link: null
title: 珠峰架构师成长计划
description: .vscode\launch.json
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=97 sentences=370, words=2707

1.调试 webpack

1.1 通过 chrome 调试

```
node --inspect-brk ./node_modules/webpack-cli/bin/cli.js
```

> 然后打开 Chrome 浏览器控制台就可以调试了

1.2 通过执行命令调试

- 打开工程目录，点击调试按钮，再点击小齿轮的配置按钮系统就会生成 `launch.json` 配置文件
- 修改好了以后直接点击 F5 就可以启动调试

.vscode\launch.json

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "type": "node",
      "request": "launch",
      "name": "debug webpack",
      "cwd": "${workspaceFolder}",
      "program": "${workspaceFolder}/node_modules/webpack-cli/bin/cli.js"
    }
  ]
}
```

1.3 debugger.js

```
const webpack = require("webpack");
const webpackOptions = require("./webpack.config");
const compiler = webpack(webpackOptions);

compiler.run((err, stats) => {
  console.log(err);
  console.log(
    stats.toJson({
      assets: true,
      chunks: true,
      modules: true,
    })
  );
});
```

2. tapable.js

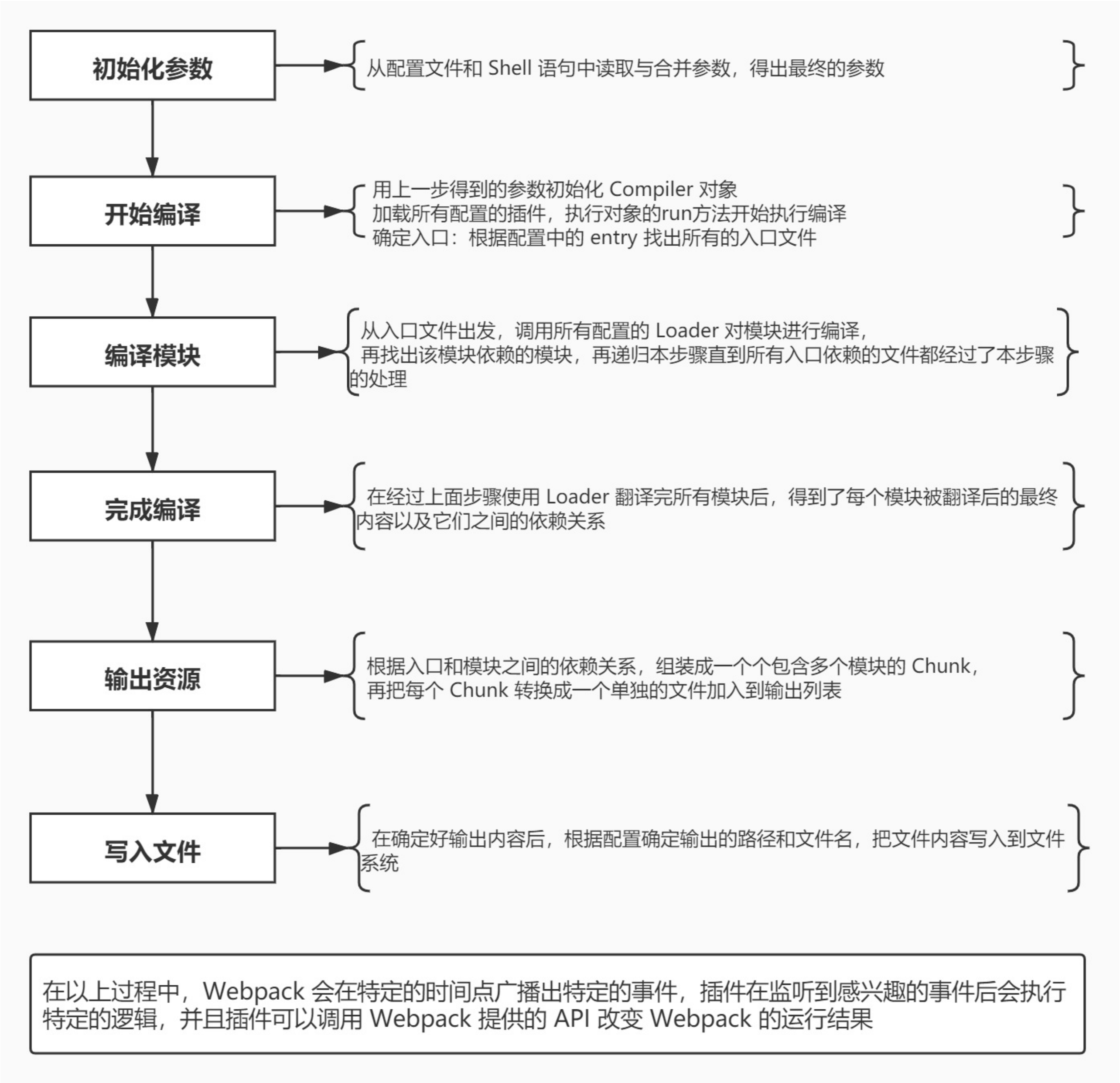
- tapable 是一个类似于 Node.js 中的 EventEmitter 的库，但更专注于自定义事件的触发和处理
- webpack 通过 tapable 将实现与流程解耦，所有具体实现通过插件的形式存在

