webpack5项目配置

介绍

我们将使用前面所学的知识来从零开始搭建 React-Cli 和 Vue-cli。
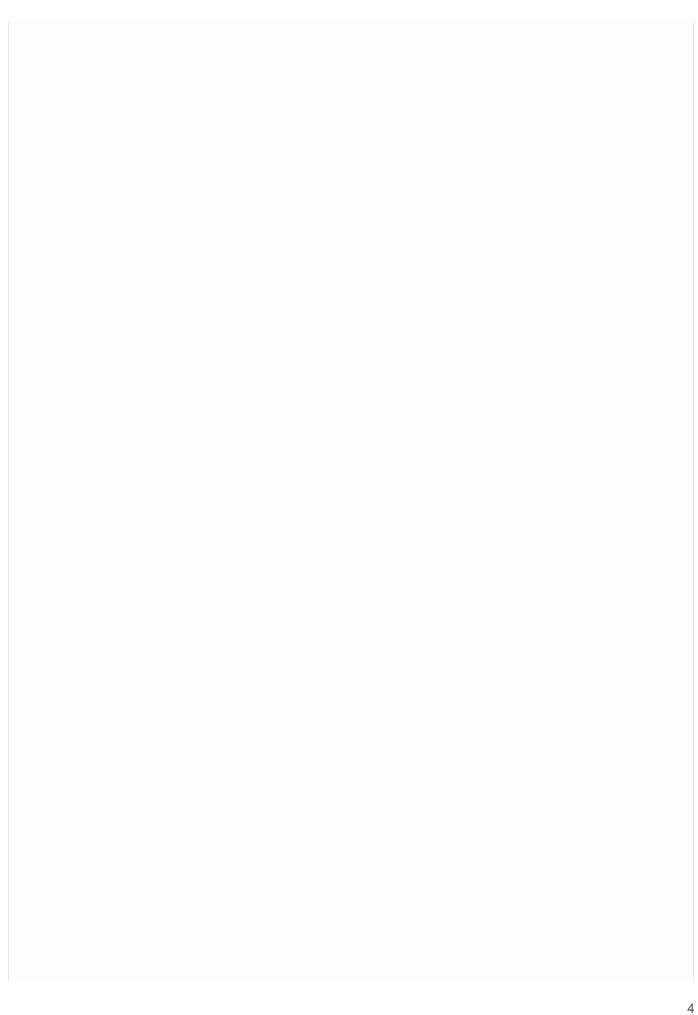
Vue 脚手架
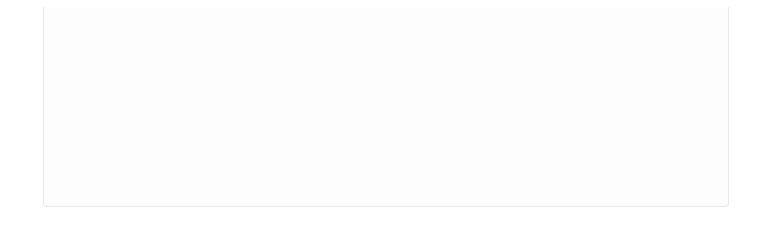
开发模式配置

```javascript
// webpack.dev.js
const path = require("path");
const ESLintWebpackPlugin = require("eslint-webpack-plugin");
const HtmlWebpackPlugin = require("html-webpack-plugin");
const { VueLoaderPlugin } = require("vue-loader");
const { DefinePlugin } = require("webpack");
const CopyPlugin = require("copy-webpack-plugin");

const getStyleLoaders = (preProcessor) => {
  return [
    "vue-style-loader",
    "css-loader",
    {
      loader: "postcss-loader",
      options: {
        postcssOptions: {
          plugins: [
            "postcss-preset-env", // 能解决大多数样式兼容性问题
          ],
        },
      },
    },
    preProcessor,
  ].filter(Boolean);
};

module.exports = {
  entry: "./src/main.js",
  output: {
    path: undefined,
    filename: "static/js/[name].js",
```

生产模式配置

```javascript
// webpack.prod.js
const path = require("path");
const ESLintWebpackPlugin = require("eslint-webpack-plugin");
const HtmlWebpackPlugin = require("html-webpack-plugin");
const MiniCssExtractPlugin = require("mini-css-extract-plugin");
const CssMinimizerPlugin = require("css-minimizer-webpack-plugin");
const TerserWebpackPlugin = require("terser-webpack-plugin");
const ImageMinimizerPlugin = require("image-minimizer-webpack-plugin");
const { VueLoaderPlugin } = require("vue-loader");
const { DefinePlugin } = require("webpack");

const getStyleLoaders = (preProcessor) => {
  return [
    MiniCssExtractPlugin.loader,
    "css-loader",
    {
      loader: "postcss-loader",
      options: {
        postcssOptions: {
          plugins: [
            "postcss-preset-env", // 能解决大多数样式兼容性问题
          ],
        },
      },
    },
    preProcessor,
  ].filter(Boolean);
};

module.exports = {
  entry: "./src/main.js",
  output: {
    path: undefined,
    filename: "static/js/[name].[contenthash:10].js",
    chunkFilename: "static/js/[name].[contenthash:10].chunk.js",
    assetModuleFilename: "static/js/[hash:10][ext][query]",
    clean: true,
  },
  module: {
    rules: [
      {
        // 用来匹配 .css 结尾的文件
        test: /\.css$/,
        // use 数组里面 Loader 执行顺序是从右到左
        use: getStyleLoaders(),
```

```
      },
      {
        test: /\.less$/,
        use: getStyleLoaders("less-loader"),
      },
      {
        test: /\.s[ac]ss$/,
        use: getStyleLoaders("sass-loader"),
      },
      {
        test: /\.styl$/,
        use: getStyleLoaders("stylus-loader"),
      },
      {
        test: /\.(png|jpe?g|gif|svg)$/,
        type: "asset",
        parser: {
          dataUrlCondition: {
            maxSize: 10 * 1024, // 小于10kb的图片会被base64处理
          },
        },
      },
      {
        test: /\.(ttf|woff2?)$/,
        type: "asset/resource",
      },
      {
        test: /\.(jsx|js)$/,
        include: path.resolve(__dirname, "../src"),
        loader: "babel-loader",
        options: {
          cacheDirectory: true,
          cacheCompression: false,
          plugins: [
            // "@babel/plugin-transform-runtime" // presets中包含了
          ],
        },
      },
      // vue-loader不支持oneOf
      {
        test: /\.vue$/,
