

link: null
title: 珠峰架构师成长计划
description: src/pages/index.js
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=181 sentences=487, words=2930

1. 约定式路由

- umijs (<https://umijs.org/zh-CN>)是可扩展的企业级前端应用框架。Umi 以 `react` 为基础的，并以此进行功能扩展。然后配以生命周期完善的插件体系，支持各种功能扩展和业务需求
- 约定式路由 (<https://umijs.org/zh-CN/docs/convention-routing>)
- umi (<https://github.com/umijs/umi>)

1.1 初始化项目

```
mkdir zhufeng-umi3  
npm init -y  
cnpm i umi -D
```

1.2 运行项目

1.2.1 pages/index.js

src/pages/index.js

```
import React from 'react';  
const Index = () => <div>首页div</div>  
export default Index
```

1.2.2 pages/profile.js

src/pages/profile.js

```
import React from 'react';  
const Profile = () => <div>个人中心div</div>  
export default Profile;
```

1.2.3 pages/user_layout.js

src/pages/user_layout.js

```
import React from 'react';  
const UserLayout = (props) => (  
  <div>  
    <ul>  
      <li><a href="/user/add">添加用户</a></li>  
      <li><a href="/user/list">用户列表</a></li>  
    </ul>  
    <div>{props.children}</div>  
  </div>  
)  
export default UserLayout;
```

1.2.4 pages/user/add.js

src/pages/user/add.js

```
import React from 'react';  
const UserAdd = () => <div>添加用户div</div>  
export default UserAdd;
```

1.2.5 pages/user/list.js

src/pages/user/list.js

```
import React from 'react';  
const UserList = () => <div>用户列表div</div>  
export default UserList;
```

1.2.6 package.json

package.json

```
{  
  "scripts": {  
    "dev": "umi dev"  
  }  
}
```

1.2.7 启动

```
npm run dev
```

2. 调试

- debugging (<https://nodejs.org/en/docs/guides/debugging-getting-started/>)
 - `--inspect-brk` 指定在第一行就设置断点。也就是说，一开始运行，就是暂停的状态

2.1 NPM调试

2.1.1 launch.json

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Launch via NPM",
      "request": "launch",
      "runtimeArgs": [
        "run-script",
        "dev"
      ],
      "runtimeExecutable": "npm",
      "skipFiles": [
        "/*"
      ],
      "type": "pwa-node"
    }
  ]
}
```

2.2 attach调试

2.2.1 package.json

package.json

```
"scripts": {
  "dev": "umi dev",
+  "debug": "node --inspect-brk=9229 ./node_modules/_umi@3.3.1@umi/bin/umi.js dev"
}
```

2.2.2 launch.json

.vscode\launch.json

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Attach to UMI",
      "port": 9230,
      "request": "attach",
      "type": "node"
    }
  ]
}
```

.vscode\launch.json

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Attach to Remote",
      "port": 9230,
      "request": "attach",
      "type": "node"
    }
  ]
}
```

3. 运行时

3.1 安装

```
cnpm i webpack webpack-cli webpack-dev-server html-webpack-plugin babel-loader @babel/core @babel/preset-env @babel/preset-react webpack-dev-middleware express
mustache cnpm react react-dom react-router-dom -S
```

3.2 .umi2\core\history.js

src.umi2\core\history.js

```
import { createBrowserHistory } from 'history';
let history = createBrowserHistory();
export default history;
```

3.3 .umi2\core\routes.js

src.umi2\core\routes.js

```

export function getRoutes() {
  const routes = [
    {
      "path": "/",
      "exact": true,
      "component": require('@pages/index.js').default
    },
    {
      "path": "/profile",
      "exact": true,
      "component": require('@pages/profile.js').default
    },
    {
      "path": "/user",
      "routes": [
        {
          "path": "/user/add",
          "exact": true,
          "component": require('@pages/user/add.js').default
        },
        {
          "path": "/user/list",
          "exact": true,
          "component": require('@pages/user/list.js').default
        }
      ]
    },
    {
      "component": require('@pages/user/_layout.js').default
    }
  ]
};
return routes;
}

```

3.4.umi2lumi.js

src.umi2lumi.js

