link: null
title: 珠峰架构师成长计划
description: rollup.config.js
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=75 sentences=75, words=676

rollup实战

- rollupjs (https://rollupjs.org/guide/en/)是下一代ES模块捆绑器**1 背景**
- webpack打包非常繁琐，打包体积比较大
- rollup主要是用来打包JS库的
- vue/react/angular都在用rollup作为打包工具

2. 安装依赖

```
cnpm i @babel/core @babel/preset-env  @rollup/plugin-commonjs @rollup/plugin-node-resolve @rollup/plugin-typescript lodash rollup rollup-plugin-babel postcss
rollup-plugin-postcss rollup-plugin-terser tslib typescript rollup-plugin-serve rollup-plugin-livereload -D
```

3 初次使用

- Asynchronous Module Definition异步模块定义
- ES6 module是es6提出了新的模块化方案
- IIFE(Immediately Invoked Function Expression)即立即执行函数表达式，所谓立即执行，就是声明一个函数，声明完了立即执行
- UMD全称为 Universal Module Definition,也就是通用模块定义
- cjs是nodejs采用的模块化标准，commonjs使用方法 require来引入模块,这里 require()接收的参数是模块名或者是模块文件的路径

rollup.config.js

```
export default {
    input:'src/main.js',
    output:{
        file:'dist/bundle.cjs.js',
        format:'cjs',
        name:'bundleName'
    }
}
```

src\main.js

```
console.log('hello');
```

package.json

```
{
 "scripts": {
    "build": "rollup --config"
  },
}
```

dist\index.html

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>rolluptitle>
head>
<body>
    <script src="bundle.cjs.js">script>
body>
html>
```

4 支持babel

- 为了使用新的语法，可以使用babel来进行编译输出
- @babel/core是babel的核心包
- @babel/preset-env是预设
- @rollup/plugin-babel是babel插件

```
cnpm install @rollup/plugin-babel @babel/core @babel/preset-env --save-dev
```

```
let sum = (a,b)=>{
    return a+b;
}
let result = sum(1,2);
console.log(result);
```

.babelrc

```
{
    "presets": [
       [
        "@babel/env",
        {
            "modules":false
        }
       ]
    ]
}
```

rollup.config.js

```
+import babel from '@rollup/plugin-babel';
export default {
    input:'src/main.js',
    output:{
        file:'dist/bundle.cjs.js',//输出文件的路径和名称
        format:'cjs',//五种输出格式：amd/es6/iife/umd/cjs
        name:'bundleName'//当format为iife和umd时必须提供，将作为全局变量挂在window下
    },
+   plugins:[
+       babel({
+           exclude:"node_modules/**"
+       })
+   ]
}
```

5 tree-shaking

- Tree-shaking的本质是消除无用的js代码
- rollup只处理函数和顶层的import/export变量

src\main.js

```
import {name,age} from './msg';
console.log(name);
```

src\msg.js

```
export var name = 'zhufeng';
export var age = 12;
```

6 使用第三方模块

- rollup.js编译源码中的模块引用默认只支持 ES6+的模块方式 `import/export`

```
cnpm install @rollup/plugin-node-resolve @rollup/plugin-commonjs lodash  --save-dev
```

src\main.js

```
import _ from 'lodash';
console.log(_);
```

rollup.config.js

```
import babel from 'rollup-plugin-babel';
+import resolve from '@rollup/plugin-node-resolve';
+import commonjs from '@rollup/plugin-commonjs';
export default {
    input:'src/main.js',
    output:{
        file:'dist/bundle.cjs.js',//输出文件的路径和名称
        format:'cjs',//五种输出格式：amd/es6/iife/umd/cjs
        name:'bundleName'//当format为iife和umd时必须提供，将作为全局变量挂在window下
    },
    plugins:[
        babel({
            exclude:"node_modules/**"
        }),
+       resolve(),
+       commonjs()
    ]
}
```

7 使用CDN

src\main.js

```
import _ from 'lodash';
import $ from 'jquery';
console.log(_.concat([1,2,3],4,5));
console.log($);
export default 'main';
```

dist\index.html

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>rolluptitle>
head>
<body>
    <script src="https://cdn.jsdelivr.net/npm/lodash/lodash.min.js">script>
    <script src="https://cdn.jsdelivr.net/npm/jquery/jquery.min.js">
    <
body>
html>
```

rollup.config.js

```
import babel from 'rollup-plugin-babel';
import resolve from '@rollup/plugin-node-resolve';
import commonjs from '@rollup/plugin-commonjs';
export default {
    input:'src/main.js',
    output:{
        file:'dist/bundle.cjs.js',//输出文件的路径和名称
+       format:'iife',//五种输出格式：amd/es6/iife/umd/cjs
+       name:'bundleName',//当format为iife和umd时必须提供，将作为全局变量挂在window下
+       globals:{
+           lodash:'_', //告诉rollup全局变量_即是lodash
+           jquery:'{{content}}#x27; //告诉rollup全局变量$即是jquery
+       }
    },
    plugins:[
        babel({
            exclude:"node_modules/**"
        }),
        resolve(),
        commonjs()
    ],
+   external:['lodash','jquery']
}
```

