用比较性思维谈工程化工具在Vue.js的优雅设计

摩拜前端负责人 - 小春

Riot

React

Angular

Q: Vue.js比jQuery好学吗

简洁与简单

jQuery

```
<div id="output"></div>
<button id="increment">Increment</button>


var counter = 0;
$(document).ready(function() {
  var $output = $('#output');
  $('#increment').click(function() {
    counter++;
    $output.html(counter);
  });
  $output.html(counter);
});
```

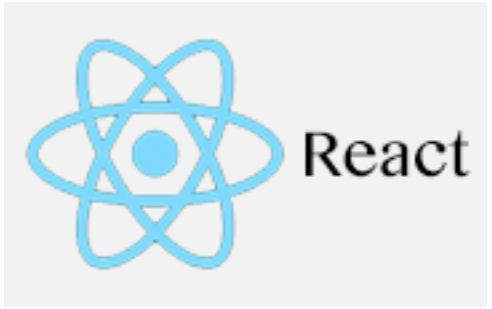
Vue.js

```
<div id="app">
  <div>{{ counter }}</div>
  <button v-on:click="increment">Increment</button>
</div>

new Vue({
  el: '#app',
  data: {     counter: 0    },
  methods: {
    increment() {
      this.counter++;
    }
  }
});
```

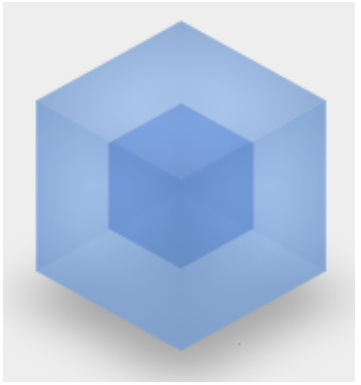
大纲

- 脚手架工具
  - vue-cli探究
  - 其他cli对比
- 基于Webpack的构建体系：
  - 本地开发
  - 测试环境部署
  - 发布上线
  - 优化工具
- 复杂业务系统
- Vue组件编译
- Rollup编译vue lib
- Babel如何在vue中应用
- ESLint如何在vue中应用
- Yarn如何在vue中应用
- 一些工具编写的收获

alleviate the amount of copy & pasting you do between projects



. . .



Pick the right tool for the job

工具复杂度是为了处理内在复杂度所做的投资 - 尤雨溪

《Vue 2.0，渐进式前端解决方案》

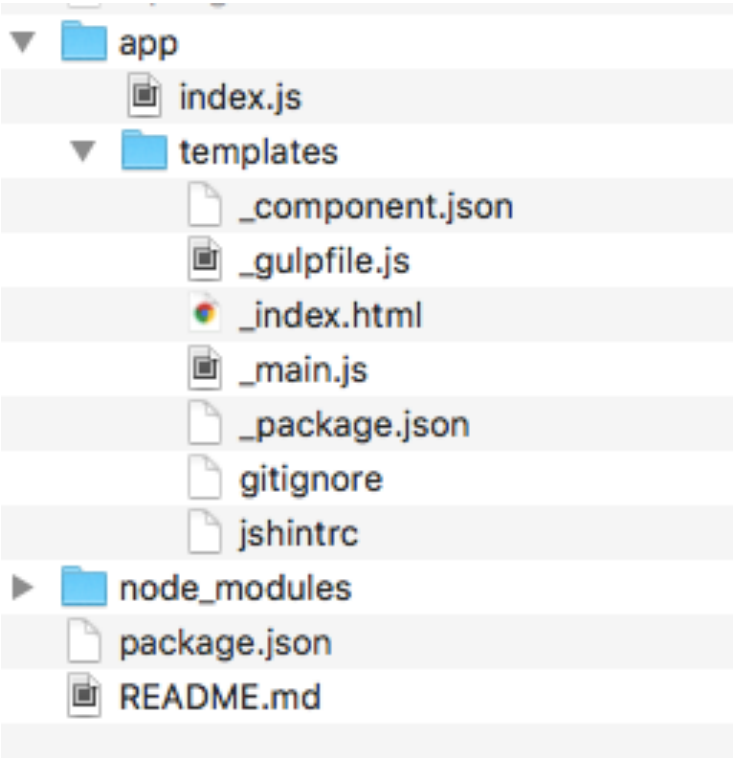
yo

最早的脚手架工具，非常简单的 API

generator-vue

```
➜  vue-conf  yo -h
Usage: yo GENERATOR [args] [options]

General options:
  -h, --help     # Print generator's options and usage
  -f, --force    # Overwrite files that already exist

Please choose a generator below.
```

```
➜  vue-conf  yo
cmd undefined
[?] What would you like to do?
  Run the Gulp-webapp generator (0.1.0)
  Run the Polymer generator (0.6.3)
  Run the React generator (0.0.1)
› Run the Vue generator (0.0.2)
  Run the Wdj generator (0.4.4)
  Run the Webapp generator (0.4.2)
  Run the Yc generator (0.0.1)
```



vue-cli

```
vue-conf  vue help

Usage: vue <command> [options]


Commands:

  init        generate a new project from a template
  list        list available official templates
  help [cmd]  display help for [cmd]

Options:

  -h, --help     output usage information
  -V, --version  output the version number
```

官方出品的命令行脚手架工具，支持：

1、vue init 按照指定模板，在指定目录生成项目结构

2、vue list 列出线上 vuejs-templates repo 支持的模板列表

vue-cli

目录结构：

bin
- vue
- vue-init
- vue-list

lib
- ask.js
- check-version.js
- eval.js
- filter.js
- generate.js
- git-user.js
- logger.js
- options.js

package.json 中配置：

```
"bin": {
 "vue": "bin/vue"
}
```

文件头：

```
#!/usr/bin/env node
```

```
→  vue-conf  vue init webpack pro

[ 'webpack', 'pro' ]
A newer version of vue-cli is available.

latest:    2.8.2
installed: 2.1.0
```

vue init

支持 3 种方式的模板初始化：

1、官方

2、本地

3、线上其他repo

目录结构：

vue-conf
build
  - build.js
  - check-version.js
  - dev-client.js
  - utils.js
  - vue-loader.conf.js
  - webpack.base.conf.js
  - webpack.dev.conf.js
  - webpack.prod.conf
config
  - dev.env.js
  - index.js
  - prod.env.js
src
static
.babelrc
…

目前官方模板

https://github.com/vuejs-templates

- webpack
- webpack-simple
- browserify
- browserify-simple
- simple

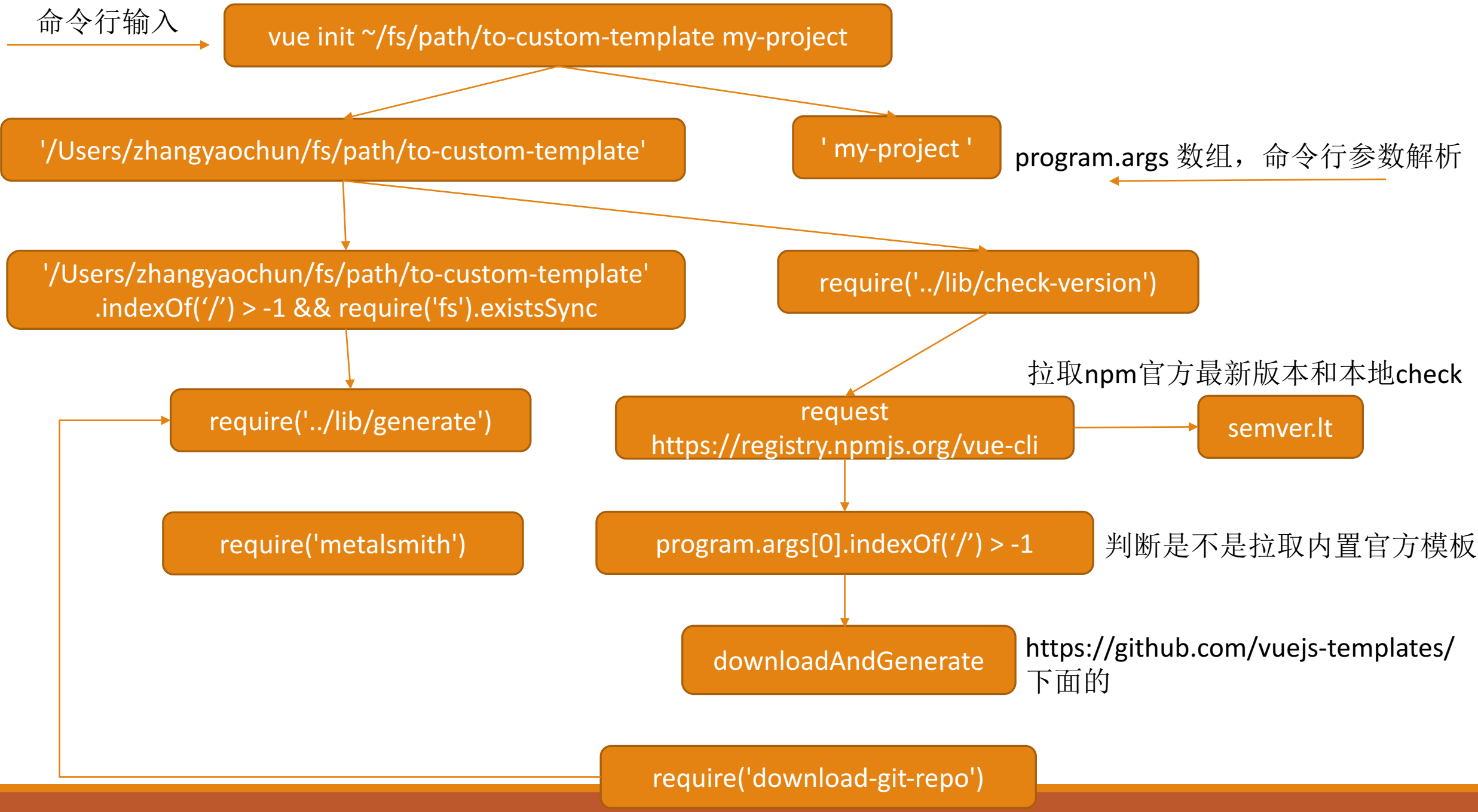
目前使用概率最高的

在线模板的问题：

比如我在家，没网，想创建一个项目：

---- sorry~

```
→  vue-conf  vue init webpack abc

[ 'webpack', 'abc' ]
   vue-cli · getaddrinfo ENOTFOUND api.github.com api.github.com:443
```


