULE  
  "browser": "dist/index.umd.js", // UMD  
  "unpkg": "dist/index.umd.js", // unpkg(UMD)  
  "jsdelivr": "dist/index.umd.js", // jsDelivr(UMD)  
  "files": ["dist"], // whitelist形式でnpmに実際に上がるファイルを指定  
  "sideEffects": false // globalに影響するポリフィルのようなものを含む場合はtrue  
}
```

型定義ファイルの指定 - フィールド名は"types"でもOK

なんかいろいろ用意しないといけ
ないっすね

Case.1

『やらない』という施策

index.js から素朴に
`module.exports`や！

アリといえバアリ

失うもの

- TreeShaking
- CodeSplitting
- 人類の時間
- etc...

言われるかもしれない

『Please support TypeScript』

d.ts 作って対応

型定義欲しい気持ちはユーザー目線で見ると
『わかる』

バージョンアップで d.ts 追従させるのしんどい

ユーザーから PR が仮に飛んできてもしんどい

TS で最初から書くかという気持ちになってしまう

TS で書く

c.f.

TypeScript 再入門 — 「がんばらない TypeScript」で、JavaScript を“柔らかい”静的型付き言語に - <https://employment.en-japan.com/engineerhub/entry/2019/04/16/103000>

Case.2

TS で書いて

よっしゃできたぞってなって

どう吐き出すか

なにかしらコンパイルが必要

いろいろ考え出してしまおう

- RollupとかWebpackとか...
 - TS向けのplugin/loader
- Babel7
 - babelどこで動かそう

見返りがあるならガンバリで config 書いてもよい
が...

- 質素なモジュールではしんどい
 - config のメンテな～

TS のコンパイラで素朴に吐く

npm-scriptsに書く

```
"scripts": {  
  "build": "npm run build:esnext && npm run build:esm && npm run build:cjs",  
  "build:esnext": "tsc --module esnext --target esnext  
    --outDir dist/es --project tsconfig.prod.json",  
  "build:esm": "tsc --module esnext --target es5  
    --outDir dist/esm --project tsconfig.prod.json",  
  "build:cjs": "tsc --module commonjs --target es5  
    --outDir dist/cjs --project tsconfig.prod.json"  
}
```

tsconfig.prod.jsonをベースにして変えたいオプションを引数指定する

Case.3

もうちょいなんかラクしっつ
いい感じになって欲しい

ツールに頼る作戦

モジュール吐き出し用のバンドラ いろいろある

いくつか見てよさげなやつを紹介

Microbundle

<https://github.com/developit/microbundle>

- developit氏が主に作っている
- TS対応
- Rollupをラップ
- zero-config
- etc...

Microbundle(v.0.11.0)

```
{
  "source": "src/foo.js",
  "main": "dist/foo.js",
  "module": "dist/foo.mjs",
  "unpkg": "dist/foo.umd.js",
  "scripts": {
    "build": "microbundle",
    "dev": "microbundle watch"
  }
}
```

Bili

<https://github.com/egoist/bili>

- egoist氏が主に作っている
- 別にVue専用ではない
- microbundleに似ているが
 - オプショナル&プラグブル
 - CSSまわりのサポートも厚い

@pika/pack

<https://github.com/pikapkg/pack>

- 全部入り
 - 野望という感じ
- Configが必要
 - 必要なプラグインを`npm install`して使う
- publish用の`package.json`入りのディレクトリを作ってくれる
- publishするためのcliも付いてくる

適材適所・十人十色なので

規模感と自分がどこまでやりたいかによるので
ベストプラクティスはない

終