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=132 sentences=459, words=2800

1.动机

- Module Federation的动机是为了不同开发小组间共同开发一个或者多个应用
- 应用将被划分为更小的应用块，一个应用块，可以是比如头部导航或者侧边栏的前端组件，也可以是数据获取逻辑的逻辑组件
- 每个应用块由不同的组开发
- 应用或应用块共享其他其他应用块或者库

2.Module Federation

- 使用Module Federation时，每个应用块都是一个独立的构建，这些构建都将编译为容器
- 容器可以被其他应用或者其他容器应用
- 一个被引用的容器被称为 `remote`, 引用者被称为 `host`， `remote`暴露模块给 `host`, `host`则可以使用这些暴露的模块，这些模块被成为 `remote`模块

3.实战

3.1 配置参数

字段 类型 含义 name string 必传值，即输出的模块名，被远程引用时路径为${name}/${expose} library object 声明全局变量的方式，name为umd的name filename string 构建输出的文件名 remotes object 远程引用的应用名及其别名的映射，使用时以key值作为name exposes object 被远程引用时可暴露的资源路径及其别名 shared object 与其他应用之间可以共享的第三方依赖，使你的代码中不用重复加载同一份依赖

3.2 安装

```
cnpm i webpack webpack-cli webpack-dev-server babel-loader @babel/core  @babel/preset-env -D
```

3.2 remote

3.2.1 remote\webpack.config.js

```
let path = require("path");
let webpack = require("webpack");
let HtmlWebpackPlugin = require("html-webpack-plugin");
const ModuleFederationPlugin = require("webpack/lib/container/ModuleFederationPlugin");
module.exports = {
    mode: "development",
    entry: "./src/index.js",
    output: {
        publicPath: "http://localhost:3000/",
    },
    devServer: {
        port: 3000
    },
    module: {
        rules: [
            {
                test: /\.jsx?$/,
                use: {
                    loader: 'babel-loader',
                    options: {
                        presets: ["@babel/preset-react"]
                    },
                },
                exclude: /node_modules/,
            },
        ]
    },
    plugins: [
        new HtmlWebpackPlugin({
            template:'./public/index.html'
        }),
        new ModuleFederationPlugin({
            filename: "remoteEntry.js",
            name: "remote",
            exposes: {
                "./NewsList": "./src/NewsList",
            }
          })
    ]
}
```

3.2.2 remote\src\index.js

remote\src\index.js

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

3.2.3 remote\src\bootstrap.js

remote\src\bootstrap.js

```
import React from "react";
import ReactDOM from "react-dom";
import App from "./App";
ReactDOM.render(<App />, document.getElementById("root"));
```

3.2.4 remote\src\App.js

remote\src\App.js

```
import React from "react";
import NewsList from './NewsList';
const App = () => (
  <div>
    <h2>本地组件NewsListh2>
    <NewsList />
  div>
);

export default App;
```

3.2.5 remote\src\NewsList.js

remote\src\NewsList.js

```
import React from "react";
export default ()=>(
    <div>新闻列表div>
)
```

3.3 host

3.3.1 host\webpack.config.js

host\webpack.config.js

```
let path = require("path");
let webpack = require("webpack");
let HtmlWebpackPlugin = require("html-webpack-plugin");
const ModuleFederationPlugin = require("webpack/lib/container/ModuleFederationPlugin");
module.exports = {
    mode: "development",
    entry: "./src/index.js",
    output: {
        publicPath: "http://localhost:8000/",
    },
    devServer: {
        port: 8000
    },
    module: {
        rules: [
            {
                test: /\.jsx?$/,
                use: {
                    loader: 'babel-loader',
                    options: {
                        presets: ["@babel/preset-react"]
                    },
                },
                exclude: /node_modules/,
            },
        ]
    },
    plugins: [
        new HtmlWebpackPlugin({
            template: './public/index.html'
        }),
        new ModuleFederationPlugin({
            filename: "remoteEntry.js",
            name: "host",
            remotes: {
                remote: "remote@http://localhost:3000/remoteEntry.js"
            }
        })
    ]
}
```

