link: null
title: 珠峰架构师成长计划
description: null
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=136 sentences=412, words=2692

---

1. 动机

- Module Federation (https://webpackjs.org/concepts/module-federation/)的动机是与多个团队一起开发一个或多个应用程序
- 应用程序分为较小的应用程序部分,这些可能是前端组件，例如Header或Sidebar组件,也可能是逻辑组件，例如 Data Fetching Logic或其他业务逻辑
- 每个部分都可以由独立的团队开发
- 应用程序或其一部分共享其他部分或库

2. 模块联邦

- 使用模块联邦，每个部分将是一个单独的构建， 这些构建被编译为 容器
- 容器可以被应用程序或其他容器引用
- 在这种关系中，容器是 远程的，容器的使用者是 主机
- 远程可以将模块公开给 主机
- 主机可以使用此类模块,它们被称为 远程模块
- 通过使用单独的构建，我们可以获得整个系统的良好构建性能

3. 模块联邦概述

- 这是模块联邦的概述
- 这里显示了模块联邦的两个方面： 公开的模块和 共享的模块
- 容器以 异步方式公开模块
- 您需要先让容器'加载要使用的模块，然后再从容器中使用它们
- 允许构建将每个公开的模块及其依赖项放在单独的文件中
- 这样，只需要加载使用过的模块，但是容器仍然可以将模块捆绑在一起
- 另外，这里使用webpack的代码分割技术(例如，在公开的模块中分割第三方模块或公共依赖模块的代码块)
- 这使我们可以保持较低的请求和总下载量，从而获得良好的Web性能
- 容器的使用者需要能够处理暴露模块的异步加载
- 共享模块的另一个方面也显示在这里。每一个部分，容器和应用程序都可以将共享模块与版本信息一起放入共享范围
- 他们还能够使用共享范围中的共享模块以及版本要求检查
- 共享范围将对共享模块进行重复数据删除，该方式可为各方提供版本要求内的共享模块的最高可用版本
- 还以异步方式提供和使用共享模块。因此，提供共享模块没有下载成本。仅下载使用/消耗的共享模块

4. 案例

4.1 总览

- 这里显示了一个构建应用程序。
- HomePage（来自团队A）使用 Dropdown组件()来自团队B)
- HomePage上的 Login链接按需加载 LoginModal(来自团队A), LoginModal使用 Button组件(来自团队B)
- 两个团队的几乎所有模块都使用 react
- 让我们将 Module Federation应用于此应用程序...

4.2 项目加载

- 现在，webpack开始...
- webpack将自动为容器生成一个 容器入口，生成的模块将包含对所有公开和共享模块以及如何加载它们的引用
- 每个公开的模块与依赖项一起放入单独的文件中
- 每个共享模块也放入单独的文件中
- 从容器加载 Button时，它将仅加载按钮块和 react,加载 Dropdown时，它将仅加载 Dropdown和 react
- 当加载 Dropdown时，但是另一方提供了另一个react版本(可能更高),它将加载 Dropdown和另一方提供的react版本的块(实际上，它将加载操作委托给另一方)

4.3 teamB

- 从B团队的角度来看,B团队只关心其组件
- 团队B想建立一个 容器，因此标记其某些模块。 Button和 Dropdown被标记为 已公开，因为它们应由其他团队使用
- react被标记为 共享,以便可以与其他团队共享

4.4 teamA

- 这是团队A消耗团队B的容器的方式
- 它们在运行时引用容器，并且将从容器中加载模块（在运行时)
- 一个示例是 Login链接，该链接打开 LoginModal.单击链接时，将并行下载 LoginModal的代码和 Button的代码
- 共享模块也会发生类似的情况

