01. 框架核心Service

vue-cli

Class Service (context, { plugins, pkg, inlineOptions, useBuiltIn } = {})

process.VUE_CLI_SERVICE = this;

this.initilazed = false;

this.context = process.cwd(); // 默认当前工程目录

this.webpackChainFns = [];

this.webpackRawConfigFns = [];

this.devServerConfigFns = [];

this.commands = {}; //命令映射

this.pkgContext = context;

this.pkg = this.resolvePkg(pkg);

this.plugins = this.resolvePlugins(plugins, useBuiltIn); //plugins解析

 [

   {id: 'built-in:commands/help', apply: [Function]},

   {id: 'local:test-plugin.js', apply:[Function]}

]

this.modes = this.plugins.reduce((modes, {apply: {defaultModes}}))... //解析插件中每一个命令的mode，通过注册命令的插件的defaultModes

{serve: 'development', build: 'production', inspect: 'development'}

methods:

this.resolvePkg //解析格式化pkg → this.pkg

this.loadEnv //加载环境变量 //挂在到 → process上

this.resolvePlugins //解析插件 → this.plugins

this.resolveWebpackConfig //解析外部的webpack配置

this.resolveChainableWebpackconfig

this.run

this.init

service init //根据不同mode初始化service的各个部分

this.initialized = true;

this.mode = mode;

this.loadEnv(mode);

this.projectOptions = defaultDeep(this.loadUserOptions(), defaults());

//apply plugins 运行插件

this.plugins.forEach({id, apply} => { apply(new PluginApi(id, this), this.projectOptions)})

```
//webpack config
if(this.projectOptions.chainWebpack) {
  this.webpackChainFns.push(this.projectOptions.chainWebpack);
}
```

service run流程

```
init(mode)
const { fn } = this.commands[name]
fn(args, rawArgv)
```

PluginAPI

```
 ({id, apply})  => {
    apply(new PluginAPI(id, this), this.projectOptions)
}

class PluginAPI (id, service){
  this.id = id
  this.service = service
}
assertVersion
   getCwd return this.service.context;
resolve(_path)
  return path.resolve(this.service.context, _path); //相对工程root的路径
hasPlugin(id)
 return this.service.plugins.some... //service是否包含某个id的plugin
registerCommand(name, opts, fn)
  this.service.commands[name] = {fn, opts} //注册命令到service
chainWebpack(fn)
  this.service.webpackChainFns.push(fn)
configureWebpack(fn)
  this.service.webpackRawConfigFns.push(fn)
configureDevServer(fn)
  this.service.devServerConfigFns.push(fn)
resolveWebpackConfig(chainableConfig)
   return this.service.resolveWebpackConfig(chainableConfig)
resolveChainableWebpackConfig
```

```
  return this.service.resolveChainableWebpackConfig()
```

genCacheConfig

bulit-in plugins

commands:

build help inspect serve

config(webpack base config)