命令行输入
vue init ~/fs/path/to-custom-template my-project
'/Users/zhangyaochun/fs/path/to-custom-template'
' my-project '
program.args 数组，命令行参数解析
'/Users/zhangyaochun/fs/path/to-custom-template'
.indexOf(‘/’) > -1 && require('fs').existsSync
require('../lib/check-version')
拉取npm官方最新版本和本地check
require('../lib/generate')
request
https://registry.npmjs.org/vue-cli
semver.lt
require('metalsmith')
program.args[0].indexOf(‘/’) > -1
判断是不是拉取内置官方模板
downloadAndGenerate
https://github.com/vuejs-templates/
下面的
require('download-git-repo')

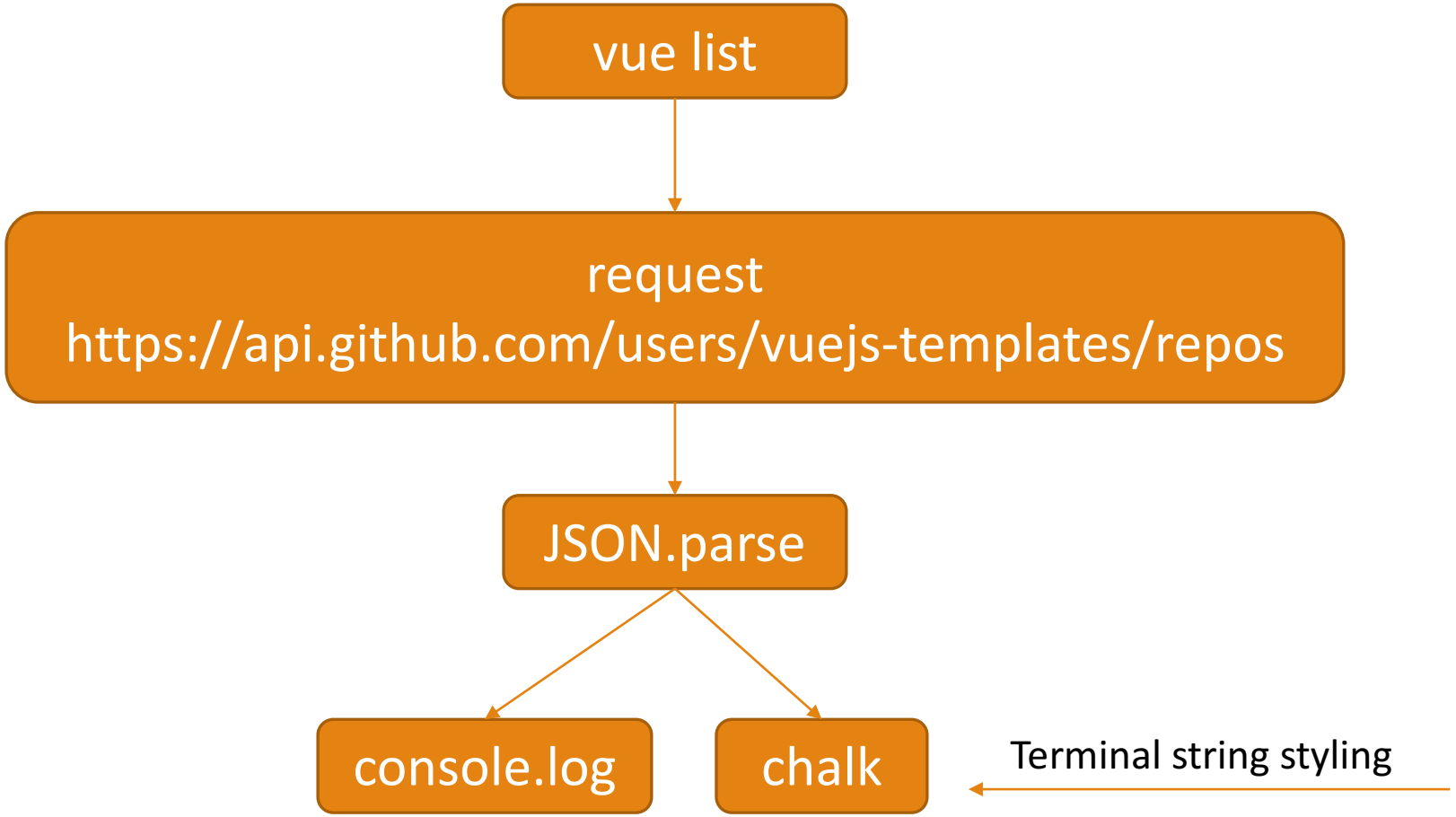
vue list

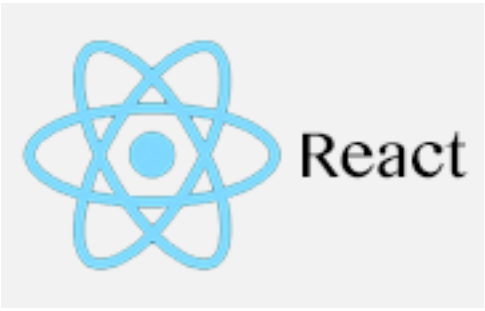
从远程拉取官方模板列表数据，然后在命令行展示

vue list

```
➜  ~  vue list

request debugger by zyc for vue-conf
  Available official templates:

  ★  browserify - A full-featured Browserify + vueify setup with hot-reload, linting & unit testing.
  ★  browserify-simple - A simple Browserify + vueify setup for quick prototyping.
  ★  pwa - PWA template for vue-cli based on the webpack template
  ★  simple - The simplest possible Vue setup in a single HTML file
  ★  webpack - A full-featured Webpack + vue-loader setup with hot reload, linting, testing & css extrac
tion.
  ★  webpack-simple - A simple Webpack + vue-loader setup for quick prototyping.
```

vue list


vue list
request
https://api.github.com/users/vuejs-templates/repos
JSON.parse
console.log
chalk
Terminal string styling



create-react-app

React 脚手架命令行工具：
源码就一个 index.js 文件，528 行 – 39 行顶部注释
依赖 react-scripts

create-react-app

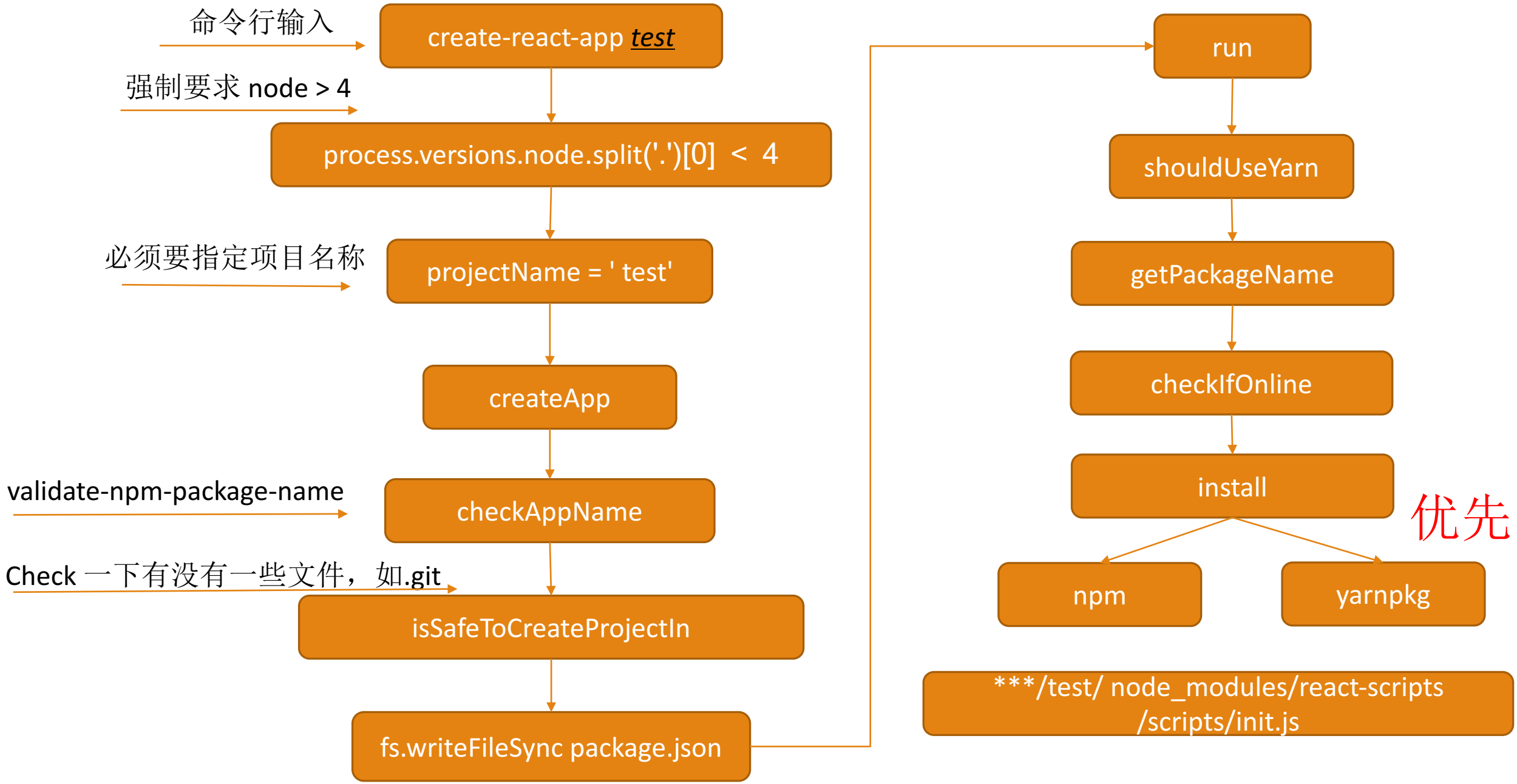
```
→  vue-conf  create-react-app -h

  Usage: create-react-app <project-directory> [options]

  Options:

    -h, --help                                output usage information
    -V, --version                             output the version number
    --verbose                                 print additional logs
    --scripts-version <alternative-package>   use a non-standard version of react
-scripts

    Only <project-directory> is required.

    A custom --scripts-version can be one of:
      - a specific npm version: 0.8.2
      - a custom fork published on npm: my-react-scripts
      - a .tgz archive: https://mysite.com/my-react-scripts-0.8.2.tgz
    It is not needed unless you specifically want to use a fork.

    If you have any problems, do not hesitate to file an issue:
      https://github.com/facebookincubator/create-react-app/issues/new
```