```
class SyncHook {
  constructor() {
    this.taps = [];
  }
  tap(name, fn) {
    this.taps.push(fn);
  }
  call() {
    this.taps.forEach((tap) => tap());
  }
}

let hook = new SyncHook();
hook.tap("some name", () => {
  console.log("some name");
});

class Plugin {
  apply() {
    hook.tap("Plugin", () => {
      console.log("Plugin ");
    });
  }
}
new Plugin().apply();
hook.call();
```

3. webpack 编译流程

1. 初始化参数：从配置文件和 Shell 语句中读取并合并参数,得出最终的配置对象
2. 用上一步得到的参数初始化 Compiler 对象
3. 加载所有配置的插件
4. 执行对象的 run 方法开始执行编译
5. 根据配置中的 `entry`找出入口文件
6. 从入口文件出发,调用所有配置的 `Loader`对模块进行编译
7. 再找出该模块依赖的模块，再递归本步骤直到所有入口依赖的文件都经过了本步骤的处理
8. 根据入口和模块之间的依赖关系，组装成一个个包含多个模块的 Chunk
9. 再把每个 Chunk 转换成一个单独的文件加入到输出列表
10. 在确定好输出内容后，根据配置确定输出的路径和文件名，把文件内容写入到文件系统

在以上过程中，Webpack 会在特定的时间点广播出特定的事件，插件在监听到感兴趣的事件后会执行特定的逻辑，并且插件可以调用 Webpack 提供的 API 改变 Webpack 的运行结果



初始化参数

从配置文件和 Shell 语句中读取与合并参数，得出最终的参数

开始编译

用上一步得到的参数初始化 Compiler 对象
加载所有配置的插件，执行对象的run方法开始执行编译
确定入口：根据配置中的 entry 找出所有的入口文件

编译模块

从入口文件出发，调用所有配置的 Loader 对模块进行编译，
再找出该模块依赖的模块，再递归本步骤直到所有入口依赖的文件都经过了本步骤的处理

完成编译

在经过上面步骤使用 Loader 翻译完所有模块后，得到了每个模块被翻译后的最终内容以及它们之间的依赖关系

输出资源

根据入口和模块之间的依赖关系，组装成一个个包含多个模块的 Chunk，
再把每个 Chunk 转换成一个单独的文件加入到输出列表

写入文件

在确定好输出内容后，根据配置确定输出的路径和文件名，把文件内容写入到文件系统

在以上过程中，Webpack 会在特定的时间点广播出特定的事件，插件在监听到感兴趣的事件后会执行特定的逻辑，并且插件可以调用 Webpack 提供的 API 改变 Webpack 的运行结果

3.1 debugger.js

debugger.js

