vue3+vite+ant-design-vue3.x + qiankun打造后台管理系统

项目初始化

项目创建

```
yarn create vite
```

> 会有两个模板，vue和vue-ts模板，根据自己的业务需求选择

项目安装

```
yarn install
```

项目启动

```
yarn dev
```

项目目录

功能

使用Jsx语法

@vitejs/plugin-vue-jsx

```
yarn add @vitejs/plugin-vue-jsx
```

前提：项目中已安装 `@vitejs/plugin-vue`

模板预发配合jsx语法，使用起来非常方便灵活

vite.config.js

```
import { defineConfig } from 'vite'
import vue from '@vitejs/plugin-vue'
import VueJsx from '@vitejs/plugin-vue-jsx'
export default defineConfig({
  plugins: [vue(),VueJsx()]
})
```

启动服务

vue.config.js

```

export default defineConfig({

```

```
  server: {
    open: true,
    port: 7001,
    host: 'localhost',
    headers: {
      'Access-Control-Allow-Origin': '*',
    },
    // 配置代理
    proxy:{
      '/api': {
        target:
'http://jsonplaceholder.typicode.com',
        changeOrigin: true,
        rewrite: (path) =>
path.replace(/^\/api/, '')
      },
    }
  },
})
```

浏览器兼容性处理

@vitejs/plugin-legacy

```
// 支持IE11 浏览器 包含IE11
legacy({

  targets: ['ie >= 11'],
  additionalLegacyPolyfills: ['regenerator-runtime/runtime'],
}),
```

目录别名配置

```
resolve: {
  alias: {
    //src目录的配置
    '@': path.resolve(__dirname, 'src'),
  },
    // 忽略后缀名的配置选项，添加 .vue 选项时要记得原本默认忽略的选项也要手动写入
    extensions: ['.mjs', '.js', '.ts', '.jsx', '.tsx', '.json', '.vue'],
},
```

自动引入组件、组件库

安装

```
yarn add unplugin-auto-import/vite -D
```

```
import AutoImport from 'unplugin-auto-import/vite'
import {
  ElementPlusResolver,
  AntDesignVueResolver,
  VantResolver,
  HeadlessUiResolver,
  ElementUiResolver} from 'unplugin-vue-components'
//配置如下
plugins:[
  // vue vue-router vuex中的方法自动导入
  AutoImport({
    //包含的文件
    include: [
      /\.[tj]sx?$/, // .ts, .tsx, .js, .jsx
      /\.vue$/,
      /\.vue\?vue/, // .vue
      /\.md$/, // .md
    ],
    dts: false, //auto-import.d.ts 使用ts建议开启
    imports: ['vue', 'vue-router', 'vuex'],
  }),
  Components({
```

```
    dts: false, //components.d.ts会自动生成 默认生成在src/
    dirs: ['src/components'],
    extensions: ['vue', 'tsx'],
    resolvers: [
      // $ant-design-prefix
      AntDesignVueResolver({
        importStyle: 'less',
        importLess: true,
      }),
      // ElementPlusResolver(),
      // VantResolver(),
      // HeadlessUiResolver,
      // ElementUiResolver
    ],
  }),
]
```

依赖自动import

```
yarn add unplugin-vue-components -D
```

自定义配置unplugin-vue-components

```
import AutoImport from 'unplugin-auto-import/vite'
```

```
2
3  export default defineConfig({
4    // ...
5    plugins: [
6      AutoImport({
7        imports: [
8          'vue',
9          'vue-router'
10       ],
11       dts: false
12     })
13   ]
14 })
```

ant-design-vue UI组件库集成、自定义主题

安装

```
1  yarn add ant-design-vue
2  yarn add less less-loader -D
```

新建core/lazy_use.js

```
1  // 引入样式
2  import 'ant-design-vue/dist/antd.less'
```

```
import {
  Button,
  message,
  Modal,
} from 'ant-design-vue'

const components = [
  Button,
  message,
  Modal
]
export default {
  install(app) {
    app.config.globalProperties.$message = message
    app.config.globalProperties.$confirm = Modal.confirm
    components.forEach((comp) => {
      app.use(comp)
    })
  },
}