5. 项目配置

- 有一个 ModuleFederationPlugin可以使用模块联邦。使用不同的属性来设置不同的部分
- 要创建容器， 暴露属性是重要的
    - 在此指定了容器的使用者应可访问的所有模块
    - 可以给他们提供一个公共名称，该名称是使用者必须使用的名称，并将其指向他们自己的代码库中的一个模块（内部请求）。支持任何模块，可能是javascript，typescript，CSS等可以在您的代码库中处理的任何模块
- 他们使用其他容器， remotes属性是goto属性。 它是一个对象，其中的所有容器都应在当前版本中可用。关键是模块作用域，在该作用域中，应在自己的代码库中访问容器公开的模块。任何以此键开头的请求都将创建一个远程模块，该模块将在运行时加载。该值是容器的位置。默认情况下，脚本外部变量用作容器位置。这里将指定脚本文件的URL和全局文件。该脚本将在运行时加载，并且可以从全局访问容器。
- 要在任何一侧共享模块，应使用共享属性。对于简单的情况，可以提供模块说明符列表，这些说明符将这些模块（在代码库中使用时）标记为共享模块。它们将以当前安装的版本提供，并以使用包的

package.json中指定的版本范围使用。
- 所有属性还支持高级配置选项。一个值得注意的高级选项是共享模块的 singleton：true选项。确保在运行时仅创建模块的单个实例。对于某些库，例如。不喜欢在同一应用程序中多次实例化的react。在这种情况下，无效的版本范围只会在运行时导致警告。
- 更高级的选项允许覆盖或禁用自动推断的值，例如版本， requiredVersion或文件名，并允许使用库和外部的不同方式。例如用于在Node.js中使用或选择加入更严格的版本检查（当版本范围无效时，这会导致错误而不是警告）

1.初始化项目

1.创建项目

```
mkdir teama
cd teama
npm init -y

mkdir teamb
cd teamb
npm init -y
```

2.安装依赖

```
cd teama
cd teamb
npm install webpack webpack-cli webpack-dev-server html-webpack-plugin --save-dev
npm install is-array --save
```

2. 配置项目

配置参数

字段 类型 含义 name string 项目名称，应用的身份证，在应用分享资源的时候使用的标识,被远程引用时路径为 ${name}/${expose}

library object 暴露项目的全局变量名 格式为 {type:'var',name:projectName} filename string 构建后的文件名，也是远程引入的文件名 remotes object 远程引用的应用名及其别名的映射，格式为 {远程项目别名:远程引入的项目名(其它应用name字段)} exposes object 被远程引用时可暴露的资源路径及其别名,格式为{别名:组件的路径} shared object 与其他应用之间可以共享的第三方依赖,可以在此控制版本号 title string head的标题 files object 远程调用项目的文件链接

2.1 teamb

2.1.1 Button.js

teamb\src\Button.js

```
import isArray from 'is-array';
export default `(Button[${isArray.name}])`;
```

2.1.2 Dropdown.js

teamb\src\Dropdown.js

```
import isArray from 'is-array';
import ArrowIcon from './ArrowIcon';
export default `(Dropdown[${ArrowIcon}][${isArray.name}])`;
```

2.1.3 ArrowIcon.js

teamb\src\ArrowIcon.js

```
export default  'ArrowIcon';
```

2.1.4 bootstrap.js

teamb\src\bootstrap.js

```
import isArray from 'is-array';
let Dropdown = await import('./Dropdown');
let Button = await import('./Button');
console.log(Dropdown.default);
console.log(Button.default);
console.log(isArray.name);
```

2.1.5 index.js

teamb\src\index.js

```
import("./bootstrap");
```

2.1.4 teamb\webpack.config.js

- proposal-top-level-await (https://github.com/tc39/proposal-top-level-await) teamb\webpack.config.js

```
let HtmlWebpackPlugin = require("html-webpack-plugin");
const ModuleFederationPlugin = require("webpack/lib/container/ModuleFederationPlugin");
module.exports = {
  mode: "development",
  devtool: false,
  devServer: {
      port: 8000
  },
  plugins: [
      new HtmlWebpackPlugin({
          template: './public/index.html'
      }),
      new ModuleFederationPlugin({
          name: "teamb",
          filename: "remoteEntry.js",
          library: { type: 'var', name: 'teambVar' },
          exposes: {
              "./Dropdown": "./src/Dropdown.js",
              "./Button": "./src/Button.js",
          },
          shared: ["is-array"]
      })
  ],
  target: ['es6', 'web'],
  experiments: {
      topLevelAwait: true
  },
}
```