3.3.2 host\src\index.js

host\src\index.js

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

3.3.3 host\src\bootstrap.js

host\src\bootstrap.js

```
import React from "react";
import ReactDOM from "react-dom";
import App from "./App";
ReactDOM.render(<App />, document.getElementById("root"));
```

3.3.4 host\src\App.js

host\src\App.js

```
import React from "react";
import Slides from './Slides';
const RemoteNewsList = React.lazy(() => import("remote/NewsList"));

const App = () => (
  <div>
    <h2 >本地组件Slides, 远程组件NewsListh2>
    <Slides />
    <React.Suspense fallback="Loading NewsList">
      <RemoteNewsList />
    React.Suspense>
  div>
);
export default App;
```

3.3.5 host\src\Slides.js

host\src\Slides.js

```
import React from "react";
export default ()=>(
    <div>轮播图div>
)
```

- shared配置主要是用来避免项目出现多个公共依赖

4.1 remote\webpack.config.js

```
    plugins: [
        new HtmlWebpackPlugin({
            template:'./public/index.html'
        }),
        new ModuleFederationPlugin({
            filename: "remoteEntry.js",
            name: "remote",
            exposes: {
                "./NewsList": "./src/NewsList",
            },
+            shared:{
+                react: { singleton: true },
+                "react-dom": { singleton: true }
+              }
          })
    ]
```

4.2 host\webpack.config.js

```
    plugins: [
        new HtmlWebpackPlugin({
            template: './public/index.html'
        }),
        new ModuleFederationPlugin({
            filename: "remoteEntry.js",
            name: "host",
            remotes: {
                remote: "remote@http://localhost:3000/remoteEntry.js"
            },
+           shared:{
+                react: { singleton: true },
+                "react-dom": { singleton: true }
+           }
        })
    ]
```

5.双向依赖

- Module Federation 的共享可以是双向的

5.1 remote\webpack.config.js

```
    plugins: [
        new HtmlWebpackPlugin({
            template:'./public/index.html'
        }),
        new ModuleFederationPlugin({
            filename: "remoteEntry.js",
            name: "remote",
+            remotes: {
+                host: "host@http://localhost:8000/remoteEntry.js"
+            },
            exposes: {
                "./NewsList": "./src/NewsList",
            },
            shared:{
                react: { singleton: true },
                "react-dom": { singleton: true }
              }
          })
    ]
```

5.2 host\webpack.config.js

```
    plugins: [
        new HtmlWebpackPlugin({
            template: './public/index.html'
        }),
        new ModuleFederationPlugin({
            filename: "remoteEntry.js",
            name: "host",
            remotes: {
                remote: "remote@http://localhost:3000/remoteEntry.js"
            },
+           exposes: {
+                "./Slides": "./src/Slides",
+           },
            shared:{
                react: { singleton: true },
                "react-dom": { singleton: true }
              }
        })
    ]
```

5.3 remote\src\App.js

remote\src\App.js

```
import React from "react";
import NewsList from './NewsList';
+const RemoteSlides = React.lazy(() => import("host/Slides"));
const App = () => (

+    本地组件NewsList,远程组件Slides

+
+
+

);

export default App;
```

6.多个remote

6.1 all\webpack.config.js

```
let path = require("path");
let webpack = require("webpack");
let HtmlWebpackPlugin = require("html-webpack-plugin");
const ModuleFederationPlugin = require("webpack/lib/container/ModuleFederationPlugin");
module.exports = {
    mode: "development",
    entry: "./src/index.js",
    output: {
        publicPath: "http://localhost:3000/",
    },
    devServer: {
        port: 5000
    },
    module: {
        rules: [
            {
                test: /\.jsx?$/,
                use: {
                    loader: 'babel-loader',
                    options: {
                        presets: ["@babel/preset-react"]
                    },
                },
                exclude: /node_modules/,
            },
        ]
    },
    plugins: [
        new HtmlWebpackPlugin({
            template:'./public/index.html'
        }),
        new ModuleFederationPlugin({
            filename: "remoteEntry.js",
            name: "all",
            remotes: {
                remote: "remote@http://localhost:3000/remoteEntry.js",
                host: "host@http://localhost:8000/remoteEntry.js",
            },
            shared:{
                react: { singleton: true },
                "react-dom": { singleton: true }
              }
          })
    ]
}
```

