

h5开源组件库技术方案

需求背景

本次需求是基于小程序组件库mp-devkit的改造，使组件库适配h5

组件准备工作

基于monorepo开发，打包成适用小程序的mp-devkit和适用于h5的h-devkit。也可以开两个项目，分别打包mp-devkit和h5-devkit。

组件打包名称@h/mp-devkit,@h/h5-devkit,感觉devkit，有点像开发工具的名称了，可以考虑其它名称

eslint + prettier + husky + commitlint

帅哥的fe-lint是否开源，commitlint是否按照fe-commitlint规范来，还是使用通用的commitlint

@h/fe-commitlint，@h/fe-lint目前只能在zaihui的npm已经登录下使用，考虑将配置直接写入代码

Px适配（这个只在实际项目中使用，在本次组件开发只用于test文件夹）

使用postcss-pxtorem，和类似于h-flexiable的函数库

h-flexiable目前只能在zaihui的npm已经登录下使用，考虑将配置直接写入代码

单元测试（UT）

- **mocha**：Mocha 是一个功能丰富的 JavaScript 测试框架；
- **jest**：facebook 出的一款 JavaScript 自动化测试框架，专注于简单性。
- **vitest**：尤大团队打造，一个由 Vite 打造的单元测试框架，而且它很快！

本次h5组件开发使用 **vitest** //之前项目使用jest，也可以使用jest开发

组件文档（这个优先级低可以放到后面做）

使用组件文档工具，vitepress。//mp-devkit之前用的是vuepress，差异较小，也可以考虑直接用

组件文档部署

目前考虑部署到github上，因为开源，或者部署到公司服务器上

组件文档参考：<https://encodedecod.github.io/h-devkit/>

打包

使用tsup打包

优点：

- 支持生成d.ts文件
- 底层实现基于rollup，esbuild,更快的打包速度

评论（6）

>>

基于monorepo开发，打包成适用小程序…



杨海啸 2022年12月9日

mp-devkit 和 h5-devkit 吧



杨海啸 2022年12月15日

用 pnpm-workspace 还是 yarn-workspace，我看到目录结构还有 yarn.lock 文件。



秦春祥 2022年12月15日

用的 pnpm-worksapce

基于monorepo开发，打包成适用小程序…



杨海啸 2022年12月9日

menorepo 不影响两个项目的打包，h5 打包肯定要重新配，mp 的就按旧配置打包就行。

postcss-pxtorem，h-flexiable是否开源



杨海啸 2022年12月9日

h-flexiable 是开源的

使用组件文档工具，vitepress。



杨海啸 2022年12月15日（编辑过）

这点感觉根据时间决定吧，优先级低一些

目前考虑部署到github上，因为开源，或…



杨海啸 2022年12月13日（编辑过）

这个不需要部署吧，用 npm 发包

打包成两个文件夹：

- `esm` 格式：ECMAScript Module，现在使用的模块方案，使用 `import` `export` 来管理依赖；
- `cjs` 格式：CommonJS，只能在 NodeJS 上运行，使用 `require("module")` 读取并加载模块；

发布

使用分包发布工具 `changesets`

这里有个疑问，发布npm包的时候使用公司的npm账号还是个人的

开源h5 & 打包说明

方案说明

方案一：

直接基于以前的mp-devkit项目开发，直接先将mp-devkit的组件（标签名，属性，事件）映射为h5组件。

映射说明：<https://docs.taro.zone/blog/2021-04-22-Taro-3.3-alpha>

标签映射：<https://github.com/NervJS/taro-rfcs/blob/html-support/rfcs/0004-rendering-html.md%E4%B8%80%E6%A0%87%E7%AD%BE%E5%90%8D%E5%BD%B1%E5%B0%84>

我们会将mp-devkit中可以全部实现映射的组件改为h5适用形式（标签名，属性，事件）并且放入common文件夹。

common打包将生成h5组件，taro使用时需要以下工作：

- 保证项目的taro版本在3.3.0以上
- 需要通过安装同构插件的方式使用，在`config`的`index.js`需要以下配置：

```
1 // config/index.js
2 const config = {
3   // ...
4   plugins: ['@tarojs/plugin-html'],
5 }
```

标签映射附录：

HTML 标签	小程序组件
块级标签	view
内联标签	text
span	view
img	image
a	navigator
a["href=java script;"]	view
input	input
input["type=checkbox"]	checkbox
input["type=radio"]	radio
button	button
textarea	textarea
progress	progress
label	label
form	form
audio	audio
canvas	canvas
video	video
iframe	web-view
slot	slot