```
const webpack = require("./webpack");
const options = require("./webpack.config");
const compiler = webpack(options);
compiler.run((err, stats) => {
  console.log(err);
  console.log(
    JSON.stringify(
      stats.toJson({
        assets: true,
        chunks: true,
        modules: true,
      }),
      null,
      2
    )
  );
});
```

3.2 webpack.config.js

webpack.config.js

```
const path = require("path");
const RunPlugin = require("./plugins/run-plugin");
const DonePlugin = require("./plugins/done-plugin");
module.exports = {
  mode: "development",
  devtool: false,
  entry: {
    entry1: "./src/entry1.js",
    entry2: "./src/entry2.js",
  },
  output: {
    path: path.resolve("dist"),
    filename: "[name].js",
  },
  resolve: {
    extensions: [".js", ".jsx", ".ts", ".tsx", ".json"],
  },
  module: {
    rules: [
      {
        test: /\.js$/,
        use: [
          path.resolve(__dirname, "loaders/logger1-loader.js"),
          path.resolve(__dirname, "loaders/logger2-loader.js"),
        ],
      },
    ],
  },
  plugins: [
    new RunPlugin(),
    new DonePlugin(),
  ],
};
```

3.3 webpack.js

webpack.js

```
let Compiler = require("./Compiler");
function webpack(options) {

  console.log(process.argv);
  let argv = process.argv.slice(2);
  let shellOptions = argv.reduce((shellOptions, option) => {
    let [key, value] = option.split("=");
    shellOptions[key.slice(2)] = value;
    return shellOptions;
  }, {});
  let finalOptions = { ...options, ...shellOptions };
  console.log("finalOptions", finalOptions);

  let compiler = new Compiler(finalOptions);

  let { plugins } = finalOptions;
  for (let plugin of plugins) {
    plugin.apply(compiler);
  }
  return compiler;
}
module.exports = webpack;
```

3.4 Compiler.js

Compiler.js

```
let { SyncHook } = require("tapable");
let fs = require("fs");
let path = require("path");
let Complication = require("./Complication");

class Compiler {
  constructor(options) {
    this.options = options;
    this.hooks = {
      run: new SyncHook(),
      done: new SyncHook(),
    };
  }

  run(callback) {
    this.hooks.run.call();

    const onCompiled = (err, stats, fileDependencies) => {

      for (let filename in stats.assets) {
        let filePath = path.join(this.options.output.path, filename);
        fs.writeFileSync(filePath, stats.assets[filename], "utf8");
      }
      callback(err, {
        toJson: () => stats,
      });
      fileDependencies.forEach((fileDependency) =>
        fs.watch(fileDependency, () => this.compile(onCompiled))
      );
    };
    this.compile(onCompiled);
    this.hooks.done.call();
  }
  compile(callback) {

    let complication = new Complication(this.options);
    complication.build(callback);
  }
}
module.exports = Compiler;
```

3.5 Complication.js

Complication.js

```
let fs = require("fs");
let types = require("babel-types");
let parser = require("@babel/parser");
let traverse = require("@babel/traverse").default;
let generator = require("@babel/generator").default;
const path = require("path");

let baseDir = toUnixPath(process.cwd());
function toUnixPath(filePath) {
  return filePath.replace(/\\/g, "/");
}
class Complication {
  constructor(options) {
    this.options = options;
    this.modules = [];
    this.chunks = [];
    this.assets = {};
    this.fileDependencies = [];
  }

  build(callback) {

    let entry = {};
    if (typeof this.options.entry === "string") {
      entry.main = this.options.entry;
    } else {
      entry = this.options.entry;
    }
    for (let entryName in entry) {

      let entryFilePath = path.posix.join(baseDir, entry[entryName]);
      this.fileDependencies.push(entryFilePath);

      let entryModule = this.buildModule(entryName, entryFilePath);

      let chunk = {
        name: entryName,
        entryModule,
        modules: this.modules.filter((item) => item.name.includes(entryName)),
      };
      this.chunks.push(chunk);
    }

    this.chunks.forEach((chunk) => {
      let filename = this.options.output.filename.replace("[name]", chunk.name);
      this.assets[filename] = getSource(chunk);
    });