6.2 remote\src\index.js

remote\src\index.js

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

6.3 remote\src\bootstrap.js

remote\src\bootstrap.js

```
import React from "react";
import ReactDOM from "react-dom";
import App from "./App";
ReactDOM.render(<App />, document.getElementById("root"));
```

6.4 remote\src\App.js

remote\src\App.js

```
import React from "react";
const RemoteSlides = React.lazy(() => import("host/Slides"));
const RemoteNewsList = React.lazy(() => import("remote/NewsList"));
const App = () => (
  <div>
    <h2>远程组件Slides,远程组件NewsListh2>
    <React.Suspense fallback="Loading Slides">
      <RemoteSlides />
    React.Suspense>
    <React.Suspense fallback="Loading NewsList">
      <RemoteNewsList />
    React.Suspense>
  div>
);

export default App;
```

7. 运行原理

7.1 源代码

7.1.1 remote

7.1.1.1 webpack.config.js

Remote\webpack.config.js

```
let path = require("path");
let webpack = require("webpack");
let HtmlWebpackPlugin = require("html-webpack-plugin");
const ModuleFederationPlugin = require("webpack/lib/container/ModuleFederationPlugin");
module.exports = {
    mode: "development",
    devtool:false,
    entry: "./src/index.js",
    output: {
        publicPath: "http://localhost:8080/",

    },
    devServer: {
        port: 3000
    },
    module: {
        rules: [
            {
                test: /\.jsx?$/,
                use: {
                    loader: 'babel-loader',
                    options: {
                        presets: ["@babel/preset-react"]
                    },
                },
                exclude: /node_modules/,
            },
        ]
    },
    plugins: [
        new HtmlWebpackPlugin({
            template:'./public/index.html'
        }),
        new ModuleFederationPlugin({
            filename: "remoteEntry.js",
            name: "remote",
            exposes: {
                "./title": "./src/title",
            }
          })
    ]
}
```

7.1.1.2 Remote\src\index.js

Remote\src\index.js

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

7.1.1.3 Remote\src\bootstrap.js

Remote\src\bootstrap.js

```
import title from "./title";
console.log(title);
```

7.1.1.4 Remote\src\title.js

Remote\src\title.js

```
module.exports = 'title';
```

7.1.2 Host

7.1.2.1 Host\webpack.config.js

Host\webpack.config.js

```
let path = require("path");
let webpack = require("webpack");
let HtmlWebpackPlugin = require("html-webpack-plugin");
const ModuleFederationPlugin = require("webpack/lib/container/ModuleFederationPlugin");
module.exports = {
    mode: "development",
    entry: "./src/index.js",
    devtool:false,
    output: {
        publicPath: "http://localhost:8081/",
        hotUpdateGlobal:'webpackHotUpdate'
    },
    devServer: {
        port: 8000
    },
    module: {
        rules: [
            {
                test: /\.jsx?$/,
                use: {
                    loader: 'babel-loader',
                    options: {
                        presets: ["@babel/preset-react"]
                    },
                },
                exclude: /node_modules/,
            },
        ]
    },
    plugins: [
        new HtmlWebpackPlugin({
            template: './public/index.html'
        }),
        new ModuleFederationPlugin({
            filename: "remoteEntry.js",
            name: "host",
            remotes: {
                remote: "remoteVar@http://localhost:3000/remoteEntry.js"
            }
        })
    ]
}
```

7.1.2.2 Host\src\index.js

Host\src\index.js

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

7.1.2.3 Host\src\bootstrap.js

Host\src\bootstrap.js

```
import("remote/title").then(result=>{
    console.log(result.default);
});
```

7.2 输出

7.2.1 Remote输出

7.2.1.1 Remote\dist\src_title_js.js

Remote\dist\src_title_js.js

```
(self["webpackChunkRemote"] = self["webpackChunkRemote"] || []).push([["src_title_js"], {
  "./src/title.js":
    ((module) => {
      module.exports = 'title';
    })
}]);
```