        loader: "vue-loader", // 内部会给vue文件注入HMR功能代码
        options: {
          // 开启缓存
          cacheDirectory: path.resolve(
            __dirname,
            "node_modules/.cache/vue-loader"
          ),
```

```
        },
      },
    ],
  },
  plugins: [
    new ESLintWebpackPlugin({
      context: path.resolve(__dirname, "../src"),
      exclude: "node_modules",
      cache: true,
      cacheLocation: path.resolve(
        __dirname,
        "../node_modules/.cache/.eslintcache"
      ),
    }),
    new HtmlWebpackPlugin({
      template: path.resolve(__dirname, "../public/index.html"),
    }),
    new CopyPlugin({
      patterns: [
        {
          from: path.resolve(__dirname, "../public"),
          to: path.resolve(__dirname, "../dist"),
          toType: "dir",
          noErrorOnMissing: true,
          globOptions: {
            ignore: ["**/index.html"],
          },
          info: {
            minimized: true,
          },
        },
      ],
    }),
    new MiniCssExtractPlugin({
      filename: "static/css/[name].[contenthash:10].css",
      chunkFilename: "static/css/[name].[contenthash:10].chunk.css",
    }),
    new VueLoaderPlugin(),
    new DefinePlugin({
      __VUE_OPTIONS_API__: "true",
      __VUE_PROD_DEVTOOLS__: "false",
    }),
  ],
  optimization: {
    // 压缩的操作
    minimizer: [
      new CssMinimizerPlugin(),
      new TerserWebpackPlugin(),
```

```
142        new ImageMinimizerPlugin({
143          minimizer: {
144            implementation: ImageMinimizerPlugin.imageminGenerate,
145            options: {
146              plugins: [
147                ["gifsicle", { interlaced: true }],
148                ["jpegtran", { progressive: true }],
149                ["optipng", { optimizationLevel: 5 }],
150                [
151                  "svgo",
152                  {
153                    plugins: [
154                      "preset-default",
155                      "prefixIds",
156                      {
157                        name: "sortAttrs",
158                        params: {
159                          xmlnsOrder: "alphabetical",
160                        },
161                      },
162                    ],
163                  },
164                ],
165              ],
166            },
167          },
168        }),
169      ],
170      splitChunks: {
171        chunks: "all",
172      },
173      runtimeChunk: {
174        name: (entrypoint) => `runtime~${entrypoint.name}`,
175      },
176    },
177    resolve: {
178      extensions: [".vue", ".js", ".json"],
179    },
180    mode: "production",
181    devtool: "source-map",
182  };
```

其他配置

- package.json

```json
{
  "name": "vue-cli",
  "version": "1.0.0",
  "description": "",
  "main": "main.js",
  "scripts": {
    "start": "npm run dev",
    "dev": "cross-env NODE_ENV=development webpack serve --config ./config/webpack.dev.js",
    "build": "cross-env NODE_ENV=production webpack --config ./config/webpack.prod.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "@babel/core": "^7.17.10",
    "@babel/eslint-parser": "^7.17.0",
    "@vue/cli-plugin-babel": "^5.0.4",
    "babel-loader": "^8.2.5",
    "copy-webpack-plugin": "^10.2.4",
    "cross-env": "^7.0.3",
    "css-loader": "^6.7.1",
