



fetch-bot

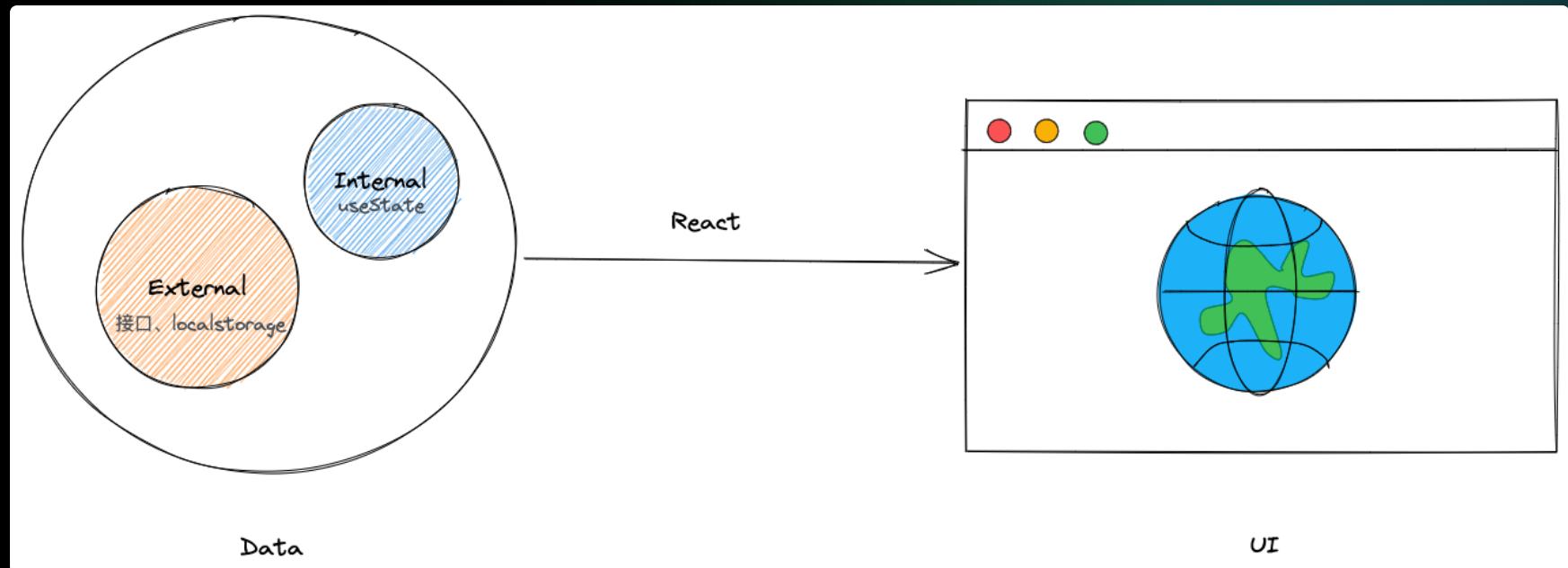
接口请求助手

启封 HAYDEN

Hikari FE
2023-08-31

Why fetch-bot

前端数据可靠性



Features

- 类型友好
- 运行时数据校验 `valibot`
- 自动 Mock
- 无侵入 适配任意请求方法
- 小尺寸 按需加载
- 高度自定义 接口延时，自定义 handler

安装

```
yarn add @velo/fetch-bot valibot  
yarn add @velo/@velo/mock msw -D
```

引入

```
1 // src/util/fetchBot.ts
2 import { initFetchBot } from '@velo/fetch-bot'
3 import * as random from '@velo/mock'
4 import { setupWorker, rest } from 'msw'
5
6 const mswWorker = import.meta.env.PROD ? null : setupWorker()
7 const mswRest = import.meta.env.PROD ? null : rest
8 const mockjsRandom = import.meta.env.PROD ? null : random
9
10 if (mswWorker) mswWorker.start()
11 const fetchBot = initFetchBot({
12   worker: mswWorker,
13   rest: mswRest,
14   random: mockjsRandom,
15 })
16
17 export default fetchBot
```

