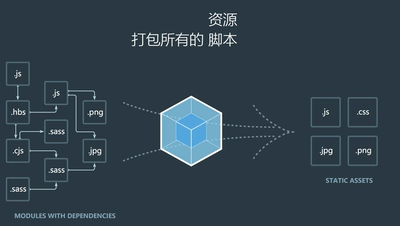
## 1. webpack介绍

* Webpack是一个前端资源加载/打包工具。它将根据模块的依赖关系进行静态分析，然后将这些模块按照指定的规则生成对应的静态资源。



## **2. 预备知识**

### **2.1 toStringTag**

* Symbol.toStringTag 是一个内置 symbol，它通常作为对象的属性键使用，对应的属性值应该为字符串类型，这个字符串用来表示该对象的自定义类型标签，通常只有内置的 Object.prototype.toString() 方法会去读取这个标签并把它包含在自己的返回值里。

console.log(Object.prototype.toString.call('foo')); // "[object String]"console.log(Object.prototype.toString.call([1, 2])); // "[object Array]"console.log(Object.prototype.toString.call(3)); // "[object Number]"console.log(Object.prototype.toString.call(true)); // "[object Boolean]"console.log(Object.prototype.toString.call(undefined)); // "[object Undefined]"console.log(Object.prototype.toString.call(null)); // "[object Null]"**let** myExports={};Object.defineProperty(myExports, Symbol.toStringTag, { value: 'Module' });console.log(Object.prototype.toString.call(myExports));

### **2.2 Object.create(null)**

* 使用create创建的对象，没有任何属性,把它当作一个非常纯净的map来使用，我们可以自己定义hasOwnProperty、toString方法,完全不必担心会将原型链上的同名方法覆盖掉
* 在我们使用for..in循环的时候会遍历对象原型链上的属性，使用create(null)就不必再对属性进行检查了

**var** ns = Object.create(null);**if** (**typeof** Object.create !== "function") {

Object.create = **function** (proto) {

**function** **F**() {}

F.prototype = proto;

**return** **new** F();

};

}console.log(ns)console.log(Object.getPrototypeOf(ns));

### **2.3 getter**

* [defineProperty](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Object/defineProperty) 方法会直接在一个对象上定义一个新属性，或者修改一个对象的现有属性， 并返回这个对象。
  + obj 要在其上定义属性的对象。
  + prop 要定义或修改的属性的名称。
  + descriptor 将被定义或修改的属性描述符。

### **2.3.1 描述符可同时具有的键值**

|  | **configurable** | **enumerable** | **value** | **writable** | **get** | **set** |
| --- | --- | --- | --- | --- | --- | --- |
| 数据描述符 | Yes | Yes | Yes | Yes | No | No |
| 存取描述符 | Yes | Yes No | No | Yes | Yes |  |

### **2.3.2 示例**

**var** ageValue = 10;**let** obj = {};Object.defineProperty(obj, "age", {

//value: 10,//数据描述符和存取描述符不能混合使用

**get**() {

**return** ageValue;

},

**set**(newValue) {

ageValue = newValue;

},

//writable: true,//是否可修改

enumerable: true,//是否可枚举

configurable: true//是否可配置可删除

});Object.defineProperty(obj, "name", {

value: 'zhufeng',//数据描述符和存取描述符不能混合使用

writable: true,//是否可修改

enumerable: true,//是否可枚举

configurable: true//是否可配置可删除

});console.log(obj.age, obj.name);

obj.age = 100;

obj.name = 'jiagou';console.log(obj.age, obj.name);

## **2. 同步加载**

### **2.1 webpack.config.js**

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

mode:'development',

devtool:"none",

context: process.cwd(),

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

output: {

filename: 'bundle.js'

},

devServer:{

contentBase:path.resolve(\_\_dirname,'./dist')

},

module: {

rules: [

{

test: /\.js$/,

use: {

loader: "babel-loader",

options: {

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

}

},

include: path.join(\_\_dirname, "src"),

exclude: /node\_modules/

}

]

},

plugins: []

};