    callback(
      null,
      {
        chunks: this.chunks,
        modules: this.modules,
        assets: this.assets,
      },
      this.fileDependencies
    );
  }
  buildModule(name, modulePath) {

    let sourceCode = fs.readFileSync(modulePath, "utf8");
    let { rules } = this.options.module;
    let loaders = [];
    rules.forEach((rule) => {
      let { test } = rule;
      if (modulePath.match(test)) {
        loaders.push(...rule.use);
      }
    });
    sourceCode = loaders.reduceRight((sourceCode, loader) => {
      return require(loader)(sourceCode);
    }, sourceCode);

    let moduleId = "./" + path.posix.relative(baseDir, modulePath);
    let module = { id: moduleId, dependencies: [], name: [name] };

    let ast = parser.parse(sourceCode, { sourceType: "module" });
    traverse(ast, {
      CallExpression: ({ node }) => {
        if (node.callee.name === "require") {

          let depModuleName = node.arguments[0].value;

          let dirname = path.posix.dirname(modulePath);

          let depModulePath = path.posix.join(dirname, depModuleName);
          let extensions = this.options.resolve.extensions;
          depModulePath = tryExtensions(depModulePath, extensions);
          this.fileDependencies.push(depModulePath);

          let depModuleId = "./" + path.posix.relative(baseDir, depModulePath);
          node.arguments = [types.stringLiteral(depModuleId)];

          module.dependencies.push({ depModuleId, depModulePath });
        }
      },
    });
    let { code } = generator(ast);
    module._source = code;

    module.dependencies.forEach(({ depModuleId, depModulePath }) => {
      let existModule = this.modules.find((item) => item.id === depModuleId);
      if (existModule) {
        existModule.name.push(name);
      } else {
        let depModule = this.buildModule(name, depModulePath);
        this.modules.push(depModule);
      }
    });
    return module;
  }
}
function tryExtensions(modulePath, extensions) {
  if (fs.existsSync(modulePath)) {
    return modulePath;
  }
  for (let i = 0; i < extensions.length; i++) {
    let filePath = modulePath + extensions[i];
    if (fs.existsSync(filePath)) {
      return filePath;
    }
  }
  throw new Error(`${modulePath}没找到`);
}

function getSource(chunk) {
  return `
   (() => {
    var modules = {
      ${chunk.modules.map(
        (module) => `
        "${module.id}": (module) => {
          ${module._source}
        },
      `
      )}
    };
    var cache = {};
    function require(moduleId) {
      var cachedModule = cache[moduleId];
      if (cachedModule !== undefined) {
        return cachedModule.exports;
      }
      var module = (cache[moduleId] = {
        exports: {},
      });
      modules[moduleId](module, module.exports, require);
      return module.exports;
    }
    var exports ={};
    ${chunk.entryModule._source}
  })();
   `;
}
module.exports = Complication;
```

3.6 run-plugin.js

plugins\run-plugin.js

```
class RunPlugin {

  apply(compiler) {
    compiler.hooks.run.tap("RunPlugin", () => {
      console.log("run:开始编译");
    });
  }
}
module.exports = RunPlugin;
```

3.7 done-plugin.js

plugins\done-plugin.js

```
class DonePlugin {

  apply(compiler) {
    compiler.hooks.done.tap("DonePlugin", () => {
      console.log("done:结束编译");
    });
  }
}
module.exports = DonePlugin;
```

3.9 logger1-loader.js

loaders\logger1-loader.js

```
function loader(source) {
  return source + "//logger1";
}
module.exports = loader;
```

3.10 logger2-loader.js

loaders\logger2-loader.js

```
function loader(source) {

  return source + "//logger2";
}
module.exports = loader;
```

3.11 src\entry1.js

src\entry1.js

```
let title = require("./title");
console.log("entry12", title);
```

3.12 src\entry2.js

src\entry2.js

```
let title = require("./title.js");
console.log("entry2", title);
```

3.13 src\title.js

src\title.js