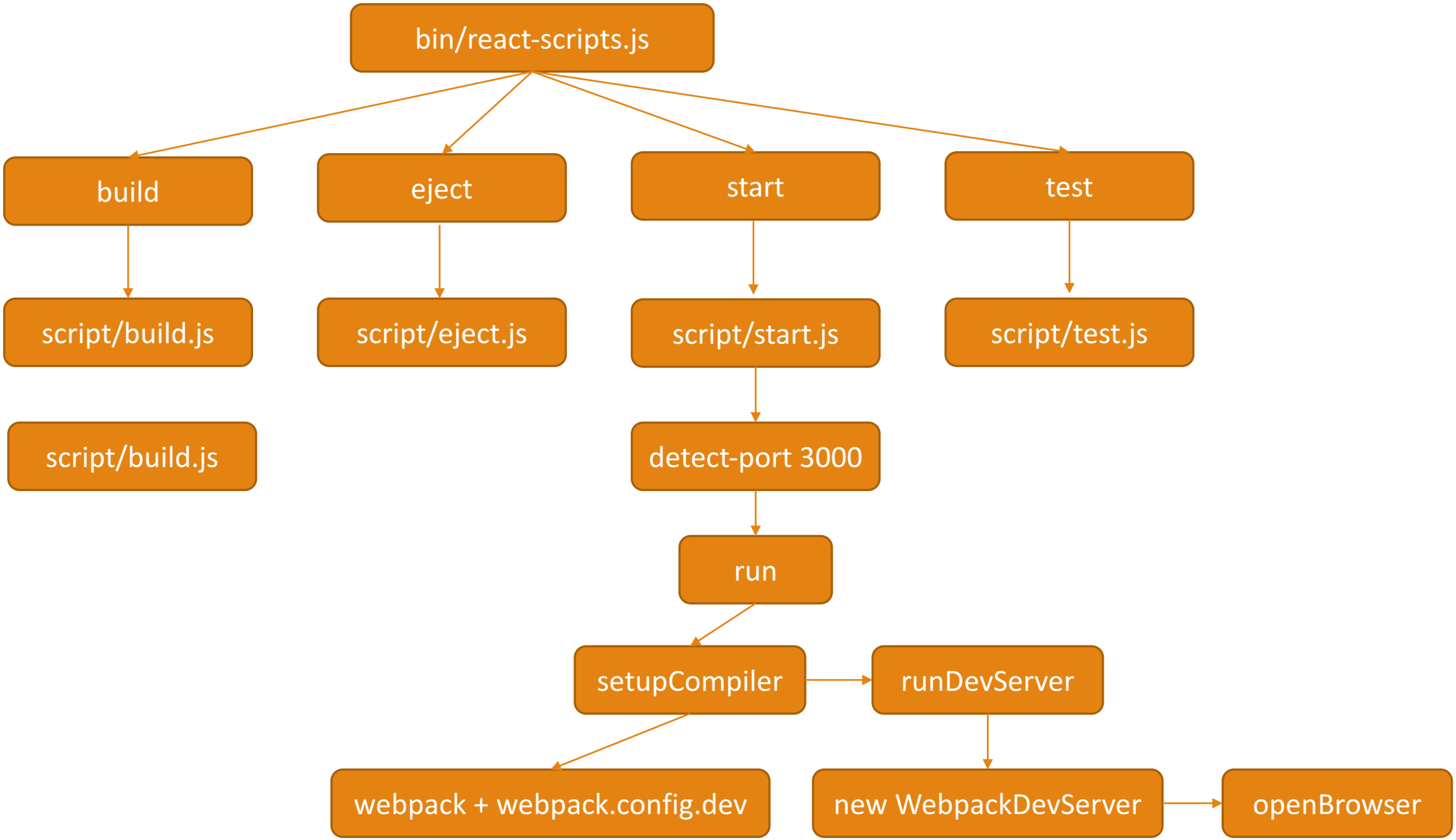
目录结构：

public
- favicon.ico
- index.html

src
- App.css
- App.js
- App.test.js
- index.css
- index.js
- logo.svg

node_modules
package.json
README.md


命令行输入
create-react-app test
强制要求 node > 4
process.versions.node.split('.')[0] < 4
必须要指定项目名称
projectName = ' test'
createApp
validate-npm-package-name
checkAppName
Check 一下有没有一些文件，如.git
isSafeToCreateProjectIn
fs.writeFileSync package.json
run
shouldUseYarn
getPackageName
checkIfOnline
install
优先
npm
yarnpkg
***/test/ node_modules/react-scripts /scripts/init.js


bin/react-scripts.js
build
eject
start
test
script/build.js
script/eject.js
script/start.js
script/test.js
script/build.js
detect-port 3000
run
setupCompiler
runDevServer
webpack + webpack.config.dev
new WebpackDevServer
openBrowser

create-react-app VS vue-cli

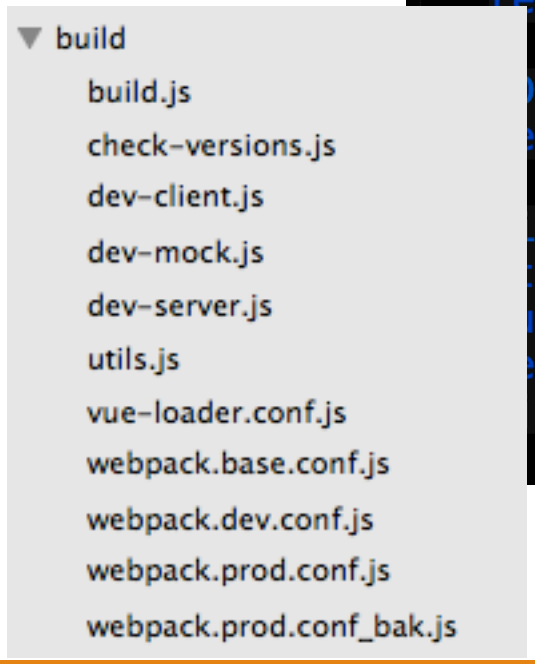
都是基于 webpack 构建

create-react-app：更多封装在了react-scripts 里面，同时把它加到了初始项目的依赖里面：

```
"dependencies": {
  "react": "^15.5.4",
  "react-dom": "^15.5.4"

Dependencies": {
eact-scripts": "0.9.5"

ipts": {
tart": "react-scripts start",
uild": "react-scripts build",
est": "react-scripts test --env=jsdom",
ject": "react-scripts eject"
```

vue-cli：

更为直接，把脚本模板都直接放到 build 文件夹中，开发者更容易修改



angular-cli

从 2 版本开始才有的，依赖 ember-cli

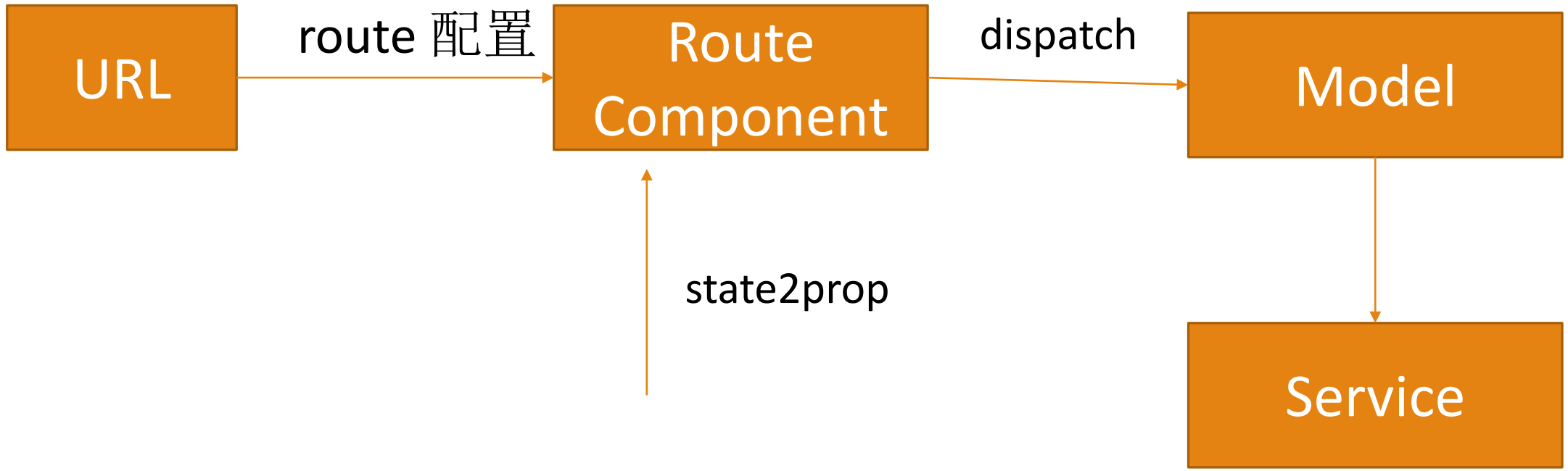
处理复杂的业务系统呢？

文件夹多，直接的各种关系
主动实例化

dva-cli + dora

dora-plugin-proxy
dora-plugin-webpack
dora-plugin-webpack-hmr

vue-cli + vuex + vue-route



这些脚手架的差异点

工具名称	命令行工具	模板	拷贝文件方式	依赖安装	Npm 还是 yarn
vue-cli	commander	远程：官方、用户自定义 本地	metalsmith	不自动，不可配置	
create-react-app	commander	本地，但是在依赖的 react-scripts/template	react-scripts /scripts/init.js	自动，不可配置	优先yarn
dva-cli	commander	本地	dva-ast | boilerplates	自动	
yo	nopt	本地	yeoman-generator	可以配置	
angular-cli	ember-cli	本地			

webpack

和 Vue.js 结合最紧密的构建工具

配置分离

采用 webpack-merge 和 webpack.base.conf.js + webpack.prod.conf.js

Npm script

webpack-merge & process.env.npm_lifecycle_event

//早期

```
var path = require('path');
var merge = require('webpack-merge');
var TARGET = process.env.npm_lifecycle_event;
var common = {
  entry: path.join(__dirname, 'app'),
  ...  module: {    loaders: [] }
};

 if(TARGET === 'build') {
    module.exports = merge(common, {    ...  });
}

...
```