使用

```
1 // src/services/demo.ts
2 import axios from 'axios'
3 import { number, object, string } from 'valibot'
4
5 import fetchBot from '@util/fetchBot'
6
7 export const getStudentInfo = fetchBot(
8   async (id: number) => {
9     const res = await axios.get(`api/student/${id}`)
10    return res.data
11  },
12  {
13    pathSchema: '/api/student/:id', // 代理接口的路径
14    responseSchema: object({        // 接口返回的数据结构
15      name: string(),
16      age: number(),
17    }),
18  },
19)
```

效果



```
Elements Recorder ▾ Console Sources Network Performance Memory Application Security Lighthouse Components Profiler
Default levels ▾ No issues 2 hidden
top ▾ Filter
script loaded success: http://shared.ydstatic.com/js/ydk/web.jsb-1.7.4.js
scripts load finish!
content script loaded
▶ [MSW] Mocking enabled.
res = ➤ {name: '贾杰', age: 49}
▶ [MSW] 17:00:25 GET /api/student/1 (200 OK)
res = ➤ {name: '阎秀兰', age: 21}
```

yd़-1.7.4.js:2
yd़-1.7.4.js:2
index.js:1
printStartMessage.ts:20
App.tsx:16
RestHandler.ts:183
App.tsx:16

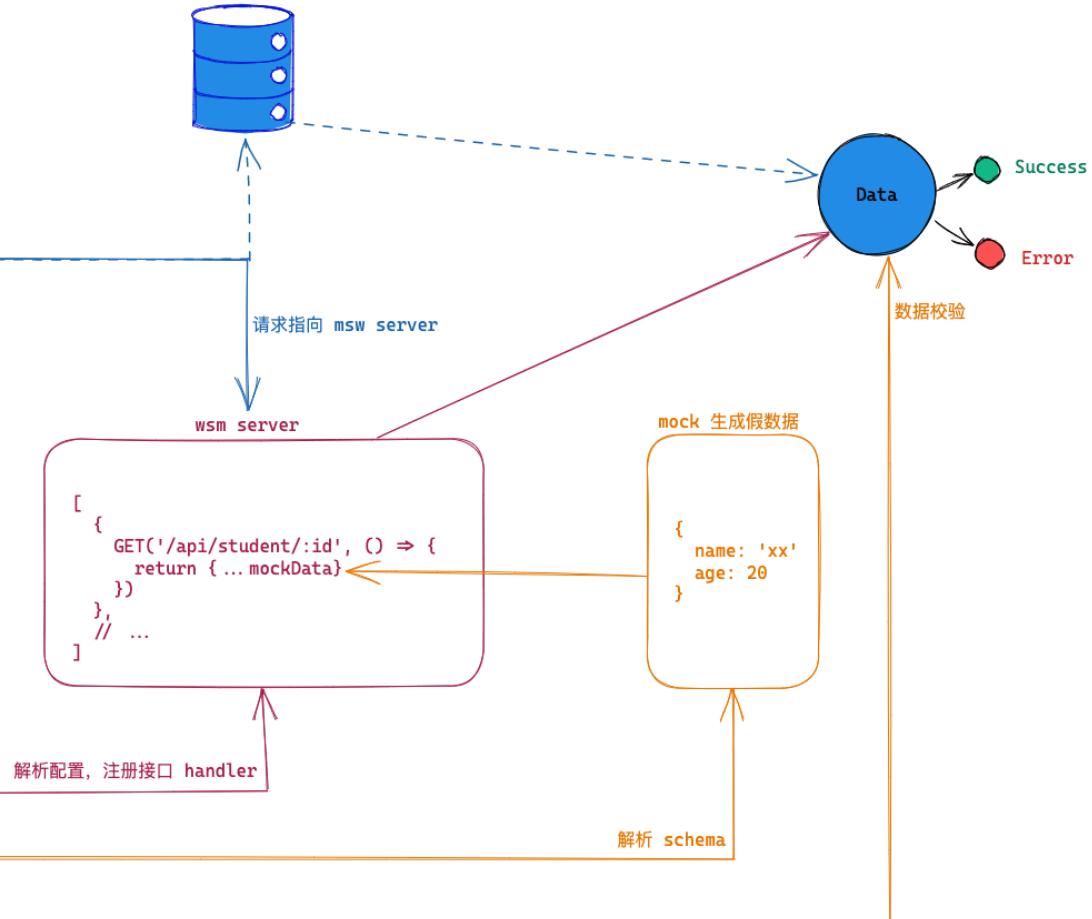
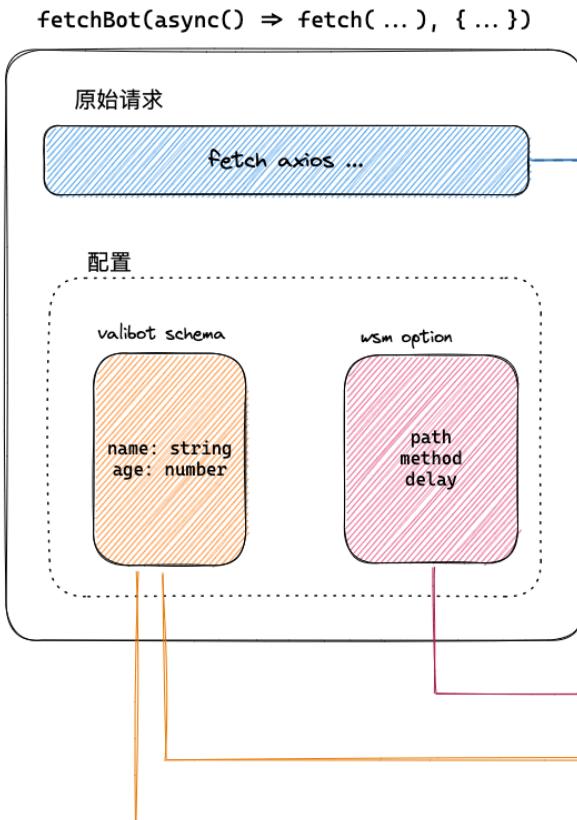
自定义