8 使用typescript

```
cnpm install tslib typescript @rollup/plugin-typescript --save-dev
```

src\main.ts

```
let myName:string = 'zhufeng';
let age:number=12;
console.log(myName,age);
```

rollup.config.js

```
import babel from 'rollup-plugin-babel';
import resolve from '@rollup/plugin-node-resolve';
import commonjs from '@rollup/plugin-commonjs';
+import typescript from '@rollup/plugin-typescript';
export default {
+   input:'src/main.ts',
    output:{
        file:'dist/bundle.cjs.js',//输出文件的路径和名称
        format:'iife',//五种输出格式：amd/es6/iife/umd/cjs
        name:'bundleName',//当format为iife和umd时必须提供，将作为全局变量挂在window下
        globals:{
            lodash:'_', //告诉rollup全局变量_即是lodash
            jquery:'{{content}}#x27; //告诉rollup全局变量$即是jquery
        }
    },
    plugins:[
        babel({
            exclude:"node_modules/**"
        }),
        resolve(),
        commonjs(),
+       typescript()
    ],
    external:['lodash','jquery']
}
```

tsconfig.json

```
{
  "compilerOptions": {
    "target": "es5",
    "module": "ESNext",
    "strict": true,
    "skipLibCheck": true,
    "forceConsistentCasingInFileNames": true
  }
}
```

9 压缩JS

- terser是支持ES6 +的JavaScript压缩器工具包

```
cnpm install rollup-plugin-terser --save-dev
```

rollup.config.js

```
import babel from 'rollup-plugin-babel';
import resolve from '@rollup/plugin-node-resolve';
import commonjs from '@rollup/plugin-commonjs';
import typescript from '@rollup/plugin-typescript';
+import {terser} from 'rollup-plugin-terser';
export default {
    input:'src/main.ts',
    output:{
        file:'dist/bundle.cjs.js',//输出文件的路径和名称
        format:'iife',//五种输出格式：amd/es6/iife/umd/cjs
        name:'bundleName',//当format为iife和umd时必须提供，将作为全局变量挂在window下
        globals:{
            lodash:'_', //告诉rollup全局变量_即是lodash
            jquery:'{{content}}#x27; //告诉rollup全局变量$即是jquery
        }
    },
    plugins:[
        babel({
            exclude:"node_modules/**"
        }),
        resolve(),
        commonjs(),
        typescript(),
+       terser(),
    ],
    external:['lodash','jquery']
}
```

10 编译css

- terser是支持ES6 +的JavaScript压缩器工具包

```
cnpm install  postcss rollup-plugin-postcss --save-dev
```

rollup.config.js

```
import babel from 'rollup-plugin-babel';
import resolve from '@rollup/plugin-node-resolve';
import commonjs from '@rollup/plugin-commonjs';
import typescript from '@rollup/plugin-typescript';
import {terser} from 'rollup-plugin-terser';
+import postcss from 'rollup-plugin-postcss';
export default {
    input:'src/main.js',
    output:{
        file:'dist/bundle.cjs.js',//输出文件的路径和名称
        format:'iife',//五种输出格式：amd/es6/iife/umd/cjs
        name:'bundleName',//当format为iife和umd时必须提供，将作为全局变量挂在window下
        globals:{
            lodash:'_', //告诉rollup全局变量_即是lodash
            jquery:'{{content}}#x27; //告诉rollup全局变量$即是jquery
        }
    },
    plugins:[
        babel({
            exclude:"node_modules/**"
        }),
        resolve(),
        commonjs(),
        typescript(),
        //terser(),
+       postcss()
    ],
    external:['lodash','jquery']
}
```

src\main.js

```
import './main.css';
```

src\main.css

```
body{
    background-color: green;
}
```

11 本地服务器

```
cnpm install rollup-plugin-serve --save-dev
```

rollup.config.dev.js

```
import babel from 'rollup-plugin-babel';
import resolve from '@rollup/plugin-node-resolve';
import commonjs from '@rollup/plugin-commonjs';
import typescript from '@rollup/plugin-typescript';
import postcss from 'rollup-plugin-postcss';
+import serve from 'rollup-plugin-serve';
export default {
    input:'src/main.js',
    output:{
        file:'dist/bundle.cjs.js',//输出文件的路径和名称
        format:'iife',//五种输出格式：amd/es6/iife/umd/cjs
        name:'bundleName',//当format为iife和umd时必须提供，将作为全局变量挂在window下
        sourcemap:true,
        globals:{
            lodash:'_', //告诉rollup全局变量_即是lodash
            jquery:'{{content}}#x27; //告诉rollup全局变量$即是jquery
        }
    },
    plugins:[
        babel({
            exclude:"node_modules/**"
        }),
        resolve(),
        commonjs(),
        typescript(),
        postcss(),
+       serve({
+           open:true,
+           port:8080,
+           contentBase:'./dist'
+       })
    ],
    external:['lodash','jquery']
}
```

package.json

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