7.2.1.2 Remote\dist\src_bootstrap_js.js

Remote\dist\src_bootstrap_js.js

```
(self["webpackChunkRemote"] = self["webpackChunkRemote"] || []).push([["src_bootstrap_js"], {
  "./src/bootstrap.js":
    ((module, exports, require) => {
      require.r(exports);
      var title = require("./src/title.js");
      var title_default = require.n(title);
      console.log((title_default()));
    }),
  "./src/title.js":
    ((module) => {
      module.exports = 'title';
    })
}]);
```

7.2.1.3 Remote\dist\remoteEntry.js

Remote\dist\remoteEntry.js

```
window.remote =
    (() => {
        var modules = ({
            "webpack/container/entry/remote":
                ((module, exports, require) => {
                    var moduleMap = {
                        "./title": () => {
                            return require.e("src_title_js").then(() => () => require("./src/title.js"));
                        }
                    };
                    var get = (module) => {
                        return moduleMap[module]();
                    };
                    var init = () => {
                        return Promise.resolve();
                    };
                    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.m = modules;
        require.d = (exports, definition) => {
            for (var key in definition) {
                Object.defineProperty(exports, key, { enumerable: true, 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) => {
            var script = document.createElement('script');
            script.src = url;
            document.head.appendChild(script);
        };
        require.p = "http://localhost:8080/";
        var installedChunks = {
            "remote": 0
        };
        require.f.j = (chunkId, promises) => {
            var installedChunkData = installedChunks[chunkId];
            if (installedChunkData !== 0) {
                if (installedChunkData) {
                    promises.push(installedChunkData[2]);
                } else {
                    var promise = new Promise((resolve, reject) => {
                        installedChunkData = installedChunks[chunkId] = [resolve, reject];
                    });
                    promises.push(installedChunkData[2] = 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 (installedChunks[chunkId]) {
                    resolves.push(installedChunks[chunkId][0]);
                }
                installedChunks[chunkId] = 0;
            }
            for (moduleId in moreModules) {
                require.m[moduleId] = moreModules[moduleId];
            }
            if (parentChunkLoadingFunction) parentChunkLoadingFunction(data);
            while (resolves.length) {
                resolves.shift()();
            }
        }
        var chunkLoadingGlobal = self["webpackChunkRemote"] = self["webpackChunkRemote"] || [];
        chunkLoadingGlobal.forEach(webpackJsonpCallback.bind(null, 0));
        chunkLoadingGlobal.push = webpackJsonpCallback.bind(null, chunkLoadingGlobal.push.bind(chunkLoadingGlobal));
        return require("webpack/container/entry/remote");
    })();
```

7.2.1.4 Remote\dist\main.js

Remote\dist\main.js

```
(() => {
  var modules = ({});
  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.m = modules;
  require.n = (module) => {
    var getter = module && module.__esModule ?
      () => module['default'] :
      () => module;
    require.d(getter, { a: getter });
    return getter;
  };
  require.d = (exports, definition) => {
    for (var key in definition) {
      Object.defineProperty(exports, key, { enumerable: true, 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.l = (url) => {
    var script = document.createElement('script');
    script.src = url;
    document.head.appendChild(script);
  };
  require.r = (exports) => {
    Object.defineProperty(exports, Symbol.toStringTag, { value: 'Module' });
    Object.defineProperty(exports, '__esModule', { value: true });
  };
  require.p = "http://localhost:8080/";
  var installedChunks = {
    "main": 0
  };
  require.f.j = (chunkId, promises) => {
    var installedChunkData = installedChunks[chunkId];
    if (installedChunkData !== 0) {
      if (installedChunkData) {
        promises.push(installedChunkData[2]);
      } else {
        var promise = new Promise((resolve, reject) => {
          installedChunkData = installedChunks[chunkId] = [resolve, reject];
        });
        promises.push(installedChunkData[2] = 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 (installedChunks[chunkId]) {
        resolves.push(installedChunks[chunkId][0]);
      }
      installedChunks[chunkId] = 0;
    }
    for (moduleId in moreModules) {
      require.m[moduleId] = moreModules[moduleId];
    }
    if (parentChunkLoadingFunction) parentChunkLoadingFunction(data);
    while (resolves.length) {
      resolves.shift()();
    }
  }
  var chunkLoadingGlobal = self["webpackChunkRemote"] = self["webpackChunkRemote"] || [];
  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"));
})();
```