### **2.2 index.js**

src\index.js

**let** title = require('./title.js');console.log(title);

### **2.3 title.js**

src\title.js

module.exports = "title";

### **2.4 打包文件分析**

(**function**(modules) {

// webpack的启动函数

//模块的缓存

**var** installedModules = {};

//定义在浏览器中使用的require方法

**function** **\_\_webpack\_require\_\_**(moduleId) {

//检查模块是否在缓存中

**if** (installedModules[moduleId]) {

**return** installedModules[moduleId].exports;

}

//创建一个新的模块并且放到模块的缓存中

**var** module = (installedModules[moduleId] = {

i: moduleId,

l: false,

exports: {}

});

//执行模块函数

modules[moduleId].call(

module.exports,

module,

module.exports,

\_\_webpack\_require\_\_

);

//把模块设置为已经加载

module.l = true;

//返回模块的导出对象

**return** module.exports;

}

//暴露出模块对象

\_\_webpack\_require\_\_.m = modules;

//暴露出模块缓存

\_\_webpack\_require\_\_.c = installedModules;

//为harmony导出定义getter函数

\_\_webpack\_require\_\_.d = **function**(exports, name, getter) {

**if** (!\_\_webpack\_require\_\_.o(exports, name)) {

Object.defineProperty(exports, name, { enumerable: true, get: getter });

}

};

//在导出对象上定义\_\_esModule属性

\_\_webpack\_require\_\_.r = **function**(exports) {

**if** (**typeof** Symbol !== "undefined" && Symbol.toStringTag) {

Object.defineProperty(exports, Symbol.toStringTag, { value: "Module" });

}

Object.defineProperty(exports, "\_\_esModule", { value: true });

};

/\*\*

\* 创建一个模拟的命名空间对象

\* mode & 1 value是模块ID直接用\_\_webpack\_require\_\_加载

\* mode & 2 把所有的属性合并到命名空间ns上

\* mode & 4 当已经是命名空间的时候(\_\_esModule=true)可以直接返回值

\* mode & 8|1 行为类似于require

\*/

\_\_webpack\_require\_\_.t = **function**(value, mode) {

**if** (mode & 1) value = \_\_webpack\_require\_\_(value);

**if** (mode & 8) **return** value;

**if** (mode & 4 && **typeof** value === "object" && value && value.\_\_esModule)

**return** value;

**var** ns = Object.create(null); //定义一个空对象

\_\_webpack\_require\_\_.r(ns);

Object.defineProperty(ns, "default", { enumerable: true, value: value });

**if** (mode & 2 && **typeof** value != "string")

**for** (**var** key **in** value)

\_\_webpack\_require\_\_.d(

ns,

key,

**function**(key) {

**return** value[key];

}.bind(null, key)

);

**return** ns;

};

// getDefaultExport函数为了兼容那些非non-harmony模块

\_\_webpack\_require\_\_.n = **function**(module) {

**var** getter =

module && module.\_\_esModule

? **function** **getDefault**() {

**return** module["default"];

}

: **function** **getModuleExports**() {

**return** module;

};

\_\_webpack\_require\_\_.d(getter, "a", getter);

**return** getter;

};

//判断对象身上是否拥有此属性

\_\_webpack\_require\_\_.o = **function**(object, property) {

**return** Object.prototype.hasOwnProperty.call(object, property);

};

//公共路径

\_\_webpack\_require\_\_.p = "";

//加载入口模块并且返回导出对象

**return** \_\_webpack\_require\_\_((\_\_webpack\_require\_\_.s = "./src/index.js"));

})({

"./src/index.js": **function**(module, exports, \_\_webpack\_require\_\_) {

**var** title = \_\_webpack\_require\_\_("./src/title.js");

console.log(title);

},

"./src/title.js": **function**(module, exports) {

module.exports = "title";

}

});

### **2.5 实现**

#### **2.5.1 webpack\_require**

* 实现require方法

(**function**(modules){

**var** installedModules = {};

**function** **\_\_webpack\_require\_\_**(moduleId){

**if**(installedModules[moduleId]){

**return** installedModules[moduleId];

}

**var** module = installedModules[moduleId] = {

i:moduleId,

l:false,

exports:{}

}

modules[moduleId].call(modules.exports,module,module.exports,\_\_webpack\_require\_\_);

module.l = true;