```

import React from 'react';
import { render } from 'react-dom';
import { Router,Route,Switch } from 'react-router-dom';
import history from './core/history';
import { getRoutes } from './core/routes';
function renderRoutes(routes){
  return routes.map(({path,exact,component:Component,routes:childRoutes=[]},index)=>{
    <Route key={index} path={path} exact={exact} render={
      props=>{
        <Component {...props}>
          <Switch>
            {renderRoutes(childRoutes)}
          </Switch>
        </Component>
      }
    }/>
  })
}
function renderClient(opts) {
  render(
    <Router history={opts.history}>
      {renderRoutes(opts.routes)}
    </Router>
    ,document.getElementById(opts.rootElement));
}
const opts = {
  routes:getRoutes(),
  history,
  rootElement: 'root'
}
renderClient(opts);

```

3.5 src.umi2index.html

src.umi2index.html

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>umi3title</title>
</head>
<body>
  <div id="root">div</div>
</body>
</html>

```

3.6 webpack.config.js

```

const path = require("path");
const HtmlWebpackPlugin = require("html-webpack-plugin");
const cwd = process.cwd();
module.exports = {
  mode: "development",
  entry: './src/.umi2/umi.js',
  output: {
    path: path.join(__dirname, "dist"),
    filename: "[name].js",
    publicPath: '/'
  },
  devtool: false,
  resolve: {
    alias: {
      '@': path.join(cwd, 'src')
    }
  },
  devServer: {
    historyApiFallback: {
      index: 'index.html'
    }
  },
  module: {
    rules: [
      {
        test: /\.js$/,
        loader: "babel-loader",
        options: {
          presets: ['@babel/preset-env', '@babel/preset-react']
        },
        exclude: /node_modules/
      },
    ]
  },
  plugins: [
    new HtmlWebpackPlugin({
      template: './src/.umi2/index.html'
    })
  ]
};

```

3.7 启动

```
webpack serve
```

4. 编译时插件

- [插件 API \(https://umijs.org/zh-CN/plugins/api\)](https://umijs.org/zh-CN/plugins/api)
- [Service.ts \(https://github.com/umijs/umi/blob/master/packages/core/src/Service/Service.ts\)](https://github.com/umijs/umi/blob/master/packages/core/src/Service/Service.ts)
- [PluginAPI.ts \(https://github.com/umijs/umi/blob/master/packages/core/src/Service/PluginAPI.ts\)](https://github.com/umijs/umi/blob/master/packages/core/src/Service/PluginAPI.ts)
- [registerCommand \(https://umijs.org/zh-CN/plugins/api#registercommand-name-string-alias-string-fn-function-\)](https://umijs.org/zh-CN/plugins/api#registercommand-name-string-alias-string-fn-function-)

4.1 bin\umi.js

```
bin\umi.js
```

```
require('../lib/cli');
```

4.2 lib\cli.js

```
lib\cli.js
```

```

let Service = require('./Service');
let dev = require('./plugins/commands/dev');
(async () => {
  const service = new Service({
    plugins: [{id: 'dev', apply: dev}]
  });
  await service.run({name: 'dev'});
})();

```

4.3 lib\PluginAPI.js

```
lib\PluginAPI.js
```

```

class PluginAPI {
  constructor(opts) {
    this.id = opts.id;
    this.service = opts.service;
  }
  registerCommand(command) {
    const { name } = command;
    this.service.commands[name] = command;
  }
}
module.exports = PluginAPI;

```

4.4 lib\Service.js

```
lib\Service.js
```

```

const PluginAPI = require('./PluginAPI');
class Service {
  constructor(opts) {
    this.commands={};
    this.plugins = opts.plugins;
  }
  async init() {
    for(let plugin of this.plugins){
      const pluginAPI = new PluginAPI({id:plugin.id,service: this });
      plugin.apply(pluginAPI);
    }
  }
  async run(args) {
    this.init();
    return this.runCommand(args);
  }
  async runCommand({name}) {
    const command = this.commands[name];
    const { fn } = command;
    return fn();
  }
}
module.exports = Service;

