今日目标

1.完成项目优化
2.完成项目上线

1.项目优化

实现步骤：
A.生成打包报告，根据报告优化项目
B.第三方库启用CDN
C.Element-UI组件按需加载
D.路由懒加载
E.首页内容定制

2.添加进度条

给项目添加进度条效果，先打开项目控制台，打开依赖，安装nprogress
打开main.js，编写如下代码

```
//导入进度条插件
import NProgress from 'nprogress'
//导入进度条样式
import 'nprogress/nprogress.css'
.....
//请求在到达服务器之前，先会调用use中的这个回调函数来添加请求头信息
axios.interceptors.request.use(config => {
  //当进入request拦截器，表示发送了请求，我们就开启进度条
  NProgress.start()
  //为请求头对象，添加token验证的Authorization字段
  config.headers.Authorization = window.sessionStorage.getItem("token")
  //必须返回config
  return config
})
//在response拦截器中，隐藏进度条
axios.interceptors.response.use(config =>{
  //当进入response拦截器，表示请求已经结束，我们就结束进度条
  NProgress.done()
  return config
})
```

3.根据报错修改代码

根据ESLint的警告提示更改对应的代码
在.prettierrc文件中更改设置"printWidth":200，将每行代码的文字数量更改为200

```
{
    "semi":false,
    "singleQuote":true,
    "printWidth":200
}
```

4.执行build

安装一个插件（babel-plugin-transform-remove-console）在项目build阶段移除所有的console信息
打开项目控制台，点击依赖->开发依赖，输入babel-plugin-transform-remove-console，安装
打开babel.config.js，编辑代码如下：

```
//项目发布阶段需要用到的babel插件
const productPlugins = []

//判断是开发还是发布阶段
if(process.env.NODE_ENV === 'production'){
  //发布阶段
  productPlugins.push("transform-remove-console")
}

module.exports = {
  "presets": [
    "@vue/app"
  ],
  "plugins": [
    [
      "component",
      {
        "libraryName": "element-ui",
        "styleLibraryName": "theme-chalk"
      }
    ],
    ...productPlugins
  ]
}
```

5.生成打包报告

A.命令行形式生成打包报告
vue-cli-service build --report
B.在vue控制台生成打包报告
点击“任务”=>“build”=>“运行”
运行完毕之后点击右侧“分析”，“控制台”面板查看报告

6.修改webpack的默认配置

默认情况下，vue-cli 3.0生成的项目，隐藏了webpack配置项，如果我们需要配置webpack
需要通过vue.config.js来配置。
在项目根目录中创建vue.config.js文件，

```
module.exports = {
    chainWebpack:config=>{
        //发布模式
        config.when(process.env.NODE_ENV === 'production',config=>{
            //entry找到默认的打包入口，调用clear则是删除默认的打包入口
            //add添加新的打包入口
            config.entry('app').clear().add('./src/main-prod.js')
        })
        //开发模式
        config.when(process.env.NODE_ENV === 'development',config=>{
```

```
            config.entry('app').clear().add('./src/main-dev.js')
        })
    }
}
```

补充：
chainWebpack可以通过链式编程的形式，修改webpack配置
configureWebpack可以通过操作对象的形式，修改webpack配置

7.加载外部CDN

默认情况下，依赖项的所有第三方包都会被打包到js/chunk-vendors.**.js文件中，导致该js文件过大
那么我们可以通过externals排除这些包，使它们不被打包到js/chunk-vendors.**.js文件中

```
module.exports = {
    chainWebpack:config=>{
        //发布模式
        config.when(process.env.NODE_ENV === 'production',config=>{
            //entry找到默认的打包入口，调用clear则是删除默认的打包入口
            //add添加新的打包入口
            config.entry('app').clear().add('./src/main-prod.js')

            //使用externals设置排除项
            config.set('externals',{
                vue:'Vue',
                'vue-router':'VueRouter',
                axios:'axios',
                lodash:'_',
                echarts:'echarts',
                nprogress:'NProgress',
                'vue-quill-editor':'VueQuillEditor'
            })
        })
        //开发模式
        config.when(process.env.NODE_ENV === 'development',config=>{
            config.entry('app').clear().add('./src/main-dev.js')
        })
    }
}
```

