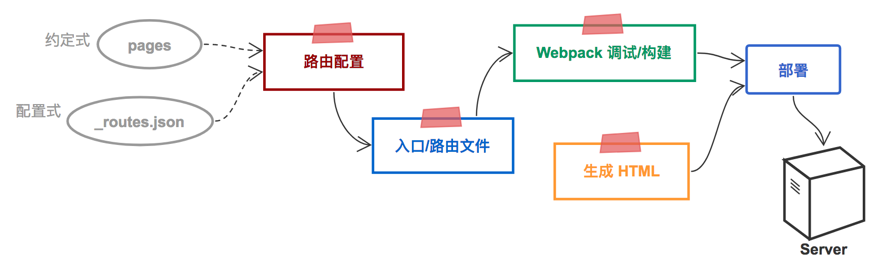
## **1.UmiJS**

* [UmiJS](https://umijs.org/zh/guide/) 是一个类 Next.JS 的 react 开发框架。
* 他基于一个约定，即 pages 目录下的文件即路由，而文件则导出 react 组件
* 然后打通从源码到产物的每个阶段，并配以完善的插件体系，让我们能把 umi 的产物部署到各种场景里。



## **2.安装**

* [umi源码](https://github.com/umijs/umi)
* [create-umi](https://github.com/umijs/create-umi)
* [umi-plugin-react文档](https://umijs.org/zh/plugin/umi-plugin-react.html)
* [umi-plugin-react源码](https://github.com/umijs/umi/tree/master/packages/umi-plugin-react)
* [umi-plugin-dva](https://github.com/umijs/umi/tree/master/packages/umi-plugin-dva)
* [dva-immer](https://github.com/dvajs/dva/blob/master/packages/dva-immer/src/index.js)
* [immer](https://immerjs.github.io/immer/docs/introduction)
* [umi-blocks](https://github.com/umijs/umi-blocks)
* [pro-blocks](https://github.com/ant-design/pro-blocks)

cnpm install -g umi

### **2.1 目录约定**

.

├── dist/ // 默认的 build 输出目录

├── mock/ // mock 文件所在目录，基于 express

├── config/

├── config.js // umi 配置，同 .umirc.js，二选一

└── src/ // 源码目录，可选

├── layouts/index.js // 全局布局

├── pages/ // 页面目录，里面的文件即路由

├── .umi/ // dev 临时目录，需添加到 .gitignore

├── .umi-production/ // build 临时目录，会自动删除

├── document.ejs // HTML 模板

├── 404.js // 404 页面

├── page1.js // 页面 1，任意命名，导出 react 组件

├── page1.test.js // 用例文件，umi test 会匹配所有 .test.js 和 .e2e.js 结尾的文件

└── page2.js // 页面 2，任意命名

├── global.css // 约定的全局样式文件，自动引入，也可以用 global.less

├── global.js // 可以在这里加入 polyfill

├── .umirc.js // umi 配置，同 config/config.js，二选一

├── .env // 环境变量

└── package.json

## **3. 新建项目**

### **3.1 新建项目目录**

mkdir zhufeng-umi

cd zhufeng-umi

cnpm init -y

### **3.2 新建pages目录**

mkdir pages

### **3.3 新建页面**

* [generate](https://github.com/umijs/umi/blob/5f450307b47f79fdd5fc8904294fd1aea6709f8c/packages/umi-build-dev/src/plugins/commands/generate/index.js)
* [link.js](https://github.com/umijs/umi/blob/master/packages/umi/src/link.js)
* [router.js](https://github.com/umijs/umi/blob/master/packages/umi/src/router.js)

#### **3.3.1 Home组件**

umi g page index

pages\index.js

**import** React, { Component } **from** 'react'**import** Link **from** 'umi/link';**export** **default** **class** **componentName** **extends** **Component** {

render() {

**return** (

<div>

首页

<Link to="/profile">个人中心</Link>

</div>

)