```

4.5 commands\dev.js <#>

lib\plugins\commands\dev.js

```

let Server = require('../../Server');
const dev = (api) => {
  api.registerCommand({
    name: "dev",
    description: "启动开发服务器",
    fn: async function () {
      new Server().start();
    }
  });
};
module.exports = dev;

```

4.6 lib\Server.js <#>

lib\Server.js

```

let express = require('express');
let http = require('http');
class Service {
  constructor(){
    this.app = express();
  }
  async start() {
    this.listeningApp = http.createServer(this.app);
    this.listeningApp.listen(8000, () => {
      console.log(`http server started at port 8000`);
    })
  }
}
module.exports = Service;

```

4.7 启动服务 <#>

```
node ./bin/umi.js
```

5. 生成临时目录 <#>

5.1 lib\Service.js <#>

lib\Service.js

```

const PluginAPI = require('./PluginAPI');
+const {AsyncParallelHook} = require('tapable');
class Service {
  constructor(opts) {
    this.commands={};
    this.plugins = opts.plugins;
+   this.hooksByPluginId = {};
+   this.hooks = {};
  }
  async init() {
    for(let plugin of this.plugins){
      const pluginAPI = new PluginAPI({id:plugin.id,service: this });
+     pluginAPI.onGenerateFiles=(fn)=>{
+       pluginAPI.register({pluginId:plugin.id,key:"onGenerateFiles",fn})
+     }
      plugin.apply(pluginAPI);
    }

+   Object.keys(this.hooksByPluginId).forEach((id) => {
+     const hooks = this.hooksByPluginId[id];
+     hooks.forEach((hook) => {
+       const { key } = hook;
+       hook.pluginId = id;
+       this.hooks[key] = (this.hooks[key] || []).concat(hook);
+     });
+   });
+   async applyPlugins(opts) {
+     const hooks = this.hooks[opts.key] || [];
+     const tEvent = new AsyncParallelHook([""]);
+     for (const hook of hooks) {
+       tEvent.tapPromise({ name: hook.pluginId },hook.fn);
+     }
+     return await tEvent.promise();
+   }
  async run(args) {
    this.init();
    return this.runCommand(args);
  }
  async runCommand({name}) {
    const command = this.commands[name];
    const { fn } = command;
    return fn();
  }
}
module.exports = Service;

```

5.2 lib\PluginAPI.js

lib\PluginAPI.js

```

class PluginAPI {
  constructor(opts) {
    this.id = opts.id;
    this.service = opts.service;
  }
  registerCommand(command) {
    const { name } = command;
    this.service.commands[name] = command;
  }
+  register(hook) {
+    this.service.hooksByPluginId[this.id] = (
+      this.service.hooksByPluginId[this.id] || []
+    ).concat(hook);
+  }
}
module.exports = PluginAPI;

```

5.3 lib\cli.js

lib\cli.js

```

let Service = require('./Service');
let dev = require('./plugins/commands/dev');
+let history = require('./plugins/generateFiles/history');
+let routes = require('./plugins/generateFiles/routes');
+let umi = require('./plugins/generateFiles/umi');
(async())=>{
  const service = new Service({
    plugins:[
      {id:'dev',apply:dev},
+     {id:'history',apply:history},
+     {id:'routes',apply:routes},
+     {id:'umi',apply:umi}
    ]
  });
  await service.run({name: 'dev'});
})();

```

5.4 webpack.config.js

webpack.config.js

```

module.exports = {
  mode: "development",
+  entry: './src/.umi3/umi.js'
}

```

5.5 lib\plugins\commands\dev.js

lib\plugins\commands\dev.js

```

let Server = require('../Server');
const dev = (api) => {
  api.registerCommand({
    name: "dev",
    description: "启动开发服务器",
    fn: async function () {
+      await api.service.applyPlugins({
+        key: 'onGenerateFiles'
+      });
      new Server().start();
    }
  });
};

module.exports = dev;