设置好排除之后，为了使我们可以使用vue，axios等内容，我们需要加载外部CDN的形式解决引入依赖项。
打开开发入口文件main-prod.js,删除掉默认的引入代码

```
import Vue from 'vue'
import App from './App.vue'
import router from './router'
// import './plugins/element.js'
//导入字体图标
import './assets/fonts/iconfont.css'
//导入全局样式
import './assets/css/global.css'
//导入第三方组件vue-table-with-tree-grid
import TreeTable from 'vue-table-with-tree-grid'
//导入进度条插件
import NProgress from 'nprogress'
```

```
//导入进度条样式
// import 'nprogress/nprogress.css'
// //导入axios
import axios from 'axios'
// //导入vue-quill-editor（富文本编辑器）
import VueQuillEditor from 'vue-quill-editor'
// //导入vue-quill-editor的样式
// import 'quill/dist/quill.core.css'
// import 'quill/dist/quill.snow.css'
// import 'quill/dist/quill.bubble.css'

axios.defaults.baseURL = 'http://127.0.0.1:8888/api/private/v1/'
//请求在到达服务器之前，先会调用use中的这个回调函数来添加请求头信息
axios.interceptors.request.use(config => {
  //当进入request拦截器，表示发送了请求，我们就开启进度条
  NProgress.start()
  //为请求头对象，添加token验证的Authorization字段
  config.headers.Authorization = window.sessionStorage.getItem("token")
  //必须返回config
  return config
})
//在response拦截器中，隐藏进度条
axios.interceptors.response.use(config =>{
  //当进入response拦截器，表示请求已经结束，我们就结束进度条
  NProgress.done()
  return config
})
Vue.prototype.$http = axios

Vue.config.productionTip = false

//全局注册组件
Vue.component('tree-table', TreeTable)
//全局注册富文本组件
Vue.use(VueQuillEditor)

//创建过滤器将秒数过滤为年月日，时分秒
Vue.filter('dateFormat',function(originVal){
  const dt = new Date(originVal)
  const y = dt.getFullYear()
  const m = (dt.getMonth()+1+'').padStart(2,'0')
  const d = (dt.getDate()+'').padStart(2,'0')

  const hh = (dt.getHours()+'').padStart(2,'0')
  const mm = (dt.getMinutes()+'').padStart(2,'0')
  const ss = (dt.getSeconds()+'').padStart(2,'0')

  return `${y}-${m}-${d} ${hh}:${mm}:${ss}`
})

new Vue({
  router,
  render: h => h(App)
}).$mount('#app')
```

然后打开public/index.html添加外部cdn引入代码

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <link rel="icon" href="<%= BASE_URL %>favicon.ico">
    <title>电商后台管理系统</title>

    <!-- nprogress 的样式表文件 -->
    <link rel="stylesheet"
href="https://cdn.staticfile.org/nprogress/0.2.0/nprogress.min.css" />
    <!-- 富文本编辑器 的样式表文件 -->
    <link rel="stylesheet"
href="https://cdn.staticfile.org/quill/1.3.4/quill.core.min.css" />
    <link rel="stylesheet"
href="https://cdn.staticfile.org/quill/1.3.4/quill.snow.min.css" />
    <link rel="stylesheet"
href="https://cdn.staticfile.org/quill/1.3.4/quill.bubble.min.css" />
    <!-- element-ui 的样式表文件 -->
    <link rel="stylesheet" href="https://cdn.staticfile.org/element-
ui/2.8.2/theme-chalk/index.css" />