属性映射：

<a> 属性	<Navigator> 属性
href	url
target	openType

target 属性值：

a[target]	Navigator[open-type]
_blank	navigate
_self	redirect

<input>

<input> 属性	<Input> 属性
autofocus	focus
readonly	disabled

type 属性值：

input[type]	Input[type]
tel	number
input[type=password]	input[password=true]

<textarea>

| <textarea> 属性 | <Textarea> 属性 |
|---------------|---------------|
| autofocus | focus |
| readonly | disabled |

<progress>

| <progress> 属性 | <Progress> 属性 |
|-------------------|---------------|
| value / max * 100 | precent |

<button>

| button[type] | Button[form-type] |
|---------------------|--------------------------|
| button[type=submit] | Button[form-type=submit] |
| button[type=reset] | Button[form-type=reset] |

事件映射：

| HTML 事件 | 小程序组件事件 |
|---------|---------|
| click | tap |

<input type='checkbox'>

| input[type=checkbox] | <Checkbox> |
|----------------------|------------|
| onChange | bindtap |

`<input type='radio'>`

| | |
|-------------------|---------|
| input[type=radio] | <Radio> |
| onChange | bindtap |

`<input>`

| | |
|----------|-----------|
| <input> | <Input> |
| onChange | onInput |
| keypress | onConfirm |

方案二：

重新构造组件，这个可以将无法完全适配同构的mp-devkit组件全部迁移。但是时间较长，可以在以后新建h5目录，实现组件的全部迁移。

开源h5

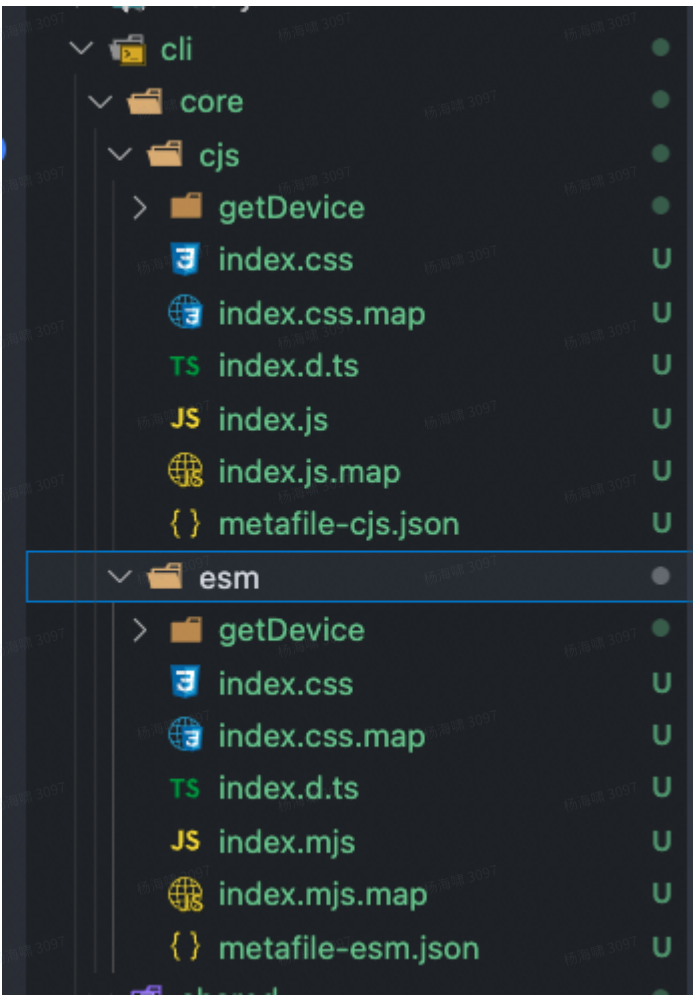
目前经过内部讨论，由于之前的组件库使用到taro开发，考虑组件开发的人力资源问题，暂时先使用方案一。