    "css-minimizer-webpack-plugin": "^3.4.1",
    "eslint-plugin-vue": "^8.7.1",
    "eslint-webpack-plugin": "^3.1.1",
    "html-webpack-plugin": "^5.5.0",
    "image-minimizer-webpack-plugin": "^3.2.3",
    "imagemin": "^8.0.1",
    "imagemin-gifsicle": "^7.0.0",
    "imagemin-jpegtran": "^7.0.0",
    "imagemin-optipng": "^8.0.0",
    "imagemin-svgo": "^10.0.1",
    "less-loader": "^10.2.0",
    "mini-css-extract-plugin": "^2.6.0",
    "postcss-preset-env": "^7.5.0",
    "sass-loader": "^12.6.0",
    "stylus-loader": "^6.2.0",
    "vue-loader": "^17.0.0",
    "vue-style-loader": "^4.1.3",
    "vue-template-compiler": "^2.6.14",
    "webpack": "^5.72.0",
    "webpack-cli": "^4.9.2",
    "webpack-dev-server": "^4.9.0"
  },
```

```
44    "dependencies": {
45      "vue": "^3.2.33",
46      "vue-router": "^4.0.15"
47    },
48    "browserslist": ["last 2 version", "> 1%", "not dead"]
49  }
```

- .eslintrc.js

```javascript
1  module.exports = {
2    root: true,
3    env: {
4      node: true,
5    },
6    extends: ["plugin:vue/vue3-essential", "eslint:recommended"],
7    parserOptions: {
8      parser: "@babel/eslint-parser",
9    },
10 };
```

- babel.config.js

```javascript
1  module.exports = {
2    presets: ["@vue/cli-plugin-babel/preset"],
3  };
```

合并开发和生产配置

```javascript
const path = require("path");
const ESLintWebpackPlugin = require("eslint-webpack-plugin");
const HtmlWebpackPlugin = require("html-webpack-plugin");
const MiniCssExtractPlugin = require("mini-css-extract-plugin");
const CssMinimizerPlugin = require("css-minimizer-webpack-plugin");
const TerserWebpackPlugin = require("terser-webpack-plugin");
const ImageMinimizerPlugin = require("image-minimizer-webpack-plugin");
const { VueLoaderPlugin } = require("vue-loader");
const { DefinePlugin } = require("webpack");
const CopyPlugin = require("copy-webpack-plugin");

// 需要通过 cross-env 定义环境变量
const isProduction = process.env.NODE_ENV === "production";

const getStyleLoaders = (preProcessor) => {
  return [
    isProduction ? MiniCssExtractPlugin.loader : "vue-style-loader",
    "css-loader",
    {
      loader: "postcss-loader",
      options: {
        postcssOptions: {
          plugins: ["postcss-preset-env"],
        },
      },
    },
    preProcessor,
  ].filter(Boolean);
};

module.exports = {
  entry: "./src/main.js",
  output: {
    path: isProduction ? path.resolve(__dirname, "../dist") : undefined,
    filename: isProduction
      ? "static/js/[name].[contenthash:10].js"
      : "static/js/[name].js",
    chunkFilename: isProduction
      ? "static/js/[name].[contenthash:10].chunk.js"
      : "static/js/[name].chunk.js",
    assetModuleFilename: "static/js/[hash:10][ext][query]",
    clean: true,
  },
  module: {
    rules: [
      {
```

```
        // 用来匹配 .css 结尾的文件
        test: /\.css$/,
        // use 数组里面 Loader 执行顺序是从右到左
        use: getStyleLoaders(),
      },
      {
        test: /\.less$/,
        use: getStyleLoaders("less-loader"),
      },
      {
        test: /\.s[ac]ss$/,
        use: getStyleLoaders("sass-loader"),
      },
      {
        test: /\.styl$/,
        use: getStyleLoaders("stylus-loader"),
      },
      {
        test: /\.(png|jpe?g|gif|svg)$/,
        type: "asset",
        parser: {