webpack.base.conf.js

常规的配置，比如 output

webpack.prod.conf.js 文件内容：

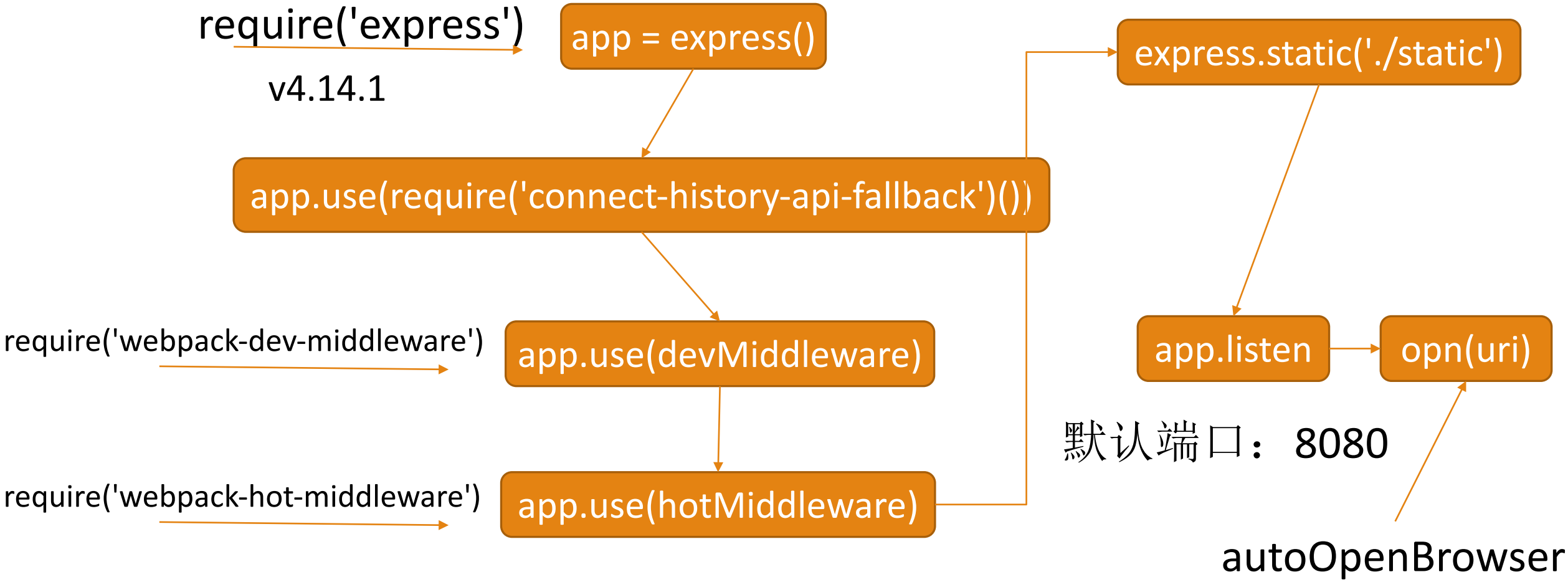
```
var merge = require('webpack-merge')
var baseWebpackConfig = require('./webpack.base.conf')

var webpackConfig = merge(baseWebpackConfig, {
})

module.exports = webpackConfig
```

本地开发

Express + 多个中间件：

- ✓ connect-history-api-fallback
- ✓ webpack-hot-middleware
- ✓ http-proxy-middleware
- ✓ express.static

dev-server


require('express')
v4.14.1
app = express()
express.static('./static')
app.use(require('connect-history-api-fallback')())
require('webpack-dev-middleware')
app.use(devMiddleware)
app.listen
opn(uri)
默认端口：8080
require('webpack-hot-middleware')
app.use(hotMiddleware)
autoOpenBrowser

Webpack插件

DefinePlugin

动态的注入一些变量，比如一些版本、环境信息等

```
new webpack.DefinePlugin({
    'process.env': env
}),
```

friendly-errors-webpack-plugin

recognizes certain classes of webpackerrors and cleans,
aggregates and prioritizes them to provide a betterDeveloper Experience

vue-devtools

```
"name": "Vue.js devtools",
"version": "3.1.2",
"description": "Chrome devtools extension for debugging Vue.js applications.",
"manifest_version": 2,
"icons": {
   "16": "icons/16.png",
   “48”: “icons/48.png”,
   "128": "icons/128.png”
},
"browser_action": {}
"devtools_page": "devtools-background.html",
"background": {}
```

```
"content_security_policy": "script-src 'self' 'unsafe-eval'; object-src 'self'",
"web_accessible_resources": [ "devtools.html", "devtools-background.html", "build/backend.js" ],

"devtools_page": "devtools-background.html",

"background": {
  "scripts": [ "build/background.js" ],
  "persistent": false
},

"permissions": [
  "http://*/*",
  "https://*/*",
  "file://*"
],

"content_scripts": [
  {
    "matches": ["<all_urls>"],
    "js": ["build/hook.js"],
    "run_at": "document_start"
  },
  {
    "matches": ["<all_urls>"],
    "js": ["build/detector.js"],
    "run_at": "document_idle"
  }
]
```

上传测试环境

webpack-sftp-client

开发过程中把本地资源 push 到开发机

webpack-sftp-client 如何使用

```
var WebpackSftpClient = require('webpack-sftp-client')

plugins: [
    new WebpackSftpClient({
      port:  '22',
      host:  '***',
      path: './dist/',
      remotePath: ' '
    })
]
```

没有用户名密码？推荐免登陆

上传多台咋办？

webpack-sftp-client

```
var client = require('scp2');  // 核心还是依赖 scp2 这个包

function WebpackSftpClient(options) {
  this.options = options;
}

WebpackSftpClient.prototype.apply = function(compiler) {
  compiler.plugin('after-emit', function(compilation) {
    // 读取配置，调用 client.scp
  }
}
```

打包上线

webpack.optimize.UglifyJsPlugin
extract-text-webpack-plugin
optimize-css-assets-webpack-plugin

打包上线 – 低频

```
if (config.build.productionGzip) {
  var CompressionWebpackPlugin = require('compression-webpack-plugin')
  webpackConfig.plugins.push(
    new CompressionWebpackPlugin({
      asset: '[path].gz[query]',
      algorithm: 'gzip',
      test: new RegExp('\\.(' + config.build.productionGzipExtensions.join('|') +
')$'),
      threshold: 10240,
      minRatio: 0.8
    })
)}
```

默认 false

html-webpack-inline-source-plugin

FIS 应用最频繁的 __inline

```
new HtmlWebpackPlugin({
  filename: config.build.index,
  template: 'index.html',
  inlineSource: 'manifest',
  inject: true,
  minify: {
  },
  // necessary to consistently
  chunksSortMode: 'dependency'
}),
```

```
new HtmlWebpackInlineSourcePlugin()
```

html-webpack-inline-source-plugin

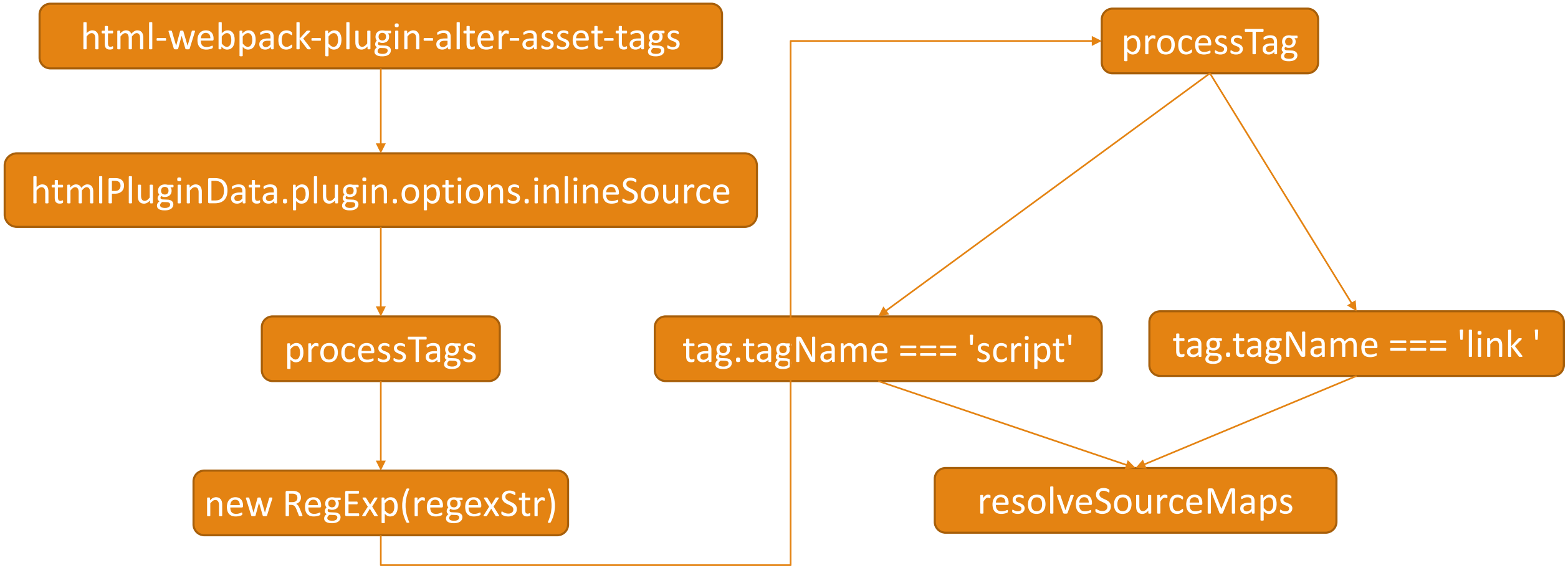
应用示例：

```
<script type=text/javascript src=/static/js/manifest.067cf2fb4cdf4b55947f.
js></script>
<script type=text/javascript src=/static/js/vendor.db4d69a89d75cecaafa2.js>
</script>
<script type=text/javascript src=/static/js/app.fb9e1e9126814c6e1495.js><
/script>
```

vendor 和 app 打包之后都强依赖 manifest 文件的： webpackJsonp()