- `overrides` 覆盖指定部分的返回值
- `delay` 指定延迟时间
- `action` 完全自定义接口的动作
- `ignoreValidation` 是否忽略接口的返回值校验结果

可能性

- 提升项目可读性
 - 全接口 Mock，随时随地能跑起来项目
 - 随意修改想要的数据，方便调试
- 提升 Mock 系统维护体验
 - 物理位置上将 Mock 与接口请求代码放在一起
 - 降低 Mock 系统维护成本
- 线上问题排查
 - 自动校验数据，打印错误日志

如何实现



问题

部分代码**需要**打包 部分代码**不需要**打包

- valibot
- msw
- mockjs

Tree shaking 友好

- 支持 ES Module
- 独立导出 对比 zod 与 valibot
- sideEffect 管理

ES6 modules go in...

Select an example... ▾

Start over

main.js

(entry module)

```
1 // TREE-SHAKING
2 import { faker } from './test.js';
3
4 if (false) {
5   const myFaker = faker
6   console.log(myFaker)
7 }
```

maths.js

```
1 import {} from './test.js'
```

test.js

```
1 class Faker {
2   constructor() {}
3   method() {
4     window.a = 1
5   }
6 }
7
8 export const faker = new Faker()
```

+ add module

... bundle comes out

⚠ Rollup completed with warnings:

Generated an empty chunk: "main".

output.format

es

amd

cjs

iife

umd

system

treeshake

false

true

smallest

safest

recommended

add option +

Rollup 3.28.0

1
2

Solution

- 迁移改造 `@velo/mock`
- 将依赖从 fetch-bot 内置改为使用者传入

Roadmap

- yapi-bee 一键生成 fetch-bot 需要的参数 valibot schema, pathSchema
- 在网页中注入调试面板，随时修改接口数据
- 线上接口数据异常时告警？

谢谢！