}

}

#### **3.3.2 个人中心**

umi g page profile

pages\profile.js

**import** React, { Component } **from** 'react'**import** router **from** 'umi/router';**export** **default** **class** **componentName** **extends** **Component** {

render() {

**return** (

<div>

<button onClick={()=>router.goBack()}>返回</button>

个人中心

</div>

)

}

}

#### **3.3.3 启动服务器**

##### **3.3.3.1 启动配置**

"scripts": {

"dev": "umi dev",

"build": "umi build",

"test": "umi test",

"serve": "serve ./dist"

}

##### **3.3.3.2 启动项目**

npm run dev

##### **3.3.3.3 部署发布**

npm run build

##### **3.3.3.4 本地验证**

npm run serve

##### **3.3.3.5 测试**

test\sum.test.js

**let** assert = require('assert');

describe('sum',()=>{

it('1+1',()=>{

assert(1+1==2);

});

});

npm run test

## **4. 全局 layout**

约定 src/layouts/index.js 为全局路由，返回一个 React 组件，通过 props.children 渲染子组件。

cnpm i bootstrap@3 -S

src/layouts/index.js

**import** React,{Component,Fragment} **from** 'react';**import** 'bootstrap/dist/css/bootstrap.css';**import** Link **from** 'umi/link';**export** **default** **class** **Layout** **extends** **Component**{

render() {

**return** (

<Fragment>

<nav className="navbar navbar-default">

<div className="container-fluid">

<div className="navbar-header">

<Link to="/" className="navbar-brand">珠峰培训</Link>

</div>

<div>

<ul className="nav navbar-nav">

<li className="active"><Link to="/" >首页</Link></li>

<li><Link to="/user" >用户管理</Link></li>

<li><Link to="/profile">个人设置</Link></li>

</ul>

</div>

</div>

</nav>

<div className="container">

<div className="row">

<div className="col-md-12">

{this.props.children}

</div>

</div>

</div>

</Fragment>

)

}

}

## **5. 用户管理**

### **5.1 嵌套路由**

* umi 里约定目录下有 \_layout.js 时会生成嵌套路由，以 \_layout.js 为该目录的 layout pages/user/\_layout.js

**import** React,{Component,Fragment} **from** 'react';**import** Link **from** 'umi/link';**export** **default** **class** **User** **extends** **Component**{

render() {

**return** (

<div className="row">

<div className="col-md-3">

<ul className="nav nav-stack">

<li><Link to="/user/list">用户列表</Link></li>

<li><Link to="/user/add">新增用户</Link></li>

</ul>

</div>

<div className="col-md-9">

{this.props.children}

</div>

</div>

)

}

}

### **5.2 user/list.js**

/pages/user/list.js

**import** React,{Component,Fragment} **from** 'react';**import** Link **from** 'umi/link';**export** **default** **class** **List** **extends** **Component**{

render() {

**return** (

<ul className="list-group">

<li className="list-group-item">

<Link to="/user/detail/1">1</Link>

</li>

</ul>

)

}

}

### **5.3 pages/user/add.js**

pages/user/add.js

**import** React,{Component,Fragment} **from** 'react';**export** **default** **class** **Add** **extends** **Component**{

render() {

**return** (

<form className="form-horizontal">

<div className="form-group">

<label className="control-label col-md-2">用户名</label>

<div className="col-md-10">

<input className="form-control" />

</div>

</div>

<div className="form-group">

<div className="col-md-10 col-offset-2">

<input type="submit" className="btn btn-primary"/>

</div>

</div>

</form>

)

}

}

### **5.3 动态路由**

* umi 里约定，带 $ 前缀的目录或文件为动态路由。 pages/user/detail/$id.js

**import** React,{Component,Fragment} **from** 'react';**export** **default** **class** **List** **extends** **Component**{

render() {

console.log(**this**.props);

**return** (

<table className="table table-bordered">

<thead>

<tr>

<td>字段</td>

<td>值</td>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>张三</td>

</tr>

</tbody>

</table>

)

}

}

## **6. 权限路由**

* umi 的权限路由是通过配置路由的 Routes 属性来实现。约定式的通过 yaml 注释添加，配置式的直接配上即可。
* PrivateRoute.js 的位置是相对于根目录的