2.1.5 teamb\public\index.html

teamb\public\index.html

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

2.1.6 package.json

```
{
  "scripts": {
    "build": "webpack",
    "start": "webpack serve"
  },
}
```

2.1.7 启动

```
npm run build
npm run start
```

2.2 teama

2.2.1 LoginModal.js

teama\src\LoginModal.js

```
import isArray from 'is-array';
let Button = await import('teamb/Button');
export default  `(LoginModal[${Button.default}][${isArray.name}])`;
```

2.2.2 HomePage.js

teama\src\HomePage.js

```
import isArray from 'is-array';
let Dropdown = await import('teamb/Dropdown');
let LoginModal = await import('./LoginModal');
export default `(HomePage[${Dropdown.default}][${LoginModal.default}][${isArray.name}])`
```

2.2.3 bootstrap.js

teama\src\bootstrap.js

```
import HomePage from './HomePage';
console.log(HomePage);
```

2.2.4 index.js

teama\src\index.js

```
import("./bootstrap");
```

2.2.5 webpack.config.js

teama\webpack.config.js

```
let HtmlWebpackPlugin = require("html-webpack-plugin");
const ModuleFederationPlugin = require("webpack/lib/container/ModuleFederationPlugin");
module.exports = {
    mode: "development",
    devtool: false,
    output: {
        publicPath: "http://localhost:3000/"
    },
    target: ['es6', 'web'],
    experiments: {
        topLevelAwait: true
    },
    devServer: {
        port: 3000
    },
    plugins: [
        new HtmlWebpackPlugin({
            template: './public/index.html'
        }),
        new ModuleFederationPlugin({
            filename: "remoteEntry.js",
            name: "teama",
            library: { type: 'var', name: 'teamaVar' },
            remotes: {
                teamb: "teambVar@http://localhost:8000/remoteEntry.js"
            },
            shared: ["is-array"]
        })
    ]
}
```

3. 运行原理

- 下载并执行 remoteEntry.js，挂载入口点对象到 window.teamb，他有两个函数属性， init 和 get
- init方法用于初始化作用域对象， get 方法用于下载 moduleMap 中导出的远程模块。
- 加载 teamb 到本地模块
- 创建 teamb.init 的执行环境，收集依赖到共享作用域对象 shareScope
- 执行 teamb.init，初始化作用域对象
- 用户 import 远程模块时调用 teamb.get(moduleName) 通过 JSONP 懒加载远程模块，然后缓存在全局对象 window[‘webpackChunk’ + appName]
- 通过 webpack_require方法读取缓存中的模块，执行用户回调