```

5.6 lib\getPaths.js

lib\getPaths.js

```

const {existsSync,statSync} = require('fs');
const {join} = require('path');
function isDirectoryAndExist(path) {
  return existsSync(path) && statSync(path).isDirectory();
}
let cwd = process.cwd();
let absSrcPath = cwd;
if (isDirectoryAndExist(join(cwd, 'src'))) {
  absSrcPath = join(cwd, 'src');
}
const absPagesPath=join(absSrcPath,'pages');
const tmpDir = '.umi3';
const absTmpPath = join(absSrcPath, tmpDir);
module.exports = {
  absSrcPath,
  absPagesPath,
  tmpDir,
  absTmpPath
}

```

5.7 lib\utils.js

lib\utils.js

```

function winPath(path) {
  return path.replace(/\\/g, "/");
}
exports.winPath = winPath;

```

5.8 lib\writeTmpFile.js

lib\writeTmpFile.js

```

let mkdirp = require('mkdirp');
let {writeFileSync} = require('fs');
let {dirname,join} = require('path');
const {absTmpPath} = require('./getPaths');
function writeTmpFile({path,content}) {
  const absPath = join(absTmpPath, path);
  mkdirp.sync(dirname(absPath));
  writeFileSync(absPath, content, 'utf8');
}
module.exports = writeTmpFile;

```

5.9 generateFiles\history.js

lib\plugins\generateFiles\history.js

```

let { readFileSync } = require("fs");
let { join } = require("path");
let writeTmpFile = require("../writeTmpFile");
let Mustache = require("mustache");

const plugin = (api) => {
  api.onGenerateFiles(async () => {
    const historyTpl = readFileSync(join(__dirname, "history.tpl"), "utf8");
    writeTmpFile({
      path: "core/history.js",
      content: Mustache.render(historyTpl),
    });
  });
};

module.exports = plugin;

```

5.10 generateFiles\history.tpl

lib\plugins\generateFiles\history.tpl

```

import { createBrowserHistory } from 'history';
let history = createBrowserHistory();
export default history;

```

5.11 generateFiles\umi.js

lib\plugins\generateFiles\umi.js

```

let { readFileSync } = require("fs");
let { join } = require("path");
let writeTmpFile = require("../writeTmpFile");
let Mustache = require("mustache");

const plugin = (api) => {
  api.onGenerateFiles(async () => {
    const umiTpl = readFileSync(join(__dirname, "umi.tpl"), "utf8");
    writeTmpFile({
      path: "umi.js",
      content: Mustache.render(umiTpl),
    });
  });
};
module.exports = plugin;

```

5.12 generateFiles\umi.tpl

lib\plugins\generateFiles\umi.tpl

```

import React from 'react';
import { render } from 'react-dom';
import { Router, Route, Switch } from 'react-router-dom';
import history from './core/history';
import { getRoutes } from './core/routes';
function renderRoutes(routes) {
  return routes.map(({ path, exact, component: Component, routes: childRoutes = [], index }) => (
    <Route key={index} path={path} exact={exact} render={
      props => (
        <Component {...props}>
          <Switch>
            {renderRoutes(childRoutes)}
          </Switch>
        </Component>
      )
    } />
  ))
}
function renderClient(opts) {
  render(
    <Router history={opts.history}>
      {renderRoutes(opts.routes)}
    </Router>
    , document.getElementById(opts.rootElement));
}
const opts = {
  routes: getRoutes(),
  history,
  rootElement: 'root'
}
renderClient(opts);