```
module.exports = "title";
```

4.Stats 对象

- 在 Webpack 的回调函数中会得到 stats 对象
- 这个对象实际来自于 Compilation.getStats()，返回的是主要含有 modules、 chunks和 assets三个属性值的对象。
- Stats 对象本质上来自于lib/Stats.js (https://github.com/webpack/webpack/blob/v4.39.3/lib/Stats.js)的类实例

字段 含义 modules 记录了所有解析后的模块 chunks 记录了所有 chunk assets 记录了所有要生成的文件

```
npx webpack --profile --json > stats.json
```

```
{
  "hash": "780231fa9b9ce4460c8a",
  "version": "5.8.0",
  "time": 83,
  "builtAt": 1606538839612,
  "publicPath": "auto",
  "outputPath": "C:\\webpack5\\dist",
  "assetsByChunkName": {

    "main": ["main.js"]
  },
  "assets": [

    {
      "type": "asset",
      "name": "main.js",
      "size": 2418,
      "chunkNames": [

        "main"
      ],
      "chunkIdHints": [],
      "auxiliaryChunkNames": [],
      "auxiliaryChunkIdHints": [],
      "emitted": false,
      "comparedForEmit": true,
      "cached": false,
      "info": {
        "javascriptModule": false,
        "size": 2418
      },
      "related": {},
      "chunks": ["main"],
      "auxiliaryChunks": [],
      "isOverSizeLimit": false
    }
  ],
  "chunks": [

    {
      "rendered": true,
      "initial": true,
      "entry": true,
      "recorded": false,
      "size": 80,
      "sizes": {
        "javascript": 80
      },
      "names": ["main"],
      "idHints": [],
      "runtime": ["main"],
      "files": ["main.js"],
      "auxiliaryFiles": [],
      "hash": "d25ad7a8144077f69783",
      "childrenByOrder": {},
      "id": "main",
      "siblings": [],
      "parents": [],
      "children": [],
      "modules": [
        {
          "type": "module",
          "moduleType": "javascript/auto",
          "identifier": "C:\\webpack5\\src\\index.js",
          "name": "./src/index.js",
          "nameForCondition": "C:\\webpack5\\src\\index.js",
          "index": 0,
          "preOrderIndex": 0,
          "index2": 1,
          "postOrderIndex": 1,
          "size": 55,
          "sizes": {
            "javascript": 55
          },
          "cacheable": true,
          "built": true,
          "codeGenerated": true,
          "cached": false,
          "optional": false,
          "orphan": false,
          "dependent": false,
          "issuer": null,
          "issuerName": null,
          "issuerPath": null,
          "failed": false,
          "errors": 0,
          "warnings": 0,
          "profile": {
            "total": 38,
            "resolving": 26,
            "restoring": 0,
            "building": 12,
            "integration": 0,
            "storing": 0,
            "additionalResolving": 0,
            "additionalIntegration": 0,
            "factory": 26,
            "dependencies": 0
          },
          "id": "./src/index.js",
          "issuerId": null,
          "chunks": ["main"],
          "assets": [],
          "reasons": [
            {
              "moduleIdentifier": null,
              "module": null,
              "moduleName": null,
              "resolvedModuleIdentifier": null,
              "resolvedModule": null,
              "type": "entry",
              "active": true,
              "explanation": "",
              "userRequest": "./src/index.js",
              "loc": "main",
              "moduleId": null,
              "resolvedModuleId": null
            }
          ],
          "usedExports": null,
          "providedExports": null,
          "optimizationBailout": [],
          "depth": 0
        },
        {