如果是 CDN 文件，时序问题随即而来

html-webpack-inline-source-plugin



CommonsChunkPlugin

```
<script type=text/javascript src=/static/js/manifest.067cf2fb4cdf4b55947f.
js></script>
<script type=text/javascript src=/static/js/vendor.db4d69a89d75cecaafa2.js>
</script>
<script type=text/javascript src=/static/js/app.fb9e1e9126814c6e1495.js><
/script>
```

Entry Chunk：入口代码块包含了 webpack 运行时需要的一些函数

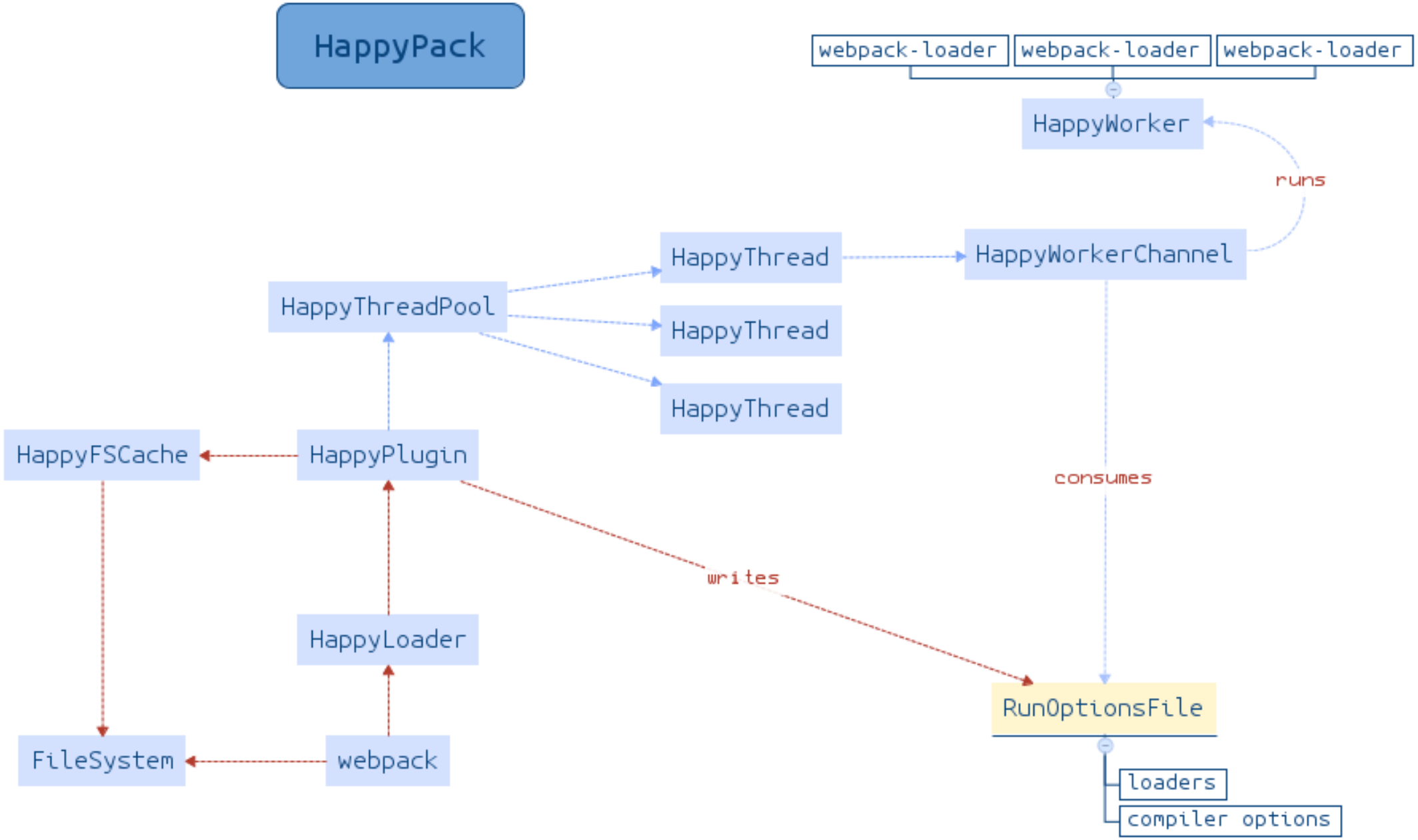
Normal Chunk

Webpack优化

happypack

问题： loader 默认都是一个进程在跑

解法：多进程、缓存


HappyPack
webpack-loader
webpack-loader
webpack-loader
HappyWorker
runs
HappyThread
HappyWorkerChannel
HappyThreadPool
HappyThread
HappyThread
HappyFSCache
HappyPlugin
consumes
writes
HappyLoader
RunOptionsFile
FileSystem
webpack
loaders
compiler options

webpack优化

如何变的更快：

```
resolve: {
  alias: {
    ‘react’: ‘***',
  }
}
```

指定一些模块的引用路径，提高搜索速度

webpack – devtool 配置

devtool	build speed	rebuild speed	production supported	quality
eval	+++	+++	no	generated code
cheap-eval-source-map	+	++	no	transformed code (lines only)
cheap-source-map	+	o	yes	transformed code (lines only)
cheap-module-eval-source-map	o	++	no	original source (lines only)
cheap-module-source-map	o	-	yes	original source (lines only)
eval-source-map	–	+	no	original source
source-map	–	–	yes	original source

```
// webpack.dev.conf.js
devtool: '#cheap-module-eval-source-map'
```

```
// webpack.prod.conf.js
devtool: config.build.productionSourceMap ? '#source-map' : false,
```

默认是 true，建议线上关闭

babel

无处不在的 ES6

babel

```
"devDependencies": {
   "babel-core":  "^6.9.0",
   "babel-loader":  "^6.2.4”,
   "babel-preset-es2015": "^6.9.0",
   "babel-preset-flow-vue": "^1.0.0",
}
```

.babelrc

```
{
  "presets": ["es2015", "flow-vue"],
   "ignore":  [
       "dist/*.js",
       "packages/**/*.js"
]}
```

“plugins”: [“transform-vue-jsx”] ➔ 2.1.0+

“plugins”: [“transform-vue-jsx”, “syntax-dynamic-import”] ➔ 2.2.0+

babel-plugin-transform-vue-jsx

支持JSX

```
➜  ~  eslint -h
eslint [options] file.js [file.js] [dir]

Basic configuration:
  -c, --config path::String    Use configuration from this file or shareable
                               config
  --no-eslintrc                Disable use of configuration from .eslintrc
  --env [String]               Specify environments
  --ext [String]               Specify JavaScript file extensions - default:
                               .js
  --global [String]            Define global variables
  --parser String              Specify the parser to be used
  --parser-options Object      Specify parser options
```

ESLint

```
Use ESLint to lint your code? (Y/n)
? Pick an ESLint preset (Use arrow keys)
› Standard (https://github.com/feross/standard)
  Airbnb (https://github.com/airbnb/javascript)
  none (configure it yourself)
```

.eslintrc
babel-eslint
eslint-loader

eslint --ext .js,.vue src
指定后缀和目录

ESLint -静态检查工具

部分配置说明：

1、parser：脚本解析，默认esprima，现在设置 babel-eslint

2、plugins：通过插件实现自定义规则

现在设置 plugins: [‘html’]，需要安装：eslint-plugin-html

3、extends：校验规则

现在设置 extends: 'standard'

4、rules：

- no-debugger：线上环境为2，其他为0

选择哪家标准：

适当关闭一些：/* eslint-disable no-undef */

ESLint -静态检查工具

```
"devDependencies": {
   "eslint":  "^3.10.1",
   "eslint-config-vue":  "^2.0.1",
   "eslint-loader":  "^1.3.0",
   "eslint-plugin-flowtype": "^2.16.0",
   "eslint-plugin-vue":  "^1.0.0"
}

"scripts": {
   "lint": "eslint src build test”
}
```

Flow

Static type checker for your JavaScript code
避免类型安全问题
v2.0.0-alpha.1 + 遇见 vue.js

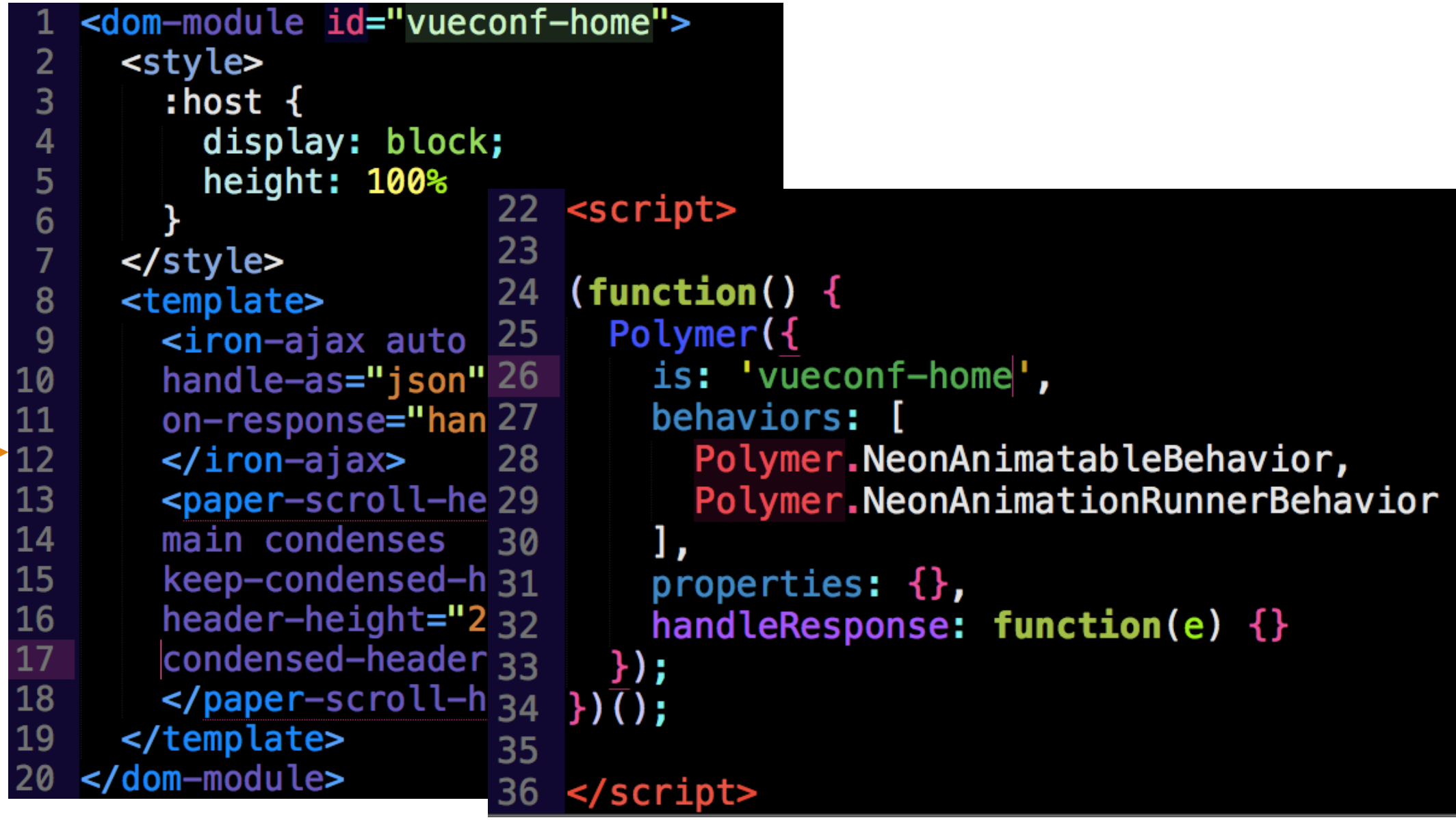
Flow 依赖


```
"devDependencies": {
    "flow-bin":  "^0.39.0"
}

"scripts": {
    "flow": "flow check"
}
```