3.1 teamb

3.1.1 remoteEntry.js

```
window.teamb = (() => {
    var modules = ({
        "webpack/container/entry/teamb":
            ((module, exports, require) => {
                var moduleMap = {
                    "./Dropdown": () => {
                        return Promise.all([require.e("webpack_sharing_consume_default_is-array_is-array"), require.e("src_Dropdown_js")]).then(() => () =>
require("./src/Dropdown.js"));
                    },
                    "./Button": () => {
                        return Promise.all([require.e("webpack_sharing_consume_default_is-array_is-array"), require.e("src_Button_js")]).then(() => () =>
require("./src/Button.js"));
                    }
                };
                var get = (module) => {
                    return moduleMap[module]();
                };

                var init = (shareScope) => {
                    var name = "default";
                    require.S[name] = shareScope;
                    return require.I(name);
                };
                require.d(exports, {
                    get: () => get,
                    init: () => init
                });
            })
    });
    var cache = {};
    function require(moduleId) {
        if (cache[moduleId]) {
            return cache[moduleId].exports;
        }
        var module = cache[moduleId] = {
            exports: {}
        };
        modules[moduleId](module, module.exports, require);
        return module.exports;
    }
    require.n = (module) => {
        var getter = module && module.__esModule ?
            () => module['default'] :
            () => module;
        return getter;
    };
    require.d = (exports, definition) => {
        for (var key in definition) {
            Object.defineProperty(exports, key, { get: definition[key] });
        }
    };
    require.f = {};
    require.e = (chunkId) => {
        return Promise.all(Object.keys(require.f).reduce((promises, key) => {
            require.f[key](chunkId, promises);
            return promises;
```

```
        }, []));
    };
    require.u = (chunkId) => {
        return "" + chunkId + ".js";
    };
    require.o = (obj, prop) => Object.prototype.hasOwnProperty.call(obj, prop)
    require.l = (url, done) => {
        var script = document.createElement('script');
        script.src = url;
        script.onload = done
        document.head.appendChild(script);
    };
    require.r = (exports) => {
        Object.defineProperty(exports, Symbol.toStringTag, { value: 'Module' });
        Object.defineProperty(exports, '__esModule', { value: true });
    };
    require.S = {};
    require.I = (name) => {
        if (require.S[name])
            return Promise.resolve();
    };
    require.p = "http://localhost:8000/";

    var init = (fn) => function (scopeName, key, version) {

        return require.I(scopeName).then(() => {

            return fn(require.S[scopeName], key, version);
        });
    };

    var loadShareScope = init((scope, key, version) => {

        var versions = scope[key];

        var entry = versions[version];
        return entry.get()
    });
    (() => {
        var moduleToHandlerMapping = {
            "webpack/sharing/consume/default/is-array/is-array": () => loadShareScope("default", "is-array", '1.0.1')
        };
        var chunkMapping = {
            "webpack_sharing_consume_default_is-array_is-array": [
                "webpack/sharing/consume/default/is-array/is-array"
            ]
        };
        require.f.consumes = (chunkId, promises) => {
            if (require.o(chunkMapping, chunkId)) {
                chunkMapping[chunkId].forEach((id) => {
                    let promise = moduleToHandlerMapping[id]().then((factory) => {
                        modules[id] = (module) => {
                            module.exports = factory();
                        }
                    })
                    promises.push(promise);
                });
            }
        }
    })();
    var installedChunks = {
        "teamb": 0
    };
    require.f.j = (chunkId, promises) => {
        if ("webpack_sharing_consume_default_is-array_is-array" != chunkId) {
            var promise = new Promise((resolve, reject) => {
                installedChunks[chunkId] = [resolve, reject];
            });
            promises.push(promise);
            var url = require.p + require.u(chunkId);
            require.l(url);
        }
    };
    var webpackJsonpCallback = (parentChunkLoadingFunction, data) => {
        var [chunkIds, moreModules, runtime] = data;
        var moduleId, chunkId, i = 0, resolves = [];
        for (; i < chunkIds.length; i++) {
            chunkId = chunkIds[i];
            if (require.o(installedChunks, chunkId) && installedChunks[chunkId]) {
                resolves.push(installedChunks[chunkId][0]);
            }
            installedChunks[chunkId] = 0;
        }
        for (moduleId in moreModules) {
            if (require.o(moreModules, moduleId)) {
                modules[moduleId] = moreModules[moduleId];
            }
        }
        if (runtime) runtime(require);
        if (parentChunkLoadingFunction) parentChunkLoadingFunction(data);
        while (resolves.length) {
            resolves.shift()();
        }
    }
    var chunkLoadingGlobal = self["webpackChunkteamb"] = self["webpackChunkteamb"] || [];
    chunkLoadingGlobal.forEach(webpackJsonpCallback.bind(null, 0));
    chunkLoadingGlobal.push = webpackJsonpCallback.bind(null, chunkLoadingGlobal.push.bind(chunkLoadingGlobal));
    return require("webpack/container/entry/teamb");
})();
```