**return** module.exports;

}

**return** \_\_webpack\_require\_\_((\_\_webpack\_require\_\_.s = "./src/index.js"));

})({

"./src/index.js":**function**(module,exports,\_\_webpack\_require\_\_){

**var** title = \_\_webpack\_require\_\_('./src/title.js');

console.log(title);

},

"./src/title.js":**function**(module,exports){

module.exports = "title";

}

})

#### **2.5.2 d(defineProperty)**

* 通过getter的方式增加属性

**function** **d**(exports, name, getter) {

**if** (!exports.hasOwnProperty(name)) {

Object.defineProperty(exports, name, { get: getter });

}

}

**let** obj = {};**function** **ageGetter**() {

**return** 10;

}

d(obj, 'age', ageGetter);console.log(obj.age);

#### **2.5.3 r**

* 表示此对象是一个ES6模块对象

**function** **r**(exports) {

Object.defineProperty(exports, Symbol.toStringTag, { value: 'Module' });

Object.defineProperty(exports, '\_\_esModule', { enumerable: true, value: true });

};**let** obj2 = {};

r(obj2);console.log(obj2);

#### **2.5.4 n**

* 获取此对象的默认导出

**function** **n**(module) {

**var** getter = module.\_\_esModule ?

**function** () { **return** module['default']; } :

**function** () { **return** module; };

d(getter, 'a', getter);

**return** getter;

};**let** m = { age: 10 };**let** getter = n(m);console.log(getter.a);**let** m2 = { default: { age: 20 }, \_\_esModule: true };**let** getter2 = n(m2);console.log(getter2.a);

#### **2.5.5 t**

* 把任意模块包装成ES6模块
* mode & 1: value is a module id, require it 表示传的是模块ID
* mode & 2: merge all properties of value into the ns 需要合并属性
* mode & 4: return value when already ns object 如果是ES6模块直接返回
* mode & 8|1: behave like require 等同于require方法

**let** modules = {

//相当于require

'moduleA': **function** (module, exports) {

module.exports = 'moduleA导出内容';

},

//是ES6模块直接返回

'moduleB': **function** (module, exports) {

module.exports = { \_\_esModule: true, default: 'moduleB导出内容' };

},

//封装成ES6模块

'moduleC': **function** (module, exports) {

module.exports = { name: 'moduleC导出内容' };

}

}**function** **\_\_webpack\_require\_\_**(moduleId) {

**var** module = {

i: moduleId,

l: false,

exports: {}

}

modules[moduleId].call(modules.exports, module, module.exports, \_\_webpack\_require\_\_);

module.l = true;

**return** module.exports;

}**function** **t**(value, mode) {

**if** (mode & 1) value = \_\_webpack\_require\_\_(value);

**if** (mode & 8) **return** value;

**if** ((mode & 4) && **typeof** value === 'object' && value && value.\_\_esModule) **return** value;

**var** ns = Object.create(null);

Object.defineProperty(ns, Symbol.toStringTag, { value: 'Module' });

Object.defineProperty(ns, '\_\_esModule', { enumerable: true, value: true });

Object.defineProperty(ns, 'default', { enumerable: true, value: value });

**if** (mode & 2 && **typeof** value != 'string')

**for** (**var** key **in** value)

Object.defineProperty(ns, key, { enumerable: true, get: () => value[key] });

**return** ns;

};**let** result1 = t('moduleA', 0b1001);console.log(result1);**let** result2 = t('moduleB', 0b0101);console.log(result2);**let** result3 = t('moduleC', 0b0011);console.log(result3, result3.name);

## **3. harmony**

### **3.1 common.js加载 common.js**

#### **3.1.1 index.js**

**let** title = require('./title');console.log(title.name);console.log(title.age);

#### **3.1.2 title.js**

exports.name = 'title\_name';

exports.age = 'title\_age';

#### **3.1.3 bundle.js**

{"./src/index.js":