7.2.1.5 Remote\dist\index.html

Remote\dist\index.html

```

<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Documenttitle>
head>
<body>
    <div id="root">div>
<script src="http://localhost:8080/main.js">script>
[illegible]
[illegible]
```

7.2.2 Host输出

7.2.2.1 Host\dist\src_bootstrap_js.js

Host\dist\src_bootstrap_js.js

```
(self["webpackChunkHost"] = self["webpackChunkHost"] || []).push([["src_bootstrap_js"], {
  "./src/bootstrap.js":
    ((module, exports, require) => {
      require.e("webpack_container_remote_remote_title").then(require.t.bind(require, "webpack/container/remote/remote/title")).then(result => {
        console.log(result.default);
      });
    })
}]);
```

7.2.2.2 Host\dist\main.js

Host\dist\main.js

```
(() => {
    var modules = ({
        "webpack/container/reference/remote":
            ((module, exports, require) => {
                module.exports = new Promise((resolve) => {
                    require.l("http://localhost:8080/remoteEntry.js",resolve);
                }).then(() => remote);
            })
    });
    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.m = modules;
    require.t = function (value) {
        value = this(value);
        var ns = Object.create(null);
        require.r(ns);
        var def = { 'default': () => value };
        require.d(ns, def);
        return ns;
    };
    require.d = (exports, definition) => {
        for (var key in definition) {
            Object.defineProperty(exports, key, { enumerable: true, 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.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 });
    };
    var chunkMapping = {
        "webpack_container_remote_remote_title": [
            "webpack/container/remote/remote/title"
        ]
    };
    var idToExternalAndNameMapping = {
        "webpack/container/remote/remote/title": [
            "./title",
            "webpack/container/reference/remote"
        ]
    };
    require.f.remotes = (chunkId,promises) => {
        chunkMapping[chunkId] && chunkMapping[chunkId].forEach((id) => {
            var [remoteModuleName,moduleId] = idToExternalAndNameMapping[id];
            let promise = require(moduleId).then(external=>{
                return external.init().then(()=>{
                    return external.get(remoteModuleName).then((factory)=>{
                        return modules[id] = (module) => {
                            module.exports = factory();
                        }
                    });
                });
            })
            promises.push(promise);
        });
    }
    require.p = "http://localhost:8081/";
    var installedChunks = {
        "main": 0
    };
    require.f.j = (chunkId, promises) => {
        var installedChunkData;
        if ("src_bootstrap_js" == chunkId) {
            var promise = new Promise((resolve, reject) => {
                installedChunkData = installedChunks[chunkId] = [resolve, reject];
            });
            promises.push(installedChunkData[2] = promise);
            var url = require.p + require.u(chunkId);
            require.l(url);
        } else {
            installedChunks[chunkId] = 0;
        }
    };
    var webpackJsonpCallback = (data) => {
        var [chunkIds, moreModules] = data;
        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;
        }
        for (moduleId in moreModules) {
            require.m[moduleId] = moreModules[moduleId];
        }
        while (resolves.length) {
            resolves.shift()();
        }
    }
    var chunkLoadingGlobal = self["webpackChunkHost"] = self["webpackChunkHost"] || [];
    chunkLoadingGlobal.push = webpackJsonpCallback;
    require.e("src_bootstrap_js").then(require.t.bind(require, "./src/bootstrap.js"));
})();
```

7.2.2.3 Host\dist\index.html

Host\dist\index.html

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Documenttitle>
head>
<body>
    <div id="root">div>
    <script src="http://localhost:8081/main.js">script>
body>
html>
```