3.1.2 src_Dropdown_js.js

teamb\dist\src_Dropdown_js.js

```
(self["webpackChunkteamb"] = self["webpackChunkteamb"] || []).push([["src_Dropdown_js"], {
    "./src/ArrowIcon.js":
        ((module, exports, require) => {
            "use strict";
            require.r(exports);
            require.d(exports, {
                "default": () => DEFAULT_EXPORT
            });
            const DEFAULT_EXPORT = ('ArrowIcon');
        }),
    "./src/Dropdown.js":
        ((module, exports, require) => {
            "use strict";
            require.r(exports);
            require.d(exports, {
                "default": () => DEFAULT_EXPORT
            });
            var is_array_0__ = require("webpack/sharing/consume/default/is-array/is-array");
            var is_array_0___default = require.n(is_array_0__);
            var _ArrowIcon_1__ = require("./src/ArrowIcon.js");
            const DEFAULT_EXPORT = (`(Dropdown[${_ArrowIcon_1__.default}][${(is_array_0___default().name)}])`);
        })
}]);
```

3.1.3 src_Button_js.js

teamb\dist\src_Button_js.js

```
(self["webpackChunkteamb"] = self["webpackChunkteamb"] || []).push([["src_Button_js"], {
    "./src/Button.js":
        ((module, exports, require) => {
            require.r(exports);
            require.d(exports, {
                "default": () => DEFAULT_EXPORT
            });
            var is_array_0__ = require("webpack/sharing/consume/default/is-array/is-array");
            var is_array_0___default = require.n(is_array_0__);
            const DEFAULT_EXPORT = (`(Button[${(is_array_0___default().name)}])`);
        })
}]);
```