          "type": "module",
          "moduleType": "javascript/auto",
          "identifier": "C:\\webpack5\\src\\title.js",
          "name": "./src/title.js",
          "nameForCondition": "C:\\webpack5\\src\\title.js",
          "index": 1,
          "preOrderIndex": 1,
          "index2": 0,
          "postOrderIndex": 0,
          "size": 25,
          "sizes": {
            "javascript": 25
          },
          "cacheable": true,
          "built": true,
          "codeGenerated": true,
          "cached": false,
          "optional": false,
          "orphan": false,
          "dependent": true,
          "issuer": "C:\\webpack5\\src\\index.js",
          "issuerName": "./src/index.js",
          "issuerPath": [
            {
              "identifier": "C:\\webpack5\\src\\index.js",
              "name": "./src/index.js",
              "profile": {
                "total": 38,
                "resolving": 26,
                "restoring": 0,
                "building": 12,
                "integration": 0,
                "storing": 0,
                "additionalResolving": 0,
                "additionalIntegration": 0,
                "factory": 26,
                "dependencies": 0
              },
              "id": "./src/index.js"
            }
          ],
          "failed": false,
          "errors": 0,
          "warnings": 0,
          "profile": {
            "total": 0,
            "resolving": 0,
            "restoring": 0,
            "building": 0,
            "integration": 0,
            "storing": 0,
            "additionalResolving": 0,
            "additionalIntegration": 0,
            "factory": 0,
            "dependencies": 0
          },
          "id": "./src/title.js",
          "issuerId": "./src/index.js",
          "chunks": ["main"],
          "assets": [],
          "reasons": [
            {
              "moduleIdentifier": "C:\\webpack5\\src\\index.js",
              "module": "./src/index.js",
              "moduleName": "./src/index.js",
              "resolvedModuleIdentifier": "C:\\webpack5\\src\\index.js",
              "resolvedModule": "./src/index.js",
              "type": "cjs require",
              "active": true,
              "explanation": "",
              "userRequest": "./title.js",
              "loc": "1:12-33",
              "moduleId": "./src/index.js",
              "resolvedModuleId": "./src/index.js"
            },
            {
              "moduleIdentifier": "C:\\webpack5\\src\\title.js",
              "module": "./src/title.js",
              "moduleName": "./src/title.js",
              "resolvedModuleIdentifier": "C:\\webpack5\\src\\title.js",
              "resolvedModule": "./src/title.js",
              "type": "cjs self exports reference",
              "active": true,
              "explanation": "",
              "userRequest": null,
              "loc": "1:0-14",
              "moduleId": "./src/title.js",
              "resolvedModuleId": "./src/title.js"
            }
          ],
          "usedExports": null,
          "providedExports": null,
          "optimizationBailout": [
            "CommonJS bailout: module.exports is used directly at 1:0-14"
          ],
          "depth": 1
        }
      ],
      "origins": [
        {
          "module": "",
          "moduleIdentifier": "",
          "moduleName": "",
          "loc": "main",
          "request": "./src/index.js"
        }
      ]
    }
  ],
  "modules": [

