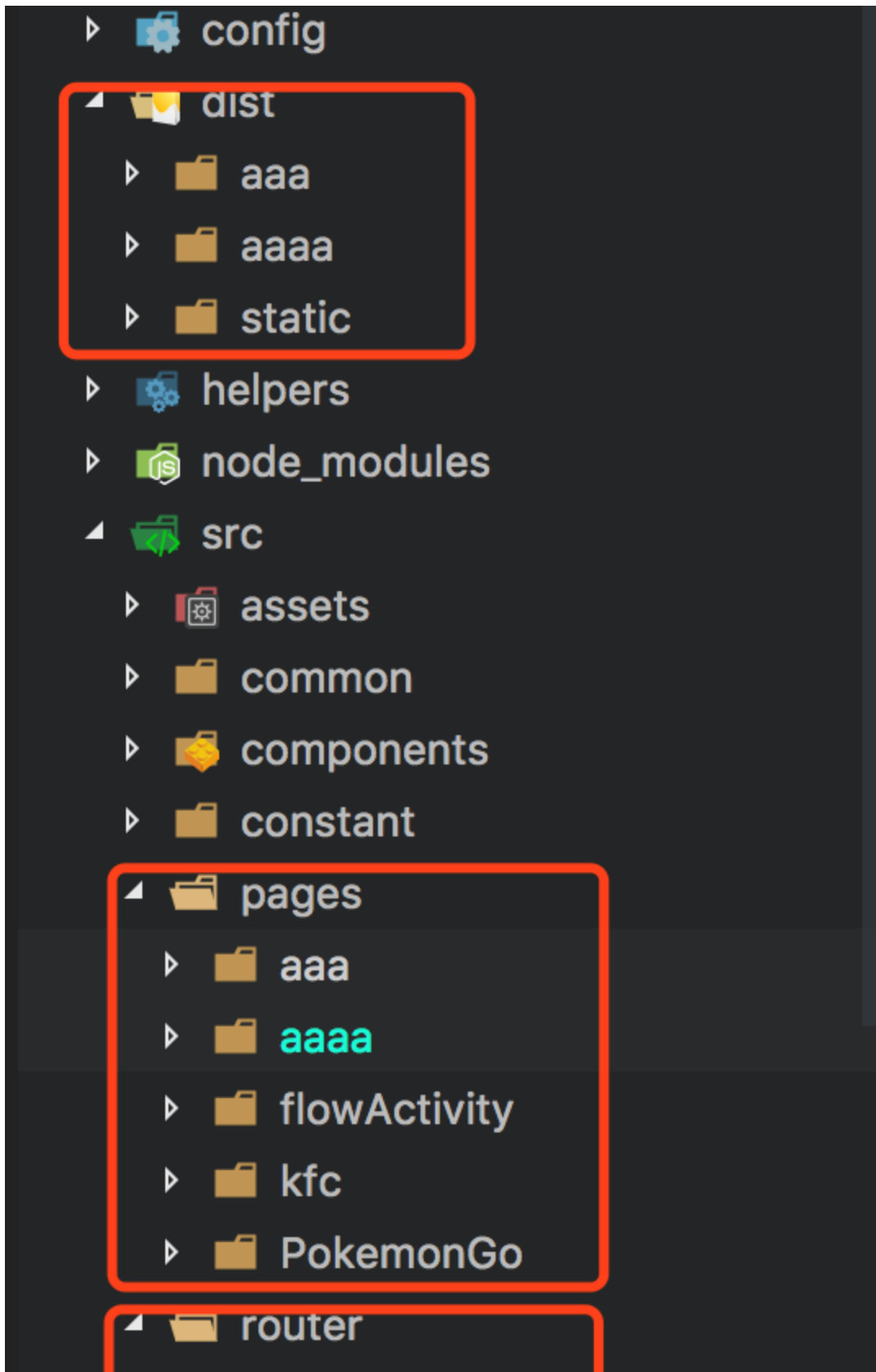
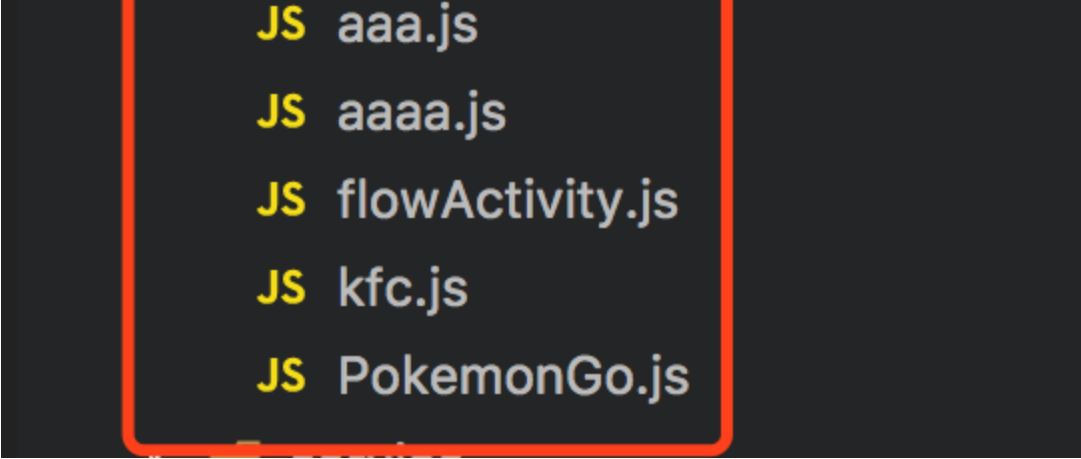