3.2 teama

3.2.1 main.js

teama\dist\main.js

```
(() => {

    var modules = ({
        "webpack/container/reference/teamb": ((module, exports, require) => {

            module.exports = new Promise((resolve) => {
                require.l("http://localhost:8000/remoteEntry.js", resolve);
            }).then(() => window.teamb);
        })
    });
    var cache = {};
    function require(moduleId) {
        if (cache[moduleId]) {
            return cache[moduleId].exports;
        }
        var module = cache[moduleId] = {
            exports: {}
        };
        modules[moduleId](module, module.exports, require);
        return module.exports;
    }

    require.n = (module) => {
        var getter = module && module.__esModule ?
            () => module['default'] :
            () => module;
        return getter;
    };
    require.d = (exports, definition) => {
        for (var key in definition) {
            Object.defineProperty(exports, key, { get: definition[key] });
        }
    };
    require.u = (chunkId) => {
        return "" + chunkId + ".js";
    };
    require.o = (obj, prop) => Object.prototype.hasOwnProperty.call(obj, prop);
    require.l = (url, done) => {
        var script = document.createElement('script');
        script.src = url;
        script.onload = done
        document.head.appendChild(script);
    };
    require.r = (exports) => {
        Object.defineProperty(exports, Symbol.toStringTag, { value: 'Module' });
        Object.defineProperty(exports, '__esModule', { value: true });
    };
    require.f = {};
    require.e = (chunkId) => {
        return Promise.all(Object.keys(require.f).reduce((promises, key) => {
            require.f[key](chunkId, promises);
            return promises;
        }, []));
    };
    (() => {
```

```
    var chunkMapping = {
        "webpack_container_remote_teamb_Dropdown": [
            "webpack/container/remote/teamb/Dropdown"
        ],
        "webpack_container_remote_teamb_Button": [
            "webpack/container/remote/teamb/Button"
        ]
    };
    var idToExternalAndNameMapping = {
        "webpack/container/remote/teamb/Dropdown": [
            "default",
            "./Dropdown",
            "webpack/container/reference/teamb"
        ],
        "webpack/container/remote/teamb/Button": [
            "default",
            "./Button",
            "webpack/container/reference/teamb"
        ]
    };

    require.f.remotes = (chunkId, promises) => {

        if (require.o(chunkMapping, chunkId)) {

            chunkMapping[chunkId].forEach((id) => {

                var [scopeName, remoteExposeName, remoteId] = idToExternalAndNameMapping[id];

                let promise = require(remoteId).then(external => {

                    return require.I(scopeName).then(() => {

                        return external.get(remoteExposeName).then(factory => {

                            modules[id] = (module) => {

                                module.exports = factory();
                            }
                        });
                    });
                });
                promises.push(promise);
            });
        }
    }
})();

require.S = {};

require.I = (name) => {
    if (require.S[name])
        return Promise.resolve();
    var scope = require.S[name] = {};

    var register = (name, version, factory) => {
        var currentScope = scope[name] = scope[name] || {};
        currentScope[version] = { get: factory };
    };
    var promises = [];

    var initExternal = (id) => {
        var module = require(id);
        let promise = module.then(module => module.init(scope));
        promises.push(promise);
    }

    switch (name) {
        case "default": {
            register("is-array", "1.0.1", () => require.e("node_modules_is-array_index_js").then(() => () => require("./node_modules/is-array/index.js")));
            initExternal("webpack/container/reference/teamb");
        }
            break;
    }
    return Promise.all(promises)
};
require.p = "http://localhost:3000/";
var init = (fn) => function (scopeName, key, version) {
    return require.I(scopeName).then(() => {
        return fn(require.S[scopeName], key, version);
    });
};
var loadShareScope = init((scope, key, version) => {
    var versions = scope[key];
    var entry = versions[version];
    return entry.get()
});
(() => {

    var moduleToHandlerMapping = {
        "webpack/sharing/consume/default/is-array/is-array": () => loadShareScope("default", "is-array", '1.0.1')
    };
    var chunkMapping = {
        "src_bootstrap_js": [
            "webpack/sharing/consume/default/is-array/is-array"
        ]
    };
    require.f.consumes = (chunkId, promises) => {
        if (require.o(chunkMapping, chunkId)) {
            chunkMapping[chunkId].forEach((id) => {
                let promise = moduleToHandlerMapping[id]().then((factory) => {
                    modules[id] = (module) => {
                        module.exports = factory();
                    }
```

```
                    })
                    promises.push(promise);
                });
            }
        }
    })();
    var installedChunks = {
        "main": 0
    };
    require.f.j = (chunkId, promises) => {
        if (!/^webpack_container_remote_teamb_(Button|Dropdown)$/.test(chunkId)) {
            var promise = new Promise((resolve, reject) => {
                installedChunks[chunkId] = [resolve, reject];
            });
            promises.push(promise);
            var url = require.p + require.u(chunkId);
            require.l(url);
        }
    };
    var webpackJsonpCallback = (parentChunkLoadingFunction, data) => {
        var [chunkIds, moreModules] = data;
        var moduleId, chunkId, i = 0, resolves = [];
        for (; i < chunkIds.length; i++) {
            chunkId = chunkIds[i];
            if (require.o(installedChunks, chunkId) && installedChunks[chunkId]) {
                resolves.push(installedChunks[chunkId][0]);
            }
            installedChunks[chunkId] = 0;
        }
        for (moduleId in moreModules) {
            if (require.o(moreModules, moduleId)) {
                modules[moduleId] = moreModules[moduleId];
            }
        }
        if (parentChunkLoadingFunction) parentChunkLoadingFunction(data);
        while (resolves.length) {
            resolves.shift()();
        }
    }
    var chunkLoadingGlobal = self["webpackChunkteama"] = self["webpackChunkteama"] || [];
    chunkLoadingGlobal.forEach(webpackJsonpCallback.bind(null, 0));
    chunkLoadingGlobal.push = webpackJsonpCallback.bind(null, chunkLoadingGlobal.push.bind(chunkLoadingGlobal));
    require.e("src_bootstrap_js").then(require.bind(require, "./src/bootstrap.js"));
})();
```

3.2.2 src_bootstrap_js.js

teama\dist\src_bootstrap_js.js

```
(self["webpackChunkteama"] = self["webpackChunkteama"] || []).push([["src_bootstrap_js"], {
    "./src/HomePage.js":
        ((module, exports, require) => {
            "use strict";
            module.exports = (async () => {
                require.r(exports);
                require.d(exports, {
                    "default": () => DEFAULT_EXPORT
                });
                var is_array_0__ = require("webpack/sharing/consume/default/is-array/is-array");
                var is_array_0___default = require.n(is_array_0__);
                let Dropdown = await require.e("webpack_container_remote_teamb_Dropdown").then(require.bind(
```