Vue 组件编译

如何处理 .vue 组件文件

Polymer
.html 文件



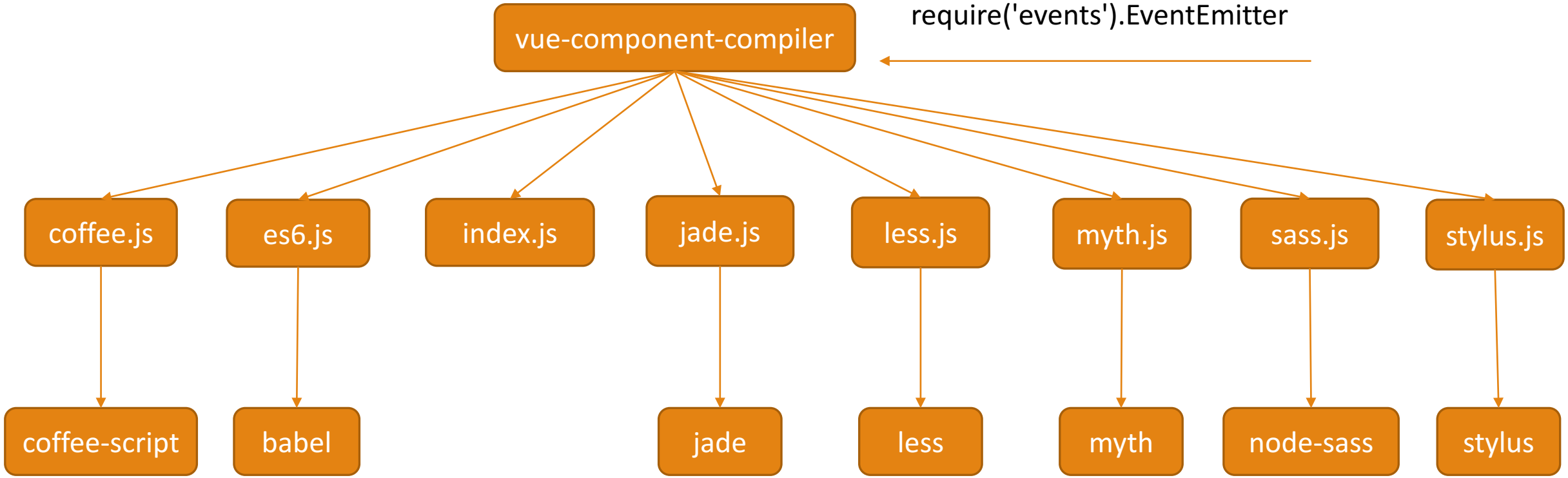
The Vulcanize tool follows HTML Imports and `<script>` tags
to inline these external assets into a single page,
to be used in production

<link rel="import" href="elements/elements.html">

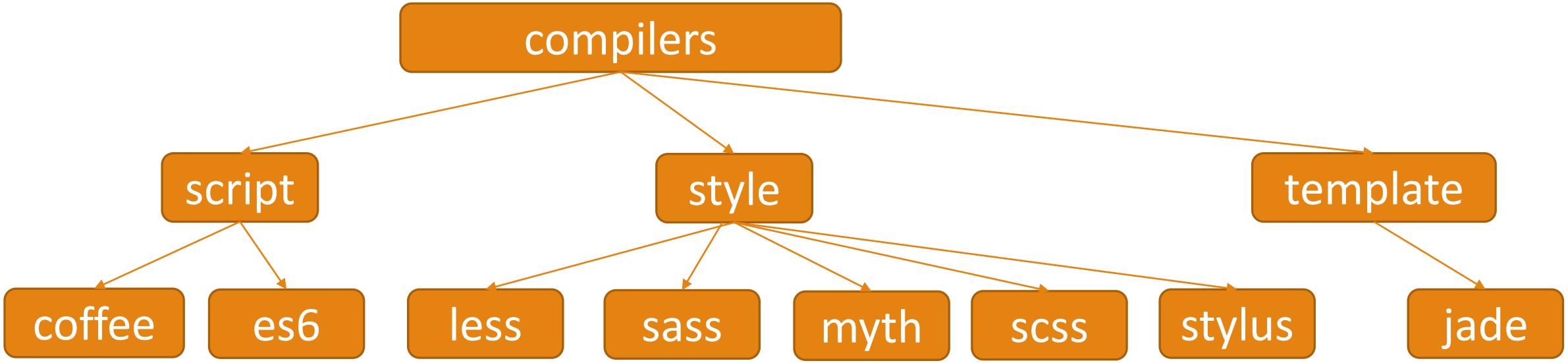
Vue.js 特殊之处
.vue 文件

```vue
<template>
  <div></div>
</template>

<script>

export default {
  data () {
    return {}
  }
}
</script>

<style lang='less' scoped>
</style>
```

vue-component-compiler



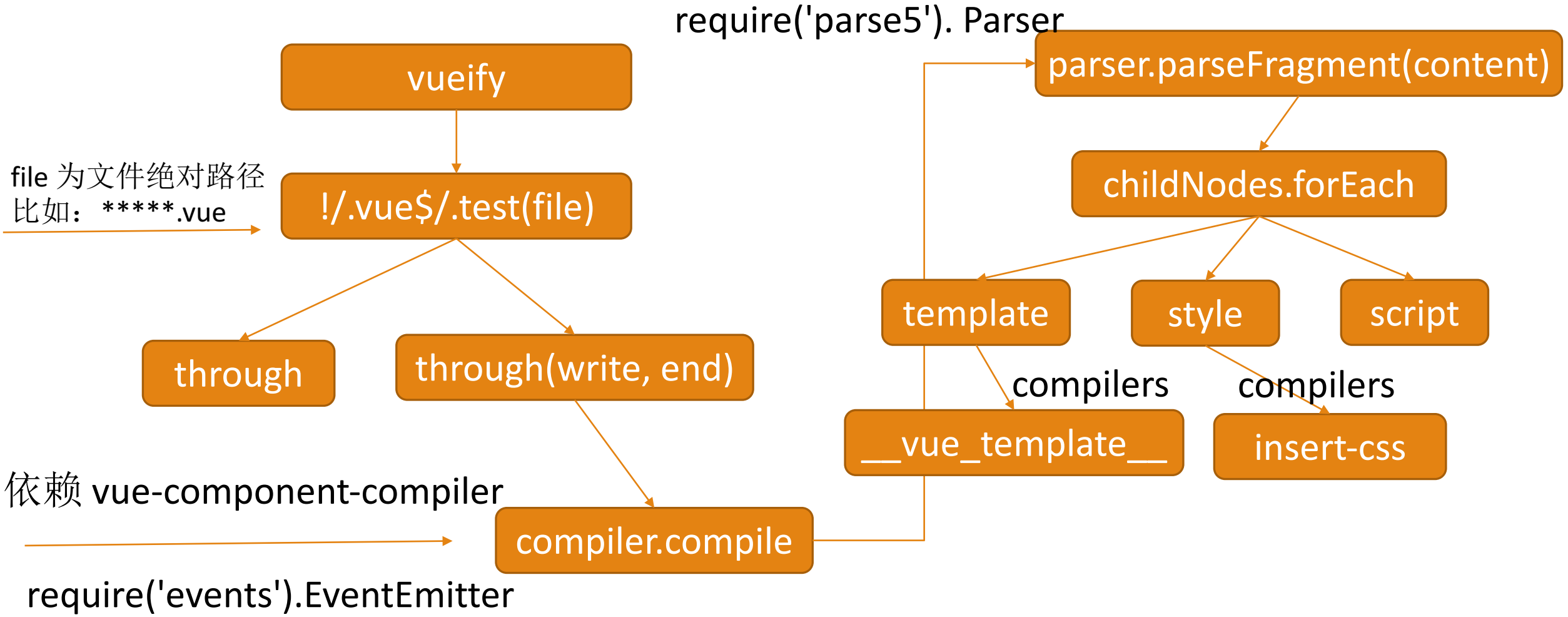
vue-component-compiler



vueify

for browserify

vueify


require('parse5'). Parser
vueify
parser.parseFragment(content)
file 为文件绝对路径
比如： *****.vue
!/.vue$/.test(file)
childNodes.forEach
template
style
script
through
through(write, end)
compilers
compilers
__vue_template__
insert-css
依赖 vue-component-compiler
compiler.compile
require('events').EventEmitter