webpack





```
JS aaa.js
JS aaaa.js
JS flowActivity.js
JS kfc.js
JS PokemonGo.js
```

npm install npm link in node

```
in node page --add demo1 //src/pages/demo1/index.js:js,demo1.vuevue-router,home/home.vuesrc/router/demo1.js:
```

```
in node page --add demo1/page1
```

```
  : in node page --del demo1/page1
```

```
    : in node page --del demo1
```

```
  npm run build --project=demo1
```

```
    npm run build --project=demo1,demo2,demo3
```

```
npm run build --report
```

```
npm run devlocalhost8080/demo1/index.htmlindex.htmllocalhost:8080/demo1/index.html#/demo2in node
```

1.

webpack.base.conf.js

```
function getEntries(srcPath, projects) {
  // var files = glob.sync(globPath)
  let projectNames = []
  if(projects && typeof projects === 'string') {
    projectNames = projects.split(',')
  } else {
    projectNames = fs.readdirSync(srcPath)
  }
  let entrys = {}, entry, dirname, basename, pathname, extname, indexJs,
  jsFiles, indexJsPath
  [].forEach.call(projectNames, (project) => {
    indexJsPath = `${srcPath}/${project}/index.js`
    if(fs.existsSync(indexJsPath)) {
      entrys[project] = ['./' + indexJsPath]
    }
  });
  return entrys
}
var entrys = getEntries('src/pages', process.env.npm_config_project)
```

src/pages

## 2. js

```
output: {
  path: config.build.assetsRoot,
  filename: '[name]/static/js/index.[chunkhash].js',
  chunkFilename: '[name]/static/js/[id].[chunkhash].js'
},
```

key[name]assetsRootdist

## 3.css

```
// extract css into its own file
new ExtractTextPlugin({
  filename: '[name]/static/css/index.[contenthash].css'
}),
```

## 4.

dist/static

```
//
new CopyWebpackPlugin([
  {
    from: path.resolve(__dirname, '../static'),
    to: config.build.assetsSubDirectory,
    ignore: ['.*']
  }
])

{
  test: /\. (png|jpe?g|gif|svg) (\?.*)?$/,
  loader: 'url-loader',
  options: {
    limit: 10000,
    context: path.resolve(__dirname, "../src/assets"),
    name: utils.assetsPath('[path]/img/[name].[hash:7].[ext]')
  }
},
{
  test: /\. (mp4|webm|ogg|mp3|wav|flac|aac) (\?.*)?$/,
  loader: 'url-loader',
  options: {
    limit: 10000,
    context: path.resolve(__dirname, "../src/assets"),
    name: utils.assetsPath('[path]/media/[name].[hash:7].[ext]')
  }
},
{
  test: /\. (woff2?|eot|ttf|otf) (\?.*)?$/,
  loader: 'url-loader',
  options: {
    limit: 10000,
    context: path.resolve(__dirname, "../src/assets"),
    name: utils.assetsPath('[path]/fonts/[name].[hash:7].[ext]')
  }
}
}
```

## 5.html

```
var pages = Object.keys(webpackConfig.entry)
pages.forEach(function(pagename) {
  var conf = {
    filename: path.resolve(config.build.assetsRoot, pagename,
'index.html'),
    template: 'index.html',
    inject: true,
    minify: {
      removeComments: true,
      collapseWhitespace: true,
```

```

        removeAttributeQuotes: true
        // more options:
        // https://github.com/kangax/html-minifier#options-quick-reference
    },
    // necessary to consistently work with multiple chunks via
    CommonsChunkPlugin
    chunksSortMode: 'dependency',
    chunks: [`${pagename}-manifest`, `${pagename}-vendor`, pagename]
  }
  webpackConfig.plugins.push(
    new HtmlWebpackPlugin(conf)
  )
  webpackConfig.plugins.push(
    new webpack.optimize.CommonsChunkPlugin({
      name: `${pagename}-vendor`,
      filename: `${pagename}/static/js/vendor.[chunkhash].js`,
      minChunks: function (module, count) {
        // any required modules inside node_modules are extracted to vendor
        return (
          module.resource &&
          /\.js$/.test(module.resource) &&
          module.resource.indexOf(
            path.join(__dirname, '../node_modules')
          ) === 0
        )
      },
      chunks: [pagename],
      // children: true
    })
  )
  webpackConfig.plugins.push(
    // extract webpack runtime and module manifest to its own file in order
    to
    // prevent vendor hash from being updated whenever app bundle is
    updated
    new webpack.optimize.CommonsChunkPlugin({
      name: `${pagename}-manifest`,
      filename: `${pagename}/static/js/manifest.[chunkhash].js`,
      chunks: [`${pagename}-vendor`]
    })
  )
)

```

} )

html