    <script src="https://cdn.staticfile.org/vue/2.5.22/vue.min.js"></script>
    <script src="https://cdn.staticfile.org/vue-router/3.0.1/vue-router.min.js">
</script>
    <script src="https://cdn.staticfile.org/axios/0.18.0/axios.min.js"></script>
    <script src="https://cdn.staticfile.org/lodash.js/4.17.11/lodash.min.js">
</script>
    <script src="https://cdn.staticfile.org/echarts/4.1.0/echarts.min.js">
</script>
    <script src="https://cdn.staticfile.org/nprogress/0.2.0/nprogress.min.js">
</script>
    <!-- 富文本编辑器的 js 文件 -->
    <script src="https://cdn.staticfile.org/quill/1.3.4/quill.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/vue-quill-editor@3.0.4/dist/vue-
quill-editor.js"></script>

    <!-- element-ui 的 js 文件 -->
    <script src="https://cdn.staticfile.org/element-ui/2.8.2/index.js"></script>

  </head>
  <body>
    <noscript>
      <strong>We're sorry but vue_shop doesn't work properly without JavaScript
enabled. Please enable it to continue.</strong>
    </noscript>
    <div id="app"></div>
    <!-- built files will be auto injected -->
  </body>
</html>
```

8.定制首页内容

开发环境的首页和发布环境的首页展示内容的形式有所不同
如开发环境中使用的是import加载第三方包，而发布环境则是使用CDN，那么首页也需根据环境不同来进行不同的实现
我们可以通过插件的方式来定制首页内容，打开vue.config.js，编写代码如下：

```
module.exports = {
    chainWebpack:config=>{
        config.when(process.env.NODE_ENV === 'production',config=>{
            ......

            //使用插件
            config.plugin('html').tap(args=>{
                //添加参数isProd
                args[0].isProd = true
                return args
            })
        })

        config.when(process.env.NODE_ENV === 'development',config=>{
            config.entry('app').clear().add('./src/main-dev.js')

            //使用插件
            config.plugin('html').tap(args=>{
                //添加参数isProd
                args[0].isProd = false
                return args
            })
        })
    }
}
```

然后在public/index.html中使用插件判断是否为发布环境并定制首页内容

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <link rel="icon" href="<%= BASE_URL %>favicon.ico">
    <title><%= htmlWebpackPlugin.options.isProd ? '' : 'dev - ' %>电商后台管理系统
</title>

    <% if(htmlWebpackPlugin.options.isProd){ %>
    <!-- nprogress 的样式表文件 -->
    <link rel="stylesheet"
href="https://cdn.staticfile.org/nprogress/0.2.0/nprogress.min.css" />
    ........
    <!-- element-ui 的 js 文件 -->
    <script src="https://cdn.staticfile.org/element-ui/2.8.2/index.js"></script>
    <% } %>
  </head>
  .......
```

9.路由懒加载

当路由被访问时才加载对应的路由文件，就是路由懒加载。
路由懒加载实现步骤：
1.安装 @babel/plugin-syntax-dynamic-import
打开vue控制台，点击依赖->安装依赖->开发依赖->搜索@babel/plugin-syntax-dynamic-import
点击安装。

2.在babel.config.js中声明该插件，打开babel.config.js

```
//项目发布阶段需要用到的babel插件
const productPlugins = []

//判断是开发还是发布阶段
if(process.env.NODE_ENV === 'production'){
  //发布阶段
  productPlugins.push("transform-remove-console")
}

module.exports = {
  "presets": [
    "@vue/app"
  ],
  "plugins": [
    [
      "component",
      {
        "libraryName": "element-ui",
        "styleLibraryName": "theme-chalk"
      }
    ],
    ...productPlugins,
    //配置路由懒加载插件
    "@babel/plugin-syntax-dynamic-import"
  ]
}
```

