# 代理的设置

在开发的过程中，前后端应用不在同一个端口下运行，也就是说不属于同一个进程，由于浏览器的同源策略，不属于同一个端口下的http请求是跨域的，所以不能够正确发出。这个时候就需要使用代理来代替浏览器发请求了，因为同源策略是浏览器才有的，而服务器没有，所以就可以起一个本地服务，浏览器将请求发给本地服务，本地服务再去请求真正的服务器，拿到请求结果之后，再返回给浏览器，就实现了请求的代理功能。具体的实现如下

|  |
| --- |
| *// webpack.config.js*  const AutoImport = require('unplugin-auto-import/webpack')  const Components = require('unplugin-vue-components/webpack')  const { ElementPlusResolver } = require('unplugin-vue-components/resolvers')  const path = require('path');  const baseURL = 'http://127.0.0.1';  module.exports = {    devServer: {      port: 9000, *//本机端口号*      host: "localhost", *//本机主机名*      https: false,      proxy: {        "/upload": {          target: `http://127.0.0.1:8083`,          changOrigin: true,          pathRewrite: {            '^/upload': '/upload'          }        },        "['/userservice', '/authservice', '/adminservice', '/bankservice']": {          target: `${baseURL}:8080`,          changOrigin: true,          pathRewrite: {            '^/userservice': '/userservice',            '^/authservice': '/authservice',            '^/adminservice': '/adminservice',            '^/bankservice': '/bankservice',          }        },        "['/banks', '/questions', '/singleSelections', '/outline']" : {          target: `${baseURL}:8080`,          changOrigin: true        }      }    },    configureWebpack: {      resolve: {        alias: {          '@': path.resolve(\_\_dirname, 'src')        }      },      devtool: 'source-map',      plugins: [        AutoImport({          resolvers: [ElementPlusResolver()],        }),        Components({          resolvers: [ElementPlusResolver()],        }),      ],    }  } |