          dataUrlCondition: {
            maxSize: 10 * 1024, // 小于10kb的图片会被base64处理
          },
        },
      },
      {
        test: /\.(ttf|woff2?)$/,
        type: "asset/resource",
      },
      {
        test: /\.(jsx|js)$/,
        include: path.resolve(__dirname, "../src"),
        loader: "babel-loader",
        options: {
          cacheDirectory: true,
          cacheCompression: false,
          plugins: [
            // "@babel/plugin-transform-runtime" // presets中包含了
          ],
        },
      },
      // vue-loader不支持oneOf
      {
        test: /\.vue$/,
        loader: "vue-loader", // 内部会给vue文件注入HMR功能代码
        options: {
          // 开启缓存
```

14

```
 96            cacheDirectory: path.resolve(
 97              __dirname,
 98              "node_modules/.cache/vue-loader"
 99            ),
100          },
101        },
102      ],
103    },
104    plugins: [
105      new ESLintWebpackPlugin({
106        context: path.resolve(__dirname, "../src"),
107        exclude: "node_modules",
108        cache: true,
109        cacheLocation: path.resolve(
110          __dirname,
111          "../node_modules/.cache/.eslintcache"
112        ),
113      }),
114      new HtmlWebpackPlugin({
115        template: path.resolve(__dirname, "../public/index.html"),
116      }),
117      new CopyPlugin({
118        patterns: [
119          {
120            from: path.resolve(__dirname, "../public"),
121            to: path.resolve(__dirname, "../dist"),
122            toType: "dir",
123            noErrorOnMissing: true,
124            globOptions: {
125              ignore: ["**/index.html"],
126            },
127            info: {
128              minimized: true,
129            },
130          },
131        ],
132      }),
133      isProduction &&
134        new MiniCssExtractPlugin({
135          filename: "static/css/[name].[contenthash:10].css",
136          chunkFilename: "static/css/[name].[contenthash:10].chunk.css",
137        }),
138      new VueLoaderPlugin(),
139      new DefinePlugin({
140        __VUE_OPTIONS_API__: "true",
141        __VUE_PROD_DEVTOOLS__: "false",
142      }),
143    ].filter(Boolean),
```

```
144    optimization: {
145      minimize: isProduction,
146      // 压缩的操作
147      minimizer: [
148        new CssMinimizerPlugin(),
149        new TerserWebpackPlugin(),
150        new ImageMinimizerPlugin({
151          minimizer: {
152            implementation: ImageMinimizerPlugin.imageminGenerate,
153            options: {
154              plugins: [
155                ["gifsicle", { interlaced: true }],
156                ["jpegtran", { progressive: true }],
157                ["optipng", { optimizationLevel: 5 }],
158                [
159                  "svgo",
160                  {
161                    plugins: [
162                      "preset-default",
163                      "prefixIds",
164                      {
165                        name: "sortAttrs",
166                        params: {
167                          xmlnsOrder: "alphabetical",
168                        },
169                      },
170                    ],
171                  },
172                ],
173              ],
174            },
175          },
176        }),