(**function**(module, exports, \_\_webpack\_require\_\_) {

**var** title = \_\_webpack\_require\_\_("./src/title.js");

console.log(title.name);

console.log(title.age);

}),"./src/title.js":

(**function**(module, exports) {

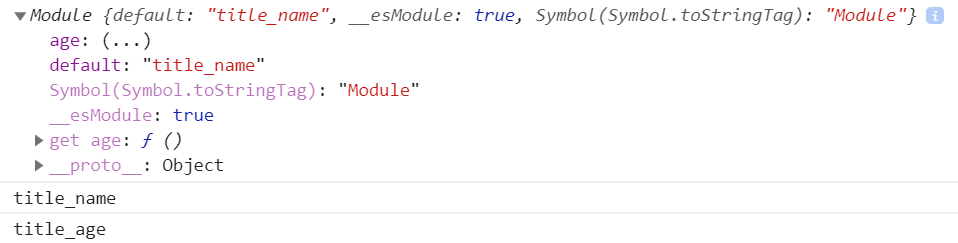
exports.name = 'title\_name';

exports.age = 'title\_age';

})

}

### **3.2 common.js加载 ES6 modules**



#### **3.2.1 index.js**

**let** title = require('./title');console.log(title);console.log(title.default);console.log(title.age);

#### **3.2.2 title.js**

**export** **default** name = 'title\_name';**export** **const** age = 'title\_age';

#### **3.2.3 bundle.js**

{

"./src/index.js":

(**function** (module, exports, \_\_webpack\_require\_\_) {

**var** title = \_\_webpack\_require\_\_(/\*! ./title \*/ "./src/title.js");

console.log(title.name);

console.log(title.age);

}),

"./src/title.js":

(**function** (module, \_\_webpack\_exports\_\_, \_\_webpack\_require\_\_) {

\_\_webpack\_require\_\_.r(\_\_webpack\_exports\_\_);

/\* harmony export (binding) \*/

\_\_webpack\_require\_\_.d(\_\_webpack\_exports\_\_, "age", **function** () { **return** age; });

/\* harmony default export \*/

\_\_webpack\_exports\_\_["default"] = (name = 'title\_name');

**var** age = 'title\_age';

})

}

### **3.3 ES6 modules 加载 ES6 modules**

#### **3.3.1 index.js**

**import** name,{age} **from** './title';console.log(name);console.log(age);

#### **3.3.2 title.js**

**export** **default** name = 'title\_name';**export** **const** age = 'title\_age';

#### **3.3.3 bundle.js**

{

"./src/index.js":

(**function** (module, \_\_webpack\_exports\_\_, \_\_webpack\_require\_\_) {

"use strict";

\_\_webpack\_require\_\_.r(\_\_webpack\_exports\_\_);

**var** \_title\_\_WEBPACK\_IMPORTED\_MODULE\_0\_\_ = \_\_webpack\_require\_\_(/\*! ./title \*/ "./src/title.js");

console.log(\_title\_\_WEBPACK\_IMPORTED\_MODULE\_0\_\_["default"]);

console.log(\_title\_\_WEBPACK\_IMPORTED\_MODULE\_0\_\_["age"]);

}),

"./src/title.js":

(**function** (module, \_\_webpack\_exports\_\_, \_\_webpack\_require\_\_) {

\_\_webpack\_require\_\_.r(\_\_webpack\_exports\_\_);

\_\_webpack\_require\_\_.d(\_\_webpack\_exports\_\_, "age", **function** () { **return** age; });

\_\_webpack\_exports\_\_["default"] = (name = 'title\_name');

**var** age = 'title\_age';

})

}

### **3.4 ES6 modules 加载 common.js**

#### **3.4.1 index.js**

**import** home, { name, age } **from** './title';console.log(name);console.log(age);console.log(home);

#### **3.4.2 title.js**

module.exports = { home: 'beijing' };module.exports.name = 'title\_name';module.exports.age = 'title\_age';

#### **3.4.3 bundle.js**

{

"./src/index.js":