### **6.1 profile.js**

/\*\*

\* title: 个人中心

\* Routes:

\* - ./PrivateRoute.js

\*/

### **6.2 PrivateRoute.js**

PrivateRoute.js

**import** { Route ,Redirect} **from** 'react-router-dom';

**export** **default** ({ render, ...others }) => {

**return** <Route

{...others}

render={props => localStorage.getItem('login')?render(props):<Redirect to={{pathname:'/login',state:{from:props.location.pathname}}}/>

}

/>;

}

### **6.3 pages/login.js**

pages/login.js

**import** React, { Component } **from** 'react'**import** Link **from** 'umi/link';**import** router **from** 'umi/router';**export** **default** **class** **componentName** **extends** **Component** {

login = ()=>{

localStorage.setItem('login','true');

**if**(**this**.props.location.state&&**this**.props.location.state.from){

router.push(**this**.props.location.state.from);

}

}

render() {

**return** (

<button onClick={this.login}>登录</button>

)

}

}

## **7. umi dev**

### **7.1 安装依赖**

cnpm i @babel/core @babel/plugin-proposal-**class**-**properties** @**babel**/**plugin**-**proposal**-**decorators** @**babel**/**preset**-**env** @**babel**/**preset**-**react** **babel**-**loader** **css**-**loader** **file**-**loader** **html**-**webpack**-**plugin** **react** **react**-**dom** **react**-**router**-**dom** **style**-**loader** **url**-**loader** **bootstrap**@3 **webpack** **webpack**-**cli** **webpack**-**dev**-**server** -**S**

### **7.2 webpack.config.js**

webpack.config.js

**const** path = require("path");**const** htmlWebpackPlugin = require('html-webpack-plugin');module.exports = {

context: process.cwd(),

entry: "./src/index.js",

output: {

path: path.resolve(\_\_dirname, "dist"),

filename: "bundle.js"

},

module: {

rules: [

{

test: /\.js$/,

use: {

loader: "babel-loader",

options: {

presets: ["@babel/preset-env","@babel/preset-react"],

plugins: [

["@babel/plugin-proposal-decorators", { legacy: true }],

["@babel/plugin-proposal-class-properties", { loose: true }]

]

}

}

},

{

test:/\.css$/,

use:[

"style-loader","css-loader"

]

}

]

},

plugins: [**new** htmlWebpackPlugin({

template:'./src/index.html'

})],

devServer: {

contentBase:path.resolve('dist')

}

};

### **7.3 index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Document</title>

</head>

<body>

<div id="root"></div>

</body>

</html>

### **7.4 index.js**

**import** React **from** 'react';**import** ReactDOM **from** 'react-dom';**import** Router **from** './router';

ReactDOM.render(<Router/>,document.getElementById('root'));

### **7.5 src\router.js**

**import** React **from** "react";**import** { HashRouter **as** Router, Route } **from** "react-router-dom";**let** routesConfig = require("./routesConfig");**function** **renderRoutes**(routesConfig) {

**return** routesConfig.map(({ path, exact = false, routes, component: RouteComponent, Routes: PrivateRoute }, index) => (

<Route key={index} path={path} exact={exact} render={

props => {

let render = props => (

<RouteComponent {...props}>

{routes && routes.length > 0 && renderRoutes(routes)}

</RouteComponent>

)

if (PrivateRoute) {

let privateRouteProps = { render, path, exact };

return <PrivateRoute {...privateRouteProps} />

} else {

return render(props);

}

}

} />

));

}**export** **default** props => <Router>{renderRoutes(routesConfig)}</Router>;

### **7.5 src\routes.config.js**

module.exports = [

{

"path": "/",

"component": require('../layouts/index.js').default,

"routes": [

{

"path": "/",

"exact": true,

"component": require('../pages/index.js').default

},

{

"path": "/login",

"exact": true,

"component": require('../pages/login.js').default

},

{

"path": "/profile",

"exact": true,

"component": require('../pages/profile.js').default,

"Routes": require('../PrivateRoute.js').default

}

]

}

];