177      ],
178      splitChunks: {
179        chunks: "all",
180      },
181      runtimeChunk: {
182        name: (entrypoint) => `runtime~${entrypoint.name}`,
183      },
184    },
185    resolve: {
186      extensions: [".vue", ".js", ".json"],
187    },
188    devServer: {
189      open: true,
190      host: "localhost",
191      port: 3000,
```

```
192       hot: true,
193       compress: true,
194       historyApiFallback: true, // 解决vue-router刷新404问题
195     },
196     mode: isProduction ? "production" : "development",
197     devtool: isProduction ? "source-map" : "cheap-module-source-map",
198   };
```

优化配置

```javascript
const path = require("path");
const ESLintWebpackPlugin = require("eslint-webpack-plugin");
const HtmlWebpackPlugin = require("html-webpack-plugin");
const MiniCssExtractPlugin = require("mini-css-extract-plugin");
const CssMinimizerPlugin = require("css-minimizer-webpack-plugin");
const ImageMinimizerPlugin = require("image-minimizer-webpack-plugin");
const TerserWebpackPlugin = require("terser-webpack-plugin");
const CopyPlugin = require("copy-webpack-plugin");
const { VueLoaderPlugin } = require("vue-loader");
const { DefinePlugin } = require("webpack");
const AutoImport = require("unplugin-auto-import/webpack");
const Components = require("unplugin-vue-components/webpack");
const { ElementPlusResolver } = require("unplugin-vue-components/resolvers");
// 需要通过 cross-env 定义环境变量
const isProduction = process.env.NODE_ENV === "production";

const getStyleLoaders = (preProcessor) => {
  return [
    isProduction ? MiniCssExtractPlugin.loader : "vue-style-loader",
    "css-loader",
    {
      loader: "postcss-loader",
      options: {
        postcssOptions: {
          plugins: ["postcss-preset-env"],
        },
      },
    },
    preProcessor && {
      loader: preProcessor,
      options:
        preProcessor === "sass-loader"
          ? {
              // 自定义主题：自动引入我们定义的scss文件
              additionalData: `@use "@/styles/element/index.scss" as *;`,
            }
          : {},
    },
  ].filter(Boolean);
};

module.exports = {
  entry: "./src/main.js",
  output: {
```

```
      path: isProduction ? path.resolve(__dirname, "../dist") : undefined,
      filename: isProduction
        ? "static/js/[name].[contenthash:10].js"
        : "static/js/[name].js",
      chunkFilename: isProduction
        ? "static/js/[name].[contenthash:10].chunk.js"
        : "static/js/[name].chunk.js",
      assetModuleFilename: "static/js/[hash:10][ext][query]",
      clean: true,
    },
    module: {
      rules: [
        {
          test: /\.css$/,
          use: getStyleLoaders(),
        },
        {
          test: /\.less$/,
          use: getStyleLoaders("less-loader"),
        },
        {
          test: /\.s[ac]ss$/,
          use: getStyleLoaders("sass-loader"),
        },
        {
          test: /\.styl$/,
          use: getStyleLoaders("stylus-loader"),
        },
        {
          test: /\.(png|jpe?g|gif|svg)$/,
          type: "asset",
          parser: {
            dataUrlCondition: {
              maxSize: 10 * 1024,
            },
          },
        },
        {
          test: /\.(ttf|woff2?)$/,
          type: "asset/resource",
        },
        {