(**function** (module, \_\_webpack\_exports\_\_, \_\_webpack\_require\_\_) {

\_\_webpack\_require\_\_.r(\_\_webpack\_exports\_\_);

/\* harmony import \*/ **var** \_title\_\_WEBPACK\_IMPORTED\_MODULE\_0\_\_ = \_\_webpack\_require\_\_(/\*! ./title \*/ "./src/title.js");

/\* harmony import \*/ **var** \_title\_\_WEBPACK\_IMPORTED\_MODULE\_0\_\_\_default = /\*#\_\_PURE\_\_\*/\_\_webpack\_require\_\_.n(\_title\_\_WEBPACK\_IMPORTED\_MODULE\_0\_\_);

console.log(\_title\_\_WEBPACK\_IMPORTED\_MODULE\_0\_\_["name"]);

console.log(\_title\_\_WEBPACK\_IMPORTED\_MODULE\_0\_\_["age"]);

console.log(\_title\_\_WEBPACK\_IMPORTED\_MODULE\_0\_\_\_default.a);

}),

"./src/title.js":

(**function** (module, exports) {

module.exports = {

home: 'beijing'

};

module.exports.name = 'title\_name';

module.exports.age = 'title\_age';

})

}

## **4. 异步加载**

### **4.1 源文件**

#### **4.1.1 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></body></html>

#### **4.1.2 src\index.js**

src\index.js

**let** button = document.createElement("button");

button.innerHTML = "点我";

button.onclick = **function** () {

**import**(/\*webpackChunkName: 'title'\*/'./title.js').then(**function** (result) {

console.log(result, result.default);

});

};document.body.appendChild(button);

#### **4.1.3 src\title.js**

src\title.js

module.exports = {

name: 'title'

}module.exports.age = 10;

### **4.2 打包后的文件**

#### **4.2.1**

(**function** (modules) { // 启动函数

//安装一个为了加载额外代码块的JSON回调函数

**function** **webpackJsonpCallback**(data) {

**var** chunkIds = data[0];//代码块ID

**var** moreModules = data[1];//更多的模块

//向模块对象上增加更多的模块，然后把所有的chunkIds设置为已经加载并触发回调

**var** moduleId, chunkId, i = 0, resolves = [];

**for** (; i < chunkIds.length; i++) {

chunkId = chunkIds[i];

**if** (installedChunks[chunkId]) {

resolves.push(installedChunks[chunkId][0]);

}

installedChunks[chunkId] = 0;//标识这个代码块为已经OK

}

**for** (moduleId **in** moreModules) {//把新拉下来的模块合并到模块对象上

**if** (Object.prototype.hasOwnProperty.call(moreModules, moduleId)) {

modules[moduleId] = moreModules[moduleId];

}

}

**if** (parentJsonpFunction) parentJsonpFunction(data);//如果有父JSONP函数就调用

**while** (resolves.length) {

resolves.shift()();//让所有的promise都OK

}

};

// 模块缓存

**var** installedModules = {};

//用来存放加载完成或加载中的代码块对象

// undefined = 代码块未加载, null = 代码块正在预加载或者预获取

// Promise = 代码块更在加载中, 0 = 代码块已经加载

**var** installedChunks = {

"main": 0

};

//JSON加载的路径

**function** **jsonpScriptSrc**(chunkId) {

**return** \_\_webpack\_require\_\_.p + "" + chunkId + ".bundle.js"

}

**function** **\_\_webpack\_require\_\_**(moduleId) {

**if** (installedModules[moduleId]) {

**return** installedModules[moduleId].exports;

}

**var** module = installedModules[moduleId] = {

i: moduleId,

l: false,

exports: {}

};

modules[moduleId].call(module.exports, module, module.exports, \_\_webpack\_require\_\_);

module.l = true;