vue-loader

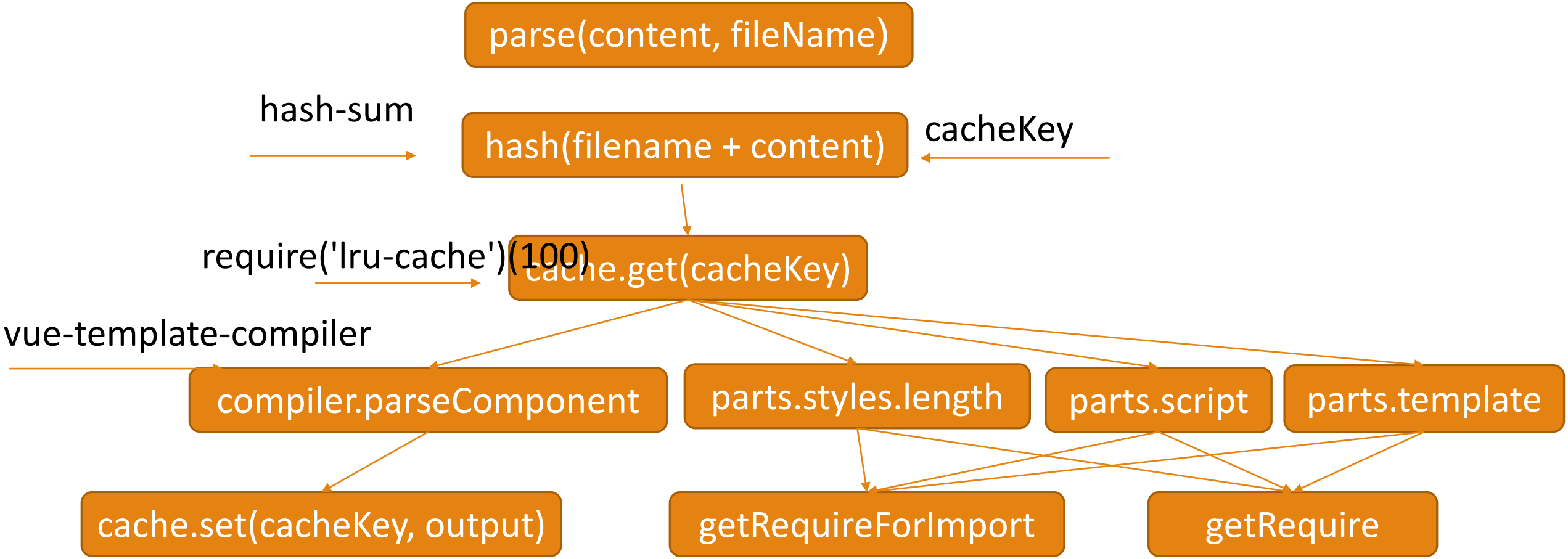
Webpack 的插件，处理 .vue 文件中的三部分：

*、 template

*、 script

*、 styles

vue-loader


parse(content, fileName)
hash-sum
hash(filename + content)
cacheKey
require('lru-cache')(100)
cache.get(cacheKey)
vue-template-compiler
compiler.parseComponent
parts.styles.length
parts.script
parts.template
cache.set(cacheKey, output)
getRequireForImport
getRequire

PostCSS

在 vue-loader 中扮演什么？

Scoped css 的实现

PostCSS

Hello.vue

```
<style scoped>
h1, h2 {
  font-weight: normal;
}
```

```
h1[data-v-29d0db9e], h2[data-   <styl
v-29d0db9e] {
  font-weight: normal;
}
```

```
<div data-v-29d0db9e class="hello"> == $0
  <h1 data-v-29d0db9e>Welcome to Your Vue.js App
  </h1>
  <h2 data-v-29d0db9e>Essential Links</h2>
▶<ul data-v-29d0db9e>…</ul>
  <h2 data-v-29d0db9e>Ecosystem</h2>
▶<ul data-v-29d0db9e>…</ul>
</div>
```

postcss

插入模板：

```
var postcss = require('postcss')

module.exports = postcss.plugin(‘**', function (opts) {
  return function (css) {
  }
})
```

```
var postcss = require('postcss')
var selectorParser = require('postcss-selector-parser')

module.exports = postcss.plugin('add-id', function (opts) {
  return function (root) {
    root.each(function rewriteSelector (node) {
      node.selector = selectorParser(function (selectors) {
       selectors.each(function (selector) {
          var node = null
          selector.each(function (n) {
            if (n.type !== 'pseudo') node = n
          })
        selector.insertAfter(node, selectorParser.attribute({
          attribute: opts.id
        }))
      })
    }).process(node.selector).result    })
}})
```

问题：data-v-29d0db9e 是如何生成的，和什么有关系？

var moduleId = 'data-v-' + genId(filePath, context, options.hashKey)

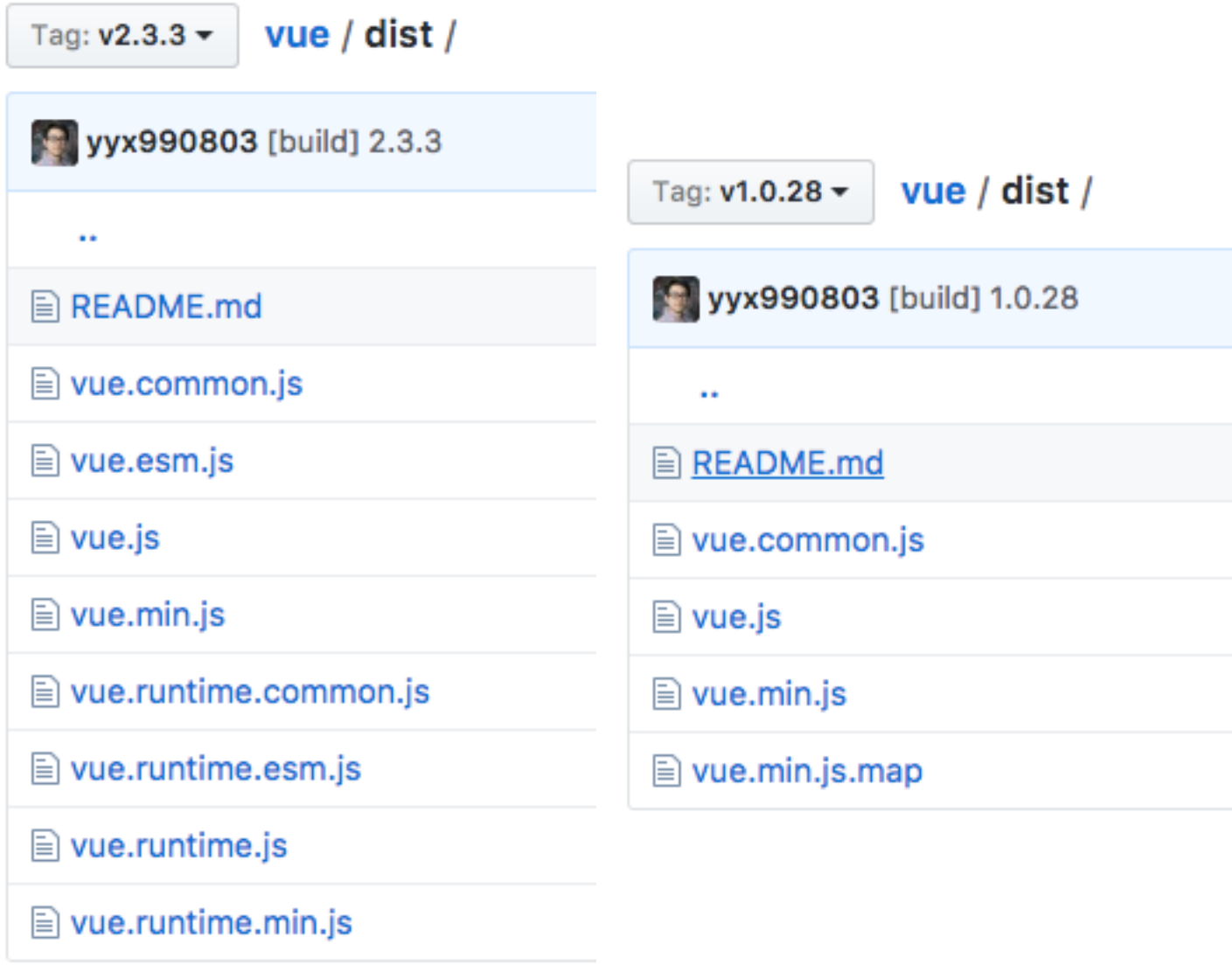
genId

```
3  var path = require('path')
4  var hash = require('hash-sum')
5  var cache = Object.create(null)
6  var sepRE = new RegExp(path.sep.replace('\\', '\\\\'), 'g')
7
8  module.exports = function genId (file, context, key) {
9    var contextPath = context.split(path.sep)
10   var rootId = contextPath[contextPath.length - 1]
11   file = rootId + '/' + path.relative(context, file).
12   replace(sepRE, '/') + (key || '')
13   return cache[file] || (cache[file] = hash(file))
14 }
```

filePath /Users/zhangyaochun/vue-conf/pro1/src/components/Hello.vue

fileName Hello.vue

Vue.js本身如何打包



Rollup

v1.0.9+ 遇见 vue.js

官方：Next-generation ES6 module bundler
目前最新版本：0.41.6
也有cli 工具
支持Node端和浏览器端

Rollup 用途

在 vue 做什么？

编译源码，输出 3 个文件：

dist/vue.common.js

dist/vue.js

dist/vue.min.js

```
"devDependencies": {
   "rollup":  "^0.21.0”
}
```

```
// build/build.js
rollup.rollup({
   entry: 'src/index.js',
   plugins: []
})
```

一般配置

```
module.exports = {
  entry: './src/index.js',
  format: 'amd',
  dest: './dist/mobike.js',
  plugins: [
    eslint({
      useEslintrc: true
    }),
    resolve({
    }),
    commonjs(),
    babel({
      exclude: 'node_modules/**'
    }),
    replace({
      exclude: 'node_modules/**',
      __VERSION__: JSON.stringify(pkg.version)
    })
  ]
};
```

Rollup 基础 – 配置

- ✓ entry

必须的配置：入口文件的路径

不然会抛错：

$ You must supply options.entry to rollup

如图：

- ✓ dest

输出文件的路径

- ✓ format

有如下几个值：

cjs

umd

amd

iife

es 默认值

- ✓ banner
- ✓ plugins
- ✓ external
- ✓ paths

Rollup 基础 – 配置文件

```
"scripts": {
  "dev": "rollup --config rollup.dev.js",
  "build": "rollup --config rollup.prod.js"
},
```

一般采用配置文件

1) -c, --config

可以用来指定配置文件，默认是 rollup.config.js

注释：不支持多个配置文件

$ rollup -c rollup.dev.js rollup.dev2.js

会报错：

```
process.argv [ '/usr/local/bin/node',
  '/usr/local/bin/rollup',
  '-c',
  'rollup.dev.js',
  'rollup.dev2.js' ]
command { _: [ 'rollup.dev2.js' ],
  c: 'rollup.dev.js',
  config: 'rollup.dev.js' }
command._ [ 'rollup.dev2.js' ]
  Could not resolve entry (rollup.dev2.js)
```