```

5.13 lib\getRoutes.js

lib\getRoutes.js


```

const { existsSync, readdirSync, readFileSync, statSync } = require("fs");
const { basename, extname, join, relative } = require("path");
const { winPath } = require('./utils');
const { absPagesPath } = require('./getPaths');
function getRoutes(opts={}) {
  const { root, relDir = "" } = opts;
  const files = getFiles(join(root, relDir));
  const routes = files.reduce(fileToRouteReducer.bind(null, opts), []);
  return routes;
}

function fileToRouteReducer(opts, routes, file) {
  const { root, relDir = "" } = opts;
  const absFile = join(root, relDir, file);
  const stats = statSync(absFile);
  if (stats.isDirectory()) {
    const relFile = join(relDir, file);
    const layoutFile = join(root, relFile, '_layout.js');
    routes.push({
      path: normalizePath(relFile),
      routes: getRoutes({
        ...opts,
        relDir: relFile
      }),
      ...(existsSync(layoutFile)?{ component: toComponentPath(root, layoutFile)}:{})
    });
  } else {
    const bName = basename(file, extname(file));
    routes.push(
      {
        path: normalizePath(join(relDir, bName)),
        exact: true,
        component: toComponentPath(root, absFile),
      }
    );
  }
  return routes;
}

const normalizePath = (path)=>{
  path= winPath(path);
  path = `/${path}`;
  path = path.replace(/\\/index$/, '/');
  return path;
}

const toComponentPath = (root, absFile)=>{
  return `@/${winPath(relative(join(root, ".."), absFile))}`;
}

function getFiles(root) {
  if (!existsSync(root)) return [];
  return readdirSync(root).filter((file) => {
    if (file.charAt(0) === '_' ) return false;
    return true;
  });
}

let routes = getRoutes({root:absPagesPath});
module.exports = routes;

```

5.14 generateFiles\routes.js

lib\plugins\generateFiles\routes.js

```

let { readFileSync } = require("fs");
let { join } = require("path");
let writeTmpFile = require("../writeTmpFile");
let Mustache = require("mustache");
let routes = require('.././getRoutes');
const plugin = (api) => {
  api.onGenerateFiles(async () => {
    const routesTpl = readFileSync(join(__dirname, "routes.tpl"), "utf8");
    writeTmpFile({
      path: "core/routes.js",
      content: Mustache.render(routesTpl, {
        routes: JSON.stringify(routes, replacer, 2).replace(/\"component\": (\\"(?:.+?)\"\/g, (global, m1, m2) => {
          return "component": ${m2.replace(/\\/g, '\\')}";
        })
      })
    });
  });
};
function replacer(key, value) {
  switch (key) {
    case "component":
      return `require('${value}').default`;
    default:
      return value;
  }
}
module.exports = plugin;

```

5.15 generateFiles\routes.tpl

lib\plugins\generateFiles\routes.tpl

```

export function getRoutes() {
  const routes = {{{ routes }}};
  return routes;
}

```

5.16 启动

webpack serve

6. 启动服务器

6.1 lib\Service.js

```
const express = require('express');
const http = require('http');
+const webpack = require('webpack');
+const {join} = require('path');
+const webpackDevMiddleware = require('webpack-dev-middleware');
+const HtmlWebpackPlugin = require("html-webpack-plugin");
+const webpackConfig = require('./webpack.config.js');
+const {absTmpPath,absSrcPath} = require('./getPaths');
class Service {
  constructor() {
    this.app = express();
  }
+  setupDevMiddleware(){
+    webpackConfig.entry = join(absTmpPath, 'umi.js');
+    webpackConfig.resolve.alias['@']=absSrcPath;
+    webpackConfig.plugins.push(new HtmlWebpackPlugin({template:join(__dirname, "index.html")}));
+    const compiler = webpack(webpackConfig);
+    const devMiddleware = webpackDevMiddleware(compiler,{writeToDisk:true});
+    this.app.use(devMiddleware);
+    this.app.use((req,res)=>{
+      res.send(compiler.outputFileSystem.readFileSync(join(__dirname,'dist/index.html'),'utf8'));
+    });
+    return devMiddleware;
+  }
  async start() {
    const devMiddleware = this.setupDevMiddleware();
+    devMiddleware.waitUntilValid() => {
+      this.listeningApp = http.createServer(this.app);
+      this.listeningApp.listen(8000, () => {
+        console.log(`http server started at port 8000`);
+      })
+    };
  }
}
module.exports = Service;
```

6.2 lib\index.html

umi3

6.3 lib\webpack.config.js

lib\webpack.config.js

```
const path = require("path");
const HtmlWebpackPlugin = require("html-webpack-plugin");
const cwd = process.cwd();
module.exports = {
  mode: "development",
  output: {
    path: path.join(__dirname, "dist"),
    filename: "[name].js",
    publicPath: '/'
  },
  devtool: false,
  resolve: {
    alias: {}
  },
  devServer: {
    historyApiFallback: {
      index: 'index.html'
    }
  },
  module: {
    rules: [
      {
        test: /\.js$/,
        loader: "babel-loader",
        options: {
          presets: ['@babel/preset-env', '@babel/preset-react']
        },
        exclude: /node_modules/
      },
    ],
  },
  plugins: []
};
```