**return** module.exports;

}

//这个文件只包含入口代码块

//用来加载额外的代码块的函数

\_\_webpack\_require\_\_.e = **function** **requireEnsure**(chunkId) {

**var** promises = [];

//JSONP代码块加载

**var** installedChunkData = installedChunks[chunkId];

**if** (installedChunkData !== 0) { // 0的意思是已经安装

// a Promise means "currently loading". 如果是一个Promise的话表示正在加载

**if** (installedChunkData) {

promises.push(installedChunkData[2]);//如果已经在加载中了，则添加Promise

} **else** {

//在代码块缓存中放置Promise

**var** promise = **new** Promise(**function** (resolve, reject) {

installedChunkData = installedChunks[chunkId] = [resolve, reject];

});

promises.push(installedChunkData[2] = promise);

// 开始加载代码块

**var** script = document.createElement('script');

**var** onScriptComplete;

script.charset = 'utf-8';

script.timeout = 120;

//// HTMLElement 接口的 nonce 属性返回只使用一次的加密数字，被内容安全政策用来决定这次请求是否被允许处理。

**if** (\_\_webpack\_require\_\_.nc) {

script.setAttribute("nonce", \_\_webpack\_require\_\_.nc);

}

//设置源文件路径

script.src = jsonpScriptSrc(chunkId);

//在栈展开之前创建错误以获取有用的堆栈信息

**var** error = **new** Error();

onScriptComplete = **function** (event) {

script.onerror = script.onload = null;

clearTimeout(timeout);

**var** chunk = installedChunks[chunkId];

**if** (chunk !== 0) {

**if** (chunk) {

**var** errorType = event && (event.type === 'load' ? 'missing' : event.type);

**var** realSrc = event && event.target && event.target.src;

error.message = 'Loading chunk ' + chunkId + ' failed.\n(' + errorType + ': ' + realSrc + ')';

error.name = 'ChunkLoadError';

error.type = errorType;

error.request = realSrc;

chunk[1](error);

}

installedChunks[chunkId] = undefined;

}

};

**var** timeout = setTimeout(**function** () {

onScriptComplete({ type: 'timeout', target: script });

}, 120000);

script.onerror = script.onload = onScriptComplete;

document.head.appendChild(script);

}

}

**return** Promise.all(promises);

};

\_\_webpack\_require\_\_.m = modules;

\_\_webpack\_require\_\_.c = installedModules;

\_\_webpack\_require\_\_.d = **function** (exports, name, getter) {

**if** (!\_\_webpack\_require\_\_.o(exports, name)) {

Object.defineProperty(exports, name, { enumerable: true, get: getter });

}

};

\_\_webpack\_require\_\_.r = **function** (exports) {

**if** (**typeof** Symbol !== 'undefined' && Symbol.toStringTag) {

Object.defineProperty(exports, Symbol.toStringTag, { value: 'Module' });

}

Object.defineProperty(exports, '\_\_esModule', { value: true });

};

\_\_webpack\_require\_\_.t = **function** (value, mode) {

**if** (mode & 1) value = \_\_webpack\_require\_\_(value);

**if** (mode & 8) **return** value;

**if** ((mode & 4) && **typeof** value === 'object' && value && value.\_\_esModule) **return** value;

**var** ns = Object.create(null);

\_\_webpack\_require\_\_.r(ns);

Object.defineProperty(ns, 'default', { enumerable: true, value: value });

**if** (mode & 2 && **typeof** value != 'string') **for** (**var** key **in** value) \_\_webpack\_require\_\_.d(ns, key, **function** (key) { **return** value[key]; }.bind(null, key));

**return** ns;

};

\_\_webpack\_require\_\_.n = **function** (module) {

**var** getter = module && module.\_\_esModule ?

**function** **getDefault**() { **return** module['default']; } :

**function** **getModuleExports**() { **return** module; };

\_\_webpack\_require\_\_.d(getter, 'a', getter);

**return** getter;

};

\_\_webpack\_require\_\_.o = **function** (object, property) { **return** Object.prototype.hasOwnProperty.call(object, property); };

\_\_webpack\_require\_\_.p = "";

//异步加载中的错误处理函数

\_\_webpack\_require\_\_.oe = **function** (err) { console.error(err); **throw** err; };