```

main.js

```
import { createApp } from 'vue'
import App from './App.vue'
import lazyUse from './core/lazy_use'

createApp(App).use(lazyUse).mount('#app')
```

vite.config.js

自定义主题

```
export default defineConfig({
  ...
  css: {
    preprocessorOptions: {
      less: {
        modifyVars: {
          // 此处也可设置直角、边框色、字体大小等
          'primary-color': '#0F48B3',
          'link-color': '#073894',
          'border-radius-base': '2px',
          // 配合configProvider组件使用
          // '@ant-prefix': 'platform-ant',
        },
        sourceMap: false,
        javascriptEnabled: true,
      },
    },
```

```
  },

})

```

环境变量定义

项目根目录下分别建立.env.development和.env.production文件

.env.development

```
VITE_APP_BASE_URL=开发环境URL
```

.env.production

```
VITE_APP_BASE_URL=生产环境URL
```

配置package.json

```
{
  scirpts:{
    "dev": "vite --mode development",
    "prod": "vite --mode production",
  }
}
```

重新启动项目

```
yarn dev // 表示开发环境启动
yarn prod // 表示生产环境启动
```

以`yarn dev`为例

vite.config.js环境变量获取

```
import {defineConfig,loadEnv} from 'vite'

export default ({mode,command})=>{
  const env = loadEnv(mode,process.cwd()) // 结果：{VITE_APP_BASE_URL:'开发环境URL'}
  return defineConfig({

  })
}
```

如果项目中使用可以使用`import.meta.env.VITE_APP_BASE_URL`来获取当前环境url

包依赖分析可视化

插件：**rollup-plugin-visualizer**

```
import { defineConfig } from 'vite'
import { visualizer } from 'rollup-plugin-visualizer'
export default {(mode,command)}=>{

```

```
  return defineConfig({
    plugins:[
       visualizer({
        filename: 'report.html', // 生成图表的文件的名称
        open: true, // 自动打开
        gzipSize: true, // 从源代码中收集 gzip 大小并将其显示在图表中
        brotliSize: true, // 从源代码中收集 brotli 大小并将其显示在图表中。
       }),
    ]
  })
}
```

代码压缩

插件：`vite-plugin-compression`

```
import compressPlugin from 'vite-plugin-compression';

compressPlugin({
  ext: '.gz',
  deleteOriginFile: false,
})
```

> 注意：前端通过此插件压缩后，必须打开**nginx**的**gzip**功能才可以

chunk拆包

比如项目下使用**ant-design-vue**或者**echart,**如果想把类似 `ant-design-vue` 这样的包依赖单独拆分出来，也可以手动配置 `manualChunks` 属性

```
// vite.config.ts
build: {
    brotliSize: false, // 压缩大型输出文件可能会很慢，因此禁用该功能可能会提高大型项目的构建性能。默认true
  chunkSizeWarningLimit: 1000, //默认500 超过1000kb发出警告
  rollupOptions: {
    output: {
      manualChunks: configManualChunk
    }
  }
}
```

config/optimizer.js

```
const vendorLibs = [
  {
```

```
    match: ['ant-design-vue'],
    output: 'antdv',
  },
  {
    match: ['echarts'],
    output: 'echarts',
  },
]

export const configManualChunk = (id) => {
  if (/[\\/]node_modules[\\/]/.test(id)) {
    const matchItem = vendorLibs.find((item) => {
      const reg = new RegExp(
        `[\\/]node_modules[\\/]_?(${item.match.join('|')})(.*)`,
        'ig'
      )
      return reg.test(id)
    })
    return matchItem ? matchItem.output : null
  }
}

```

生产环境去除console

```
build:{
  minify:"terser",
    terserOptions: {
      compress: {
        keep_infinity: mode === 'production',
         drop_console: mode === 'production',
      },
    },
}
```

如何全局中使用变量?

externals

vite启动项目慢 如何解决?

vite中如何图片进行优化

vite中如何开启cdn?