打包

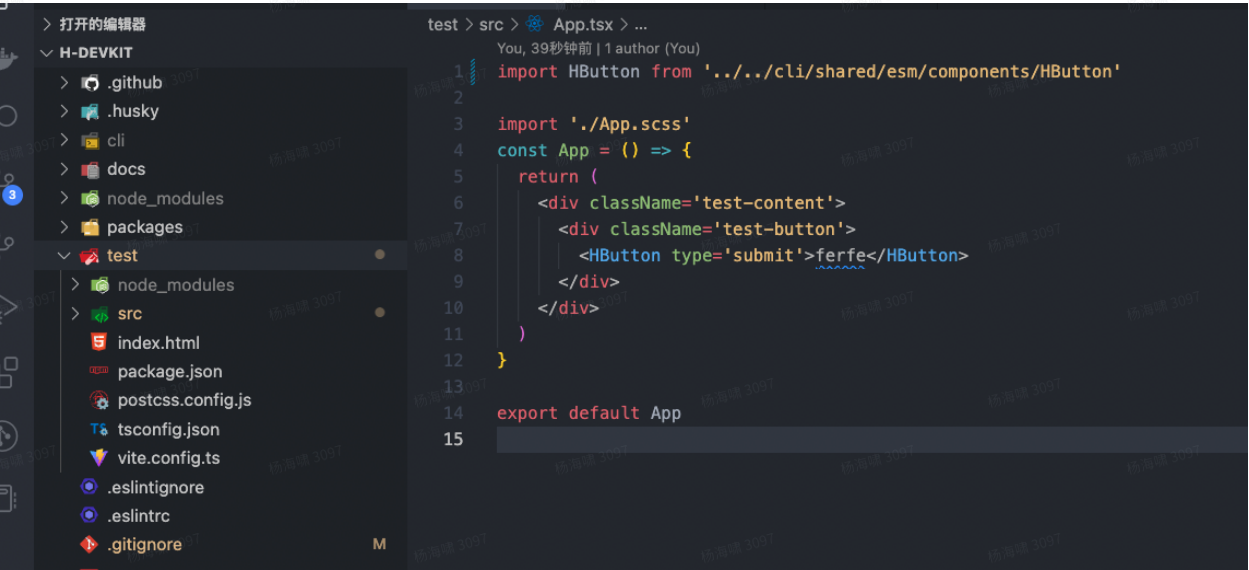
mp-devkit目前使用tsc打包，由于打包的多包话和分包话，考虑使用tsup做打包处理。避免打包无用文件的同时，利用esbuild提高打包效率，并且自动清除打包目录，压缩，添加d.ts文件等一系列能力。