7. 运行时插件

7.1 src\app.js

src\app.js

```
export function patchRoutes((routes)) {
  routes.unshift({
    path: '/foo',
    exact: true,
    component: require('@foo').default,
  });
}
```

7.2 src\foo.js

src\foo.js

```
import React from 'react';
const Foo = (props) => {
  return (
    <div>Foodiv</div>
  )
}
export default Foo;
```

7.3 src.umi3\core\plugin.js

src.umi3\core\plugin.js

```
class Plugin {
  constructor() {
    this.hooks = {};
  }
  register(plugin) {
    Object.keys(plugin.apply).forEach((key) => {
      if (!this.hooks[key]) this.hooks[key] = [];
      this.hooks[key] = this.hooks[key].concat(plugin.apply[key]);
    });
  }
  applyPlugins(({ key, args }) {
    const hooks = this.hooks[key] || [];
    hooks.forEach((hook) => {
      hook(args);
    });
  })
}

const plugin = new Plugin();
import * as PluginApp from 'C:/aprepare/zhufeng-umi3-prepare3/src/app.js';
plugin.register({
  apply: PluginApp,
  path: 'C:/aprepare/zhufeng-umi3-prepare3/src/app.js',
});

export { plugin }
```

7.4 src.umi3\core\routes.js

src.umi3\core\routes.js

```
+import { plugin } from './plugin';
export function getRoutes() {
  const routes = [
    {
      "path": "/",
      "exact": true,
      "component": require('@pages/index.js').default
    },
    {
      "path": "/profile",
      "exact": true,
      "component": require('@pages/profile.js').default
    }
  ];
  + plugin.applyPlugins({
  +   key: 'patchRoutes',
  +   args: { routes }
  + });
  return routes;
}
```

8. 实现运行时插件

8.1 plugin.js

lib\plugins\generateFiles\plugin.js

```
let { readFileSync, existsSync } = require("fs");
let { join } = require("path");
let writeTmpFile = require("../writeTmpFile");
let Mustache = require("mustache");
const { winPath } = require("../utils");
const { absSrcPath } = require("../getPaths");

const plugin = (api) => {
  api.onGenerateFiles(async () => {
    const pluginTpl = readFileSync(join(__dirname, "plugin.tpl"), "utf8");
    const plugins = [join(absSrcPath, 'app.js')];
    writeTmpFile({
      path: "core/plugin.js",
      content: Mustache.render(pluginTpl, {
        plugins: plugins.map((plugin, index) => {
          return {
            index,
            path: winPath(plugin)
          };
        })
      })
    });
  });
};
module.exports = plugin;
```

8.2 plugin.tpl

lib\plugins\generateFiles\plugin.tpl

```

import { plugin } from './plugin';
class Plugin {
  constructor() {
    this.hooks = {};
  }
  register(plugin) {
    Object.keys(plugin.apply).forEach((key) => {
      if (!this.hooks[key]) this.hooks[key] = [];
      this.hooks[key] = this.hooks[key].concat(plugin.apply[key]);
    });
  }
  applyPlugins({ key, args }) {
    const hooks = this.hooks[key] || [];
    hooks.forEach((hook) => {
      hook(args);
    });
  }
}

const plugin = new Plugin();
{{#plugins}}
import * as Plugin_{{ index }} from '{{ path }}';
{{/plugins}}

{{#plugins}}
plugin.register({
  apply: Plugin_{{ index }},
  path: '{{ path }}',
});
{{/plugins}}

export { plugin }

```

8.3 routes.tpl

lib\plugins\generateFiles\routes.tpl

```

+import { plugin } from './plugin';
export function getRoutes() {
  const routes = {{ routes }};
+  plugin.applyPlugins({
+    key: 'patchRoutes',
+    args: { routes },
+  });
  return routes;
}

```