3.将路由更改为按需加载的形式，打开router.js，更改引入组件代码如下：

```
import Vue from 'vue'
import Router from 'vue-router'
const Login = () => import(/* webpackChunkName:"login_home_welcome" */
'./components/Login.vue')
// import Login from './components/Login.vue'
const Home = () => import(/* webpackChunkName:"login_home_welcome" */
'./components/Home.vue')
// import Home from './components/Home.vue'
const Welcome = () => import(/* webpackChunkName:"login_home_welcome" */
'./components/Welcome.vue')
// import Welcome from './components/Welcome.vue'
const Users = () => import(/* webpackChunkName:"user" */
'./components/user/Users.vue')
// import Users from './components/user/Users.vue'
const Rights = () => import(/* webpackChunkName:"power" */
'./components/power/Rights.vue')
// import Rights from './components/power/Rights.vue'
```

```
const Roles = () => import(/* webpackChunkName:"power" */
'./components/power/Roles.vue')
// import Roles from './components/power/Roles.vue'
const Cate = () => import(/* webpackChunkName:"goods" */
'./components/goods/Cate.vue')
// import Cate from './components/goods/Cate.vue'
const Params = () => import(/* webpackChunkName:"goods" */
'./components/goods/Params.vue')
// import Params from './components/goods/Params.vue'
const GoodList = () => import(/* webpackChunkName:"goods" */
'./components/goods/List.vue')
// import GoodList from './components/goods/List.vue'
const GoodAdd = () => import(/* webpackChunkName:"goods" */
'./components/goods/Add.vue')
// import GoodAdd from './components/goods/Add.vue'
const Order = () => import(/* webpackChunkName:"order" */
'./components/order/Order.vue')
// import Order from './components/order/Order.vue'
const Report = () => import(/* webpackChunkName:"report" */
'./components/report/Report.vue')
// import Report from './components/report/Report.vue'
```

###1 0.项目上线

A.通过node创建服务器

在vue_shop同级创建一个文件夹vue_shop_server存放node服务器
使用终端打开vue_shop_server文件夹，输入命令 npm init -y
初始化包之后，输入命令 npm i express -S
打开vue_shop目录，复制dist文件夹，粘贴到vue_shop_server中
在vue_shop_server文件夹中创建app.js文件,编写代码如下：

```
const express = require('express')

const app = express()

app.use(express.static('./dist'))

app.listen(8998,()=>{
    console.log("server running at http://127.0.0.1:8998")
})
```

然后再次在终端中输入 node app.js

B.开启gzip压缩

打开vue_shop_server文件夹的终端，输入命令：npm i compression -D
打开app.js,编写代码：

```
const express = require('express')

const compression = require('compression')

const app = express()

app.use(compression())
app.use(express.static('./dist'))

app.listen(8998,()=>{
    console.log("server running at http://127.0.0.1:8998")
})
```

C.配置https服务

配置https服务一般是后台进行处理，前端开发人员了解即可。
首先，需要申请SSL证书，进入https://freessl.cn官网
在后台导入证书，打开今天资料/素材，复制素材中的两个文件到vue_shop_server中
打开app.js文件，编写代码导入证书，并开启https服务

```
const express = require('express')
const compression = require('compression')
const https = require('https')
const fs = require('fs')

const app = express()
//创建配置对象设置公钥和私钥
const options = {
    cert:fs.readFileSync('./full_chain.pem'),
    key:fs.readFileSync('./private.key')
}

app.use(compression())
app.use(express.static('./dist'))

// app.listen(8998,()=>{
//     console.log("server running at http://127.0.0.1:8998")
// })

//启动https服务
https.createServer(options,app).listen(443)
```

注意：因为我们使用的证书有问题，所以无法正常使用https服务

D.使用pm2管理应用

打开vue_shop_server文件夹的终端，输入命令：npm i pm2 -g
使用pm2启动项目，在终端中输入命令：pm2 start app.js --name 自定义名称
查看项目列表命令：pm2 ls
重启项目：pm2 restart 自定义名称
停止项目：pm2 stop 自定义名称
删除项目：pm2 delete 自定义名称