    {
      "type": "module",
      "moduleType": "javascript/auto",
      "identifier": "C:\\webpack5\\src\\index.js",
      "name": "./src/index.js",
      "nameForCondition": "C:\\webpack5\\src\\index.js",
      "index": 0,
      "preOrderIndex": 0,
      "index2": 1,
      "postOrderIndex": 1,
      "size": 55,
      "sizes": {
        "javascript": 55
      },
      "cacheable": true,
      "built": true,
      "codeGenerated": true,
      "cached": false,
      "optional": false,
      "orphan": false,
      "issuer": null,
      "issuerName": null,
      "issuerPath": null,
      "failed": false,
      "errors": 0,
      "warnings": 0,
      "profile": {
        "total": 38,
        "resolving": 26,
        "restoring": 0,
        "building": 12,
        "integration": 0,
        "storing": 0,
        "additionalResolving": 0,
        "additionalIntegration": 0,
        "factory": 26,
        "dependencies": 0
      },
      "id": "./src/index.js",
      "issuerId": null,
      "chunks": ["main"],
      "assets": [],
      "reasons": [
        {
          "moduleIdentifier": null,
          "module": null,
          "moduleName": null,
          "resolvedModuleIdentifier": null,
          "resolvedModule": null,
          "type": "entry",
          "active": true,
          "explanation": "",
          "userRequest": "./src/index.js",
          "loc": "main",
          "moduleId": null,
          "resolvedModuleId": null
        }
      ],
      "usedExports": null,
      "providedExports": null,
      "optimizationBailout": [],
      "depth": 0
    },
    {
      "type": "module",
      "moduleType": "javascript/auto",
      "identifier": "C:\\webpack5\\src\\title.js",
      "name": "./src/title.js",
      "nameForCondition": "C:\\webpack5\\src\\title.js",
      "index": 1,
      "preOrderIndex": 1,
      "index2": 0,
      "postOrderIndex": 0,
      "size": 25,
      "sizes": {
        "javascript": 25
      },
      "cacheable": true,
      "built": true,
      "codeGenerated": true,
      "cached": false,
      "optional": false,
      "orphan": false,
      "issuer": "C:\\webpack5\\src\\index.js",
      "issuerName": "./src/index.js",
      "issuerPath": [
        {
          "identifier": "C:\\webpack5\\src\\index.js",
          "name": "./src/index.js",
          "profile": {
            "total": 38,
            "resolving": 26,
            "restoring": 0,
            "building": 12,
            "integration": 0,
            "storing": 0,
            "additionalResolving": 0,
            "additionalIntegration": 0,
            "factory": 26,
            "dependencies": 0
          },
          "id": "./src/index.js"
        }
      ],
      "failed": false,
      "errors": 0,
      "warnings": 0,
      "profile": {
        "total": 0,
        "resolving": 0,
        "restoring": 0,
        "building": 0,
        "integration": 0,
        "storing": 0,
        "additionalResolving": 0,
        "additionalIntegration": 0,
        "factory": 0,
        "dependencies": 0
      },
      "id": "./src/title.js",
      "issuerId": "./src/index.js",
      "chunks": ["main"],
      "assets": [],
      "reasons": [
        {
          "moduleIdentifier": "C:\\webpack5\\src\\index.js",
          "module": "./src/index.js",
          "moduleName": "./src/index.js",
          "resolvedModuleIdentifier": "C:\\webpack5\\src\\index.js",
          "resolvedModule": "./src/index.js",
          "type": "cjs require",
          "active": true,
          "explanation": "",
          "userRequest": "./title.js",
          "loc": "1:12-33",
          "moduleId": "./src/index.js",
          "resolvedModuleId": "./src/index.js"
        },
        {
          "moduleIdentifier": "C:\\webpack5\\src\\title.js",
          "module": "./src/title.js",
          "moduleName": "./src/title.js",
          "resolvedModuleIdentifier": "C:\\webpack5\\src\\title.js",
          "resolvedModule": "./src/title.js",
          "type": "cjs self exports reference",
          "active": true,
          "explanation": "",
          "userRequest": null,
          "loc": "1:0-14",
          "moduleId": "./src/title.js",
          "resolvedModuleId": "./src/title.js"
        }
      ],
      "usedExports": null,
      "providedExports": null,
      "optimizationBailout": [
        "CommonJS bailout: module.exports is used directly at 1:0-14"
      ],
      "depth": 1
    }
  ],
  "entrypoints": {

    "main": {
      "name": "main",
      "chunks": ["main"],
      "assets": [
        {
          "name": "main.js",
          "size": 2418
        }
      ],
      "filteredAssets": 0,
      "assetsSize": 2418,
      "auxiliaryAssets": [],
      "filteredAuxiliaryAssets": 0,
      "auxiliaryAssetsSize": 0,
      "children": {},
      "childAssets": {},
      "isOverSizeLimit": false
    }
  },
  "namedChunkGroups": {

    "main": {
      "name": "main",
      "chunks": ["main"],
      "assets": [
        {
          "name": "main.js",
          "size": 2418
        }
      ],
      "filteredAssets": 0,
      "assetsSize": 2418,
      "auxiliaryAssets": [],
      "filteredAuxiliaryAssets": 0,
      "auxiliaryAssetsSize": 0,
      "children": {},
      "childAssets": {},
      "isOverSizeLimit": false
    }
  },
  "errors": [],
  "errorsCount": 0,
  "warnings": [],
  "warningsCount": 0,
  "children": []
}
```