          test: /\.(jsx|js)$/,
          include: path.resolve(__dirname, "../src"),
          loader: "babel-loader",
          options: {
            cacheDirectory: true,
            cacheCompression: false,
```

```javascript
              plugins: [
                // "@babel/plugin-transform-runtime" // presets中包含了
              ],
            },
          },
          // vue-loader不支持oneOf
          {
            test: /\.vue$/,
            loader: "vue-loader", // 内部会给vue文件注入HMR功能代码
            options: {
              // 开启缓存
              cacheDirectory: path.resolve(
                __dirname,
                "node_modules/.cache/vue-loader"
              ),
            },
          },
        ],
      },
      plugins: [
        new ESLintWebpackPlugin({
          context: path.resolve(__dirname, "../src"),
          exclude: "node_modules",
          cache: true,
          cacheLocation: path.resolve(
            __dirname,
            "../node_modules/.cache/.eslintcache"
          ),
        }),
        new HtmlWebpackPlugin({
          template: path.resolve(__dirname, "../public/index.html"),
        }),
        new CopyPlugin({
          patterns: [
            {
              from: path.resolve(__dirname, "../public"),
              to: path.resolve(__dirname, "../dist"),
              toType: "dir",
              noErrorOnMissing: true,
              globOptions: {
                ignore: ["**/index.html"],
              },
              info: {
                minimized: true,
              },
            },
          ],
        }),
```

```
141      isProduction &&
142        new MiniCssExtractPlugin({
143          filename: "static/css/[name].[contenthash:10].css",
144          chunkFilename: "static/css/[name].[contenthash:10].chunk.css",
145        }),
146    new VueLoaderPlugin(),
147    new DefinePlugin({
148      __VUE_OPTIONS_API__: "true",
149      __VUE_PROD_DEVTOOLS__: "false",
150    }),
151    // 按需加载element-plus组件样式
152    AutoImport({
153      resolvers: [ElementPlusResolver()],
154    }),
155    Components({
156      resolvers: [
157        ElementPlusResolver({
158          importStyle: "sass", // 自定义主题
159        }),
160      ],
161    }),
162  ].filter(Boolean),
163  optimization: {
164    minimize: isProduction,
165    // 压缩的操作
166    minimizer: [
167      new CssMinimizerPlugin(),
168      new TerserWebpackPlugin(),
169      new ImageMinimizerPlugin({
170        minimizer: {
171          implementation: ImageMinimizerPlugin.imageminGenerate,
172          options: {
173            plugins: [
174              ["gifsicle", { interlaced: true }],
175              ["jpegtran", { progressive: true }],
176              ["optipng", { optimizationLevel: 5 }],
177              [
178                "svgo",
179                {
180                  plugins: [
181                    "preset-default",
182                    "prefixIds",
183                    {
184                      name: "sortAttrs",
185                      params: {
186                        xmlnsOrder: "alphabetical",
187                      },
188                    },
```

```
                        ],
                    },
                ],
            ],
        },
    },
}),
```