Rollup 插件

依赖的插件到底有哪些呢？

rollup-plugin-babel

```
var rollup = require('rollup')
var babel = require('rollup-plugin-babel')

rollup.rollup({
  entry: 'src/index.js',
  plugins: [
    babel({
      loose: 'all'
    })
  ]
})
```

```
babel({
  exclude: 'node_modules/**'
})
```

rollup-plugin-replace

```
var rollup = require('rollup')
var babel = require('rollup-plugin-babel')

rollup.rollup({
  entry: 'src/index.js',
  plugins: [
    replace({
      'process.env.NODE_ENV': "'development'"
    })
  ]
})
```

支持的配置项：

1) delimiters

支持自定义占位变量的分隔符，默认是['', '']如：

```
delimiters: [ '<@', '@>' ]
```

2) include

3) exclude

4) values

值是一个对象，可以直接用 values 来包裹要替换的变量和值：

```
values: {
  __VERSION__: '0.0.2'
}
```

rollup-plugin-alias

v2.0.0-alpha.1 新增

```
var path = require('path')
module.exports = {
  core: path.resolve(__dirname, '../src/core'),
  web: path.resolve(__dirname, '../src/platforms/web')
}
```

```
var rollup = require('rollup')
var aliasPlugin = require('rollup-plugin-alias')

var alias = baseAlias
if (opts.alias) {
  alias = Object.assign({}, baseAlias, opts.alias)
}
plugins.push(aliasPlugin(alias))
```

rollup-plugin-flow-no-whitespace

v2.0.0-rc.4新增
而且抽离出来 config.js

```
var rollup = require('rollup')
const flow = require('./rollup-plugin-flow')

const config = {
  entry: opts.entry,
  dest: opts.dest,
  format: opts.format,
  plugins: [
    flow()
  ]
})
```

Rollup 插件入门

rollup-plugin-replace

源码初始结构：

```
export default function replace ( options = {} ) {
  return {
    name: 'replace',
    transform ( code, id ) {}
  }
}
```

Rollup 插件解刨

配置和编译后

如何设置：

```
//rollup 的配置文件
module.exports = {

  replace({
    exclude: 'node_modules/**',
    __VERSION__:   '"0.0.2"'
  })
}
```

我们的源文件：

```
window.rolluptest.Version = __VERSION__;
```

编译后：

```
window.rolluptest.Version = '0.0.2'
```

1、解析传入 options :

```
{ exclude: 'node_modules/**', __VERSION__: '"0.0.2"' }
```

首先会判断是否采用 values 的方式传递要替换变量和值：

```
const values = options.values || options;
```

然后查看变量的分隔符：

```
function escape ( str ) {
    return str.replace( /[\-\[\]\/\{\}\(\)\*\+\?\.\\\^\$\|]/g, '\\$&' );
}
const delimiters = ( options.delimiters || [ '', '' ] ).map( escape );
```

解析之后返回：['\\$', '\\$']

然后会生成一个正则：

```
const pattern = new RegExp( delimiters[0] + '(' + Object.keys( values ).join( '|' ) +
')' + delimiters[1], 'g' );
```

解析之后：/\$(__VERSION__)\$/g

变量替换是在 transform 里面：

生成匹配正则

通过 MagicString 覆盖内容

```
1)  先实例化
const magicString = new MagicString( code );
2)  然后通过上面的正则去判断
一种：返回被替换之后的对象

// 默认 false，匹配占位的变量
let hasReplacements = false;
let match;
let start, end, replacement;
while ( match = pattern.exec( code ) ) {
    hasReplacements = true;

    start = match.index;
    end = start + match[0].length;
    replacement = String( values[ match[1] ] );

    magicString.overwrite( start, end, replacement );
}
let result = { code: magicString.toString() };
return result;
```

rollup-pluginutils

A set of functions commonly used by Rollup plugins.

包含的方法：

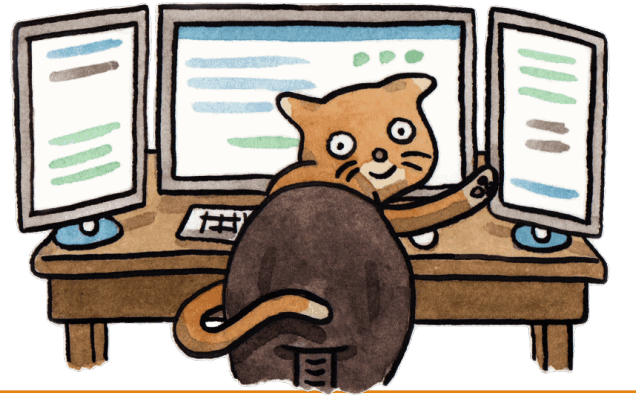
- ✓ addExtension
- ✓ attachScopes
- ✓ createFilter
- ✓ makeLegalIdentifier

rollup-plugin-vue

As vue-loader is for webpack and vueify is for browserify, similarly its for rollup.

```
export default function vue (opts = {}) {
    debug('Yo! rolling vue!')
    const filter = createFilter(opts.include, opts.exclude)

    delete opts.include
    delete opts.exclude

    const config = mergeOptions(DEFAULT_OPTIONS, opts)
    const styles = {}

    return {
        name: 'vue',
        options (opts) {},
        resolveId (id) {},
        load (id) {},
        async transform (source, id) {},
        ongenerate () {}
    }
}
```

yarn & vue

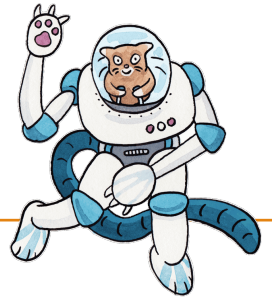


相遇在 v2.0.6 (2016.11)
2016-10-11 来自 Facebook 并和 Google 协作的包管理
快速
安全 - npm publisher 恶意使用
可靠

```
→  ~  brew install yarn
Error: No available formula for yarn
```

安装方式： npm install --global yarn
目前非最新版本： **0.24.4**

yarn



```
# THIS IS AN AUTOGENERATED FILE. DO NOT EDIT THIS FILE DIRECTLY.
# yarn lockfile v1
css-loading@^1.4.0:
  version "1.4.0"
  resolved "https://registry.yarnpkg.com/css-loading/-/css-
  loading-1.4.0.tgz#45e92356745c8ed06b5610d8c17c5f5fba2b52c5"

fastclick@^1.0.6:
  version "1.0.6"
  resolved "https://registry.yarnpkg.com/fastclick/-/fastclick-1
  .0.6.tgz#161625b27b1a5806405936bda9a2c1926d06be6a"
```

1、**离线模式**

相比 Npm，每次安装依赖都得从网络下载

2、**依赖关系确定性**

当没有指定固定的版本号，发生版本更新，会导致再次安装

3、**yarn.lock 文件提交 git，保持依赖包版本的一致 (npm shrinkwrap)**

记录安装的所有工具包的具体版本号，不删除这个文件，再次 yarn install 会依据记录的版本号来获取依赖包

4、扁平模式

5、yarn config set registry 'https://registry.npm.taobao.org'

yarn.lock

https://github.com/vuejs/vue/blob/v2.0.6/yarn.lock

内容如下：

```
# THIS IS AN AUTOGENERATED FILE. DO NOT EDIT THIS FILE DIRECTLY.
# yarn lockfile v1
abbrev@1, abbrev@1.0.x:
  version "1.0.9"
  resolved https://registry.yarnpkg.com/abbrev/-/abbrev-1.0.9.tgz#91b4792588a7738c25f35dd6f63752a2f8776135

accepts@1.1.4:
  version "1.1.4"
  resolved https://registry.yarnpkg.com/accepts/-/accepts-1.1.4.tgz#d71c96f7d41d0feda2c38cd14e8a27c04158df4a
  dependencies:
    mime-types "~2.0.4"
    negotiator "0.4.9"
```

yarn - FAQ

1、**不支持 ***

不能指定版本为 *，需要改成 latest

2、不支持本地 module，不支持除了 npm registry 以外的其他安装方式

3、和 cnpm、pnpm 和 npminstall

宁 js 提到 npminstall
cnpm 采用 link 的方式

3、yarn add/remove

4、采用 deterministic 算法来确保无论安装顺序如何，结果一致。

5、yarn run 不完全达到 npm run 效果

一些工具编写的收获

如下几点：

获取当前运行环境Node.js的版本号：

```
var currentNodeVersion = process.versions.node;
process.version
```

如何设置当前运行环境Node.js的版本大于某个版本：

Package.json设置 engine: "node": ">=4.0.0"

获取：

```
require('../package.json').engines.node
```

在目录下面通过脚手架写入模板文件之前要check：

```
function isSafeToCreateProjectIn(root) {
  var validFiles = [    '.DS_Store', 'Thumbs.db', '.git', '.gitignore',
    '.idea', 'README.md', 'LICENSE', 'web.iml'
  ];
  return fs.readdirSync(root)
    .every(function(file) {
      return validFiles.indexOf(file) >= 0;
    });
}
```

命令行工具包必备之：
commander

检测项目创建的名称是否合法：
validate-npm-package-name

命令行打印一些错误日志：
var chalk = require('chalk')

命令行交互式循环：
Inquirer

输出当前工具的版本：
require('../package').version

如何获取当前 git 配置的用户：

```
require('child_process').execSync('git config --get user.name')
require('child_process').execSync('git config --get user. email')
```

获取 npm 的版本：

```
require('child_process').execSync('npm --version').toString()
```

模板引擎哪个好？

我们推荐 consolidate，支持市面上基本所有的模板引擎

监控某个端口是否被占用：

react-scripts 里面有 detect-port

如何判断是否连网：
dns.resolve(‘registry.yarnpkg.com’, function(err) {})

测试框架选哪个(vue.js用)：
Nightwatch –
an easy to use Node.js based End-to-End (E2E) testing solution for browser based apps and websites.
It uses the powerful W3C WebDriver API to perform commands and assertions on DOM elements

执行脚本：
推荐cross-env：
cross-env NODE_ENV=production webpack --progress --hide-modules

Thanks for all

感谢在场的小伙伴们
感谢尤大大
感谢大会主办方