针对h5和小程序分别打包出适应es module 和CommonJS的打包文件。

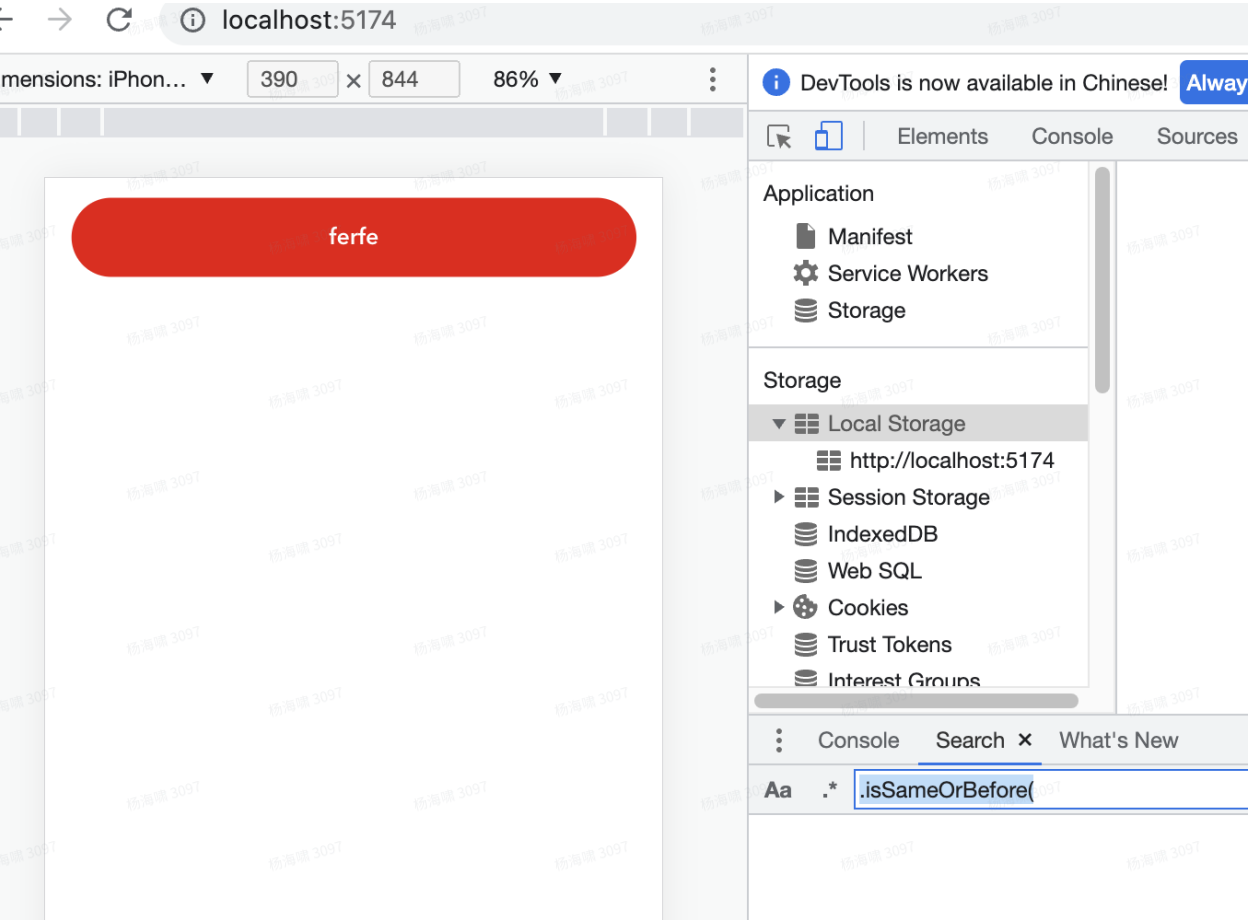
实际打包效果：



组件引用：



效果：



组件目录



杨海啸 2022年12月15日

组件库，打包目录用 lib 比 dist 更好，打包后目录结构还要确认下

组件目录

- 1 |— .github // github,cicd相关
- 2 |— test // 测试文件夹
- 3 |— src
- 4 |— README.md
- 5 |— index.html
- 6 |— package.json
- 7 |— tsconfig.json
- 8 |— tsconfig.node.json
- 9 |— vite.config.ts
- 10 |— yarn.lock
- 11 |— docs // 文档
- 12 |— .vitepress
- 13 |— h5-devkit //h5文档路径
- 14 |— mp-devkit //小程序文档路径
- 15 |— index.md
- 16 |— shared // 共享文件夹
- 17 |— styles // 共有的样式
- 18 |— configs // 共用的配置
- 19 |— utils // 共有的方法
- 20 |— lib // 打包文件夹
- 21 |— h5 //用于h5的组件
- 22 |— esm //esm格式打包文件
- 23 |— cjs //cjs格式打包文件
- 24 |— mp //小程序组件
- 25 |— esm //esm格式打包文件
- 26 |— cjs //cjs格式打包文件
- 27 |— packages
- 28 | |— common //mp-devkit中可以替换为h5标签的组件
- 29 | | |— 公有组件
- 30 | | |— index.ts
- 31 | |— mp-devkit //这个文件可以考虑不变
- 32 | | |— 小程序组件
- 33 | | |— index.ts
- 34 |— README.md
- 35 |— package.json
- 36 |— tsconfig.json
- 37 |— .eslintrc
- 38 |— .prettierrc
- 39 |— commitlint.config.ts
- 40 |— stylelint.config.js
- 41 |— vitest.config.ts
- 42 |— tsconfig.node.json
- 43 |— tsup.config.ts //tsup配置
- 44 |— yarn.lock

项目拆分







组件目前现状

已经本地搭好基础的架子：<https://github.com/encodedecod/h-devkit>

组件文档参考：<https://encodedecod.github.io/h-devkit/>

目前需要讨论组件打包方案，小程序需要引用过来，还是只打包当前的h5库，taro打包引用过来可能会导致打包文件过来，tsup打包。

分工以及估时

- @h/fe-lint, @h/fe-commitlint, h-flexiable配置手动写入 1d  秦春祥
- mp-devkit引入 1d  杨海啸
- h5组件开发，目前是将可以切换h5标签的taro组件直接切换h5标签  秦春祥
 杨海啸 1d以上
- 组件npm发包（使用公司npm账号） 2h

tsup文档：<https://tsup.egoist.dev/#conditional-config>

参考文档：

非大厂的我们，如何去搞Vue/React Hooks和Utils的企业开源工具库

每个前端都值得拥有自己的组件库，就像每个夏天都拥有西瓜🍉

必知必会tsup打包库原理分析