//刚开始的时候会把数组赋给window["webpackJsonp"],并且赋给jsonpArray

**var** jsonpArray = window["webpackJsonp"] = window["webpackJsonp"] || [];

//绑定push函数为oldJsonpFunction

**var** oldJsonpFunction = jsonpArray.push.bind(jsonpArray);

//狸猫换太子，把webpackJsonpCallback赋给了jsonpArray.push方法

jsonpArray.push = webpackJsonpCallback;

//把数组进行截取得到一个新的数组

jsonpArray = jsonpArray.slice();

//如果数组不为空，就把全部安装一次

**for** (**var** i = 0; i < jsonpArray.length; i++) webpackJsonpCallback(jsonpArray[i]);

//把oldJsonpFunction赋给parentJsonpFunction

**var** parentJsonpFunction = oldJsonpFunction;

**return** \_\_webpack\_require\_\_(\_\_webpack\_require\_\_.s = "./src/index.js");

})

({

"./src/index.js":

(**function** (module, exports, \_\_webpack\_require\_\_) {

**var** button = document.createElement("button");

button.innerHTML = "点我";

button.onclick = **function** () {

\_\_webpack\_require\_\_.e("title").then(\_\_webpack\_require\_\_.t.bind(null, "./src/title.js", 7)).then(**function** (result) {

console.log(result["default"]);

});

};

document.body.appendChild(button);

})

});

#### **4.2.2 title.bundle.js**

dist\title.bundle.js

(window["webpackJsonp"] = window["webpackJsonp"] || []).push([["title"], {

"./src/title.js":

(**function** (module, exports) {

module.exports = {

name: 'title'

};

module.exports.age = 10;

})

}]);

### **4.3 实现bundle.js**

(**function** (modules) {

**var** installedModules = {};

**function** **\_\_webpack\_require\_\_**(moduleId) {

**if** (installedModules[moduleId]) {

**return** installedModules[moduleId].exports;

}

**var** module = installedModules[moduleId] = {

i: moduleId,

l: false,

exports: {}

}

modules[moduleId].call(modules.exports, module, module.exports, \_\_webpack\_require\_\_);

module.l = true;

**return** module.exports;

}

**var** installedChunks = {

"main": 0

};

**function** **webpackJsonpCallback**(data) {

**var** chunkIds = data[0];//代码块ID

**var** moreModules = data[1];//更多的模块

**var** moduleId, chunkId, resolves = [];

**for** (**let** i = 0; i < chunkIds.length; i++) {

chunkId = chunkIds[i];

resolves.push(installedChunks[chunkId][0]);

installedChunks[chunkId] = 0;//标识这个代码块为已经OK

}

**for** (moduleId **in** moreModules) {//把新拉下来的模块合并到模块对象上

modules[moduleId] = moreModules[moduleId];

}

**while** (resolves.length) {

resolves.shift()();//让所有的promise都OK

}

}

\_\_webpack\_require\_\_.t = **function** (value) {

value = \_\_webpack\_require\_\_(value);

**var** ns = Object.create(null);

ns.default = value;

**return** ns;

};

\_\_webpack\_require\_\_.e = **function** (chunkId) {

**var** promises = [];

**var** installedChunkData = installedChunks[chunkId];

**if** (installedChunkData !== 0) {

**var** promise = **new** Promise(**function** (resolve, reject) {

installedChunkData = installedChunks[chunkId] = [resolve, reject];

});

installedChunkData[2] = promise;

promises.push(promise);

**var** script = document.createElement('script');

script.src = chunkId + ".bundle.js";

document.head.appendChild(script);

}

**return** Promise.all(promises);

}

**var** jsonpArray = window["webpackJsonp"] = window["webpackJsonp"] || [];

jsonpArray.push = webpackJsonpCallback;

**return** \_\_webpack\_require\_\_("./src/index.js");

})({

"./src/index.js":

(**function** (module, exports, \_\_webpack\_require\_\_) {

**var** button = document.createElement("button");

button.innerHTML = "点我";

button.onclick = **function** () {

\_\_webpack\_require\_\_.e("title").then(\_\_webpack\_require\_\_.t.bind(null, "./src/title.js", 7)).then(**function** (result) {

console.log(result["default"]);

});

};

document.body.appendChild(button);

})

})