web.xml

<filter>

<filter-name>CORSFilter</filter-name>

<filter-class>com.ruizton.main.filters.CORSFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>CORSFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

关于跨域

package com.ruizton.main.filters;

import java.io.IOException;

import java.util.Arrays;

import java.util.HashSet;

import java.util.Set;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class CORSFilter implements Filter {

@Override

public void init(FilterConfig var1) throws ServletException {}

@SuppressWarnings("all")

@Override

public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, FilterChain filterChain) throws IOException, ServletException {

HttpServletResponse response = (HttpServletResponse) servletResponse;

String [] allowDomain= {"[http://localhost:8081](http://localhost:8081/)", "http://127.0.0.1:8081"};

Set allowedOrigins= new HashSet(Arrays.asList(allowDomain));

String originHeader=((HttpServletRequest) servletRequest).getHeader("Origin");

if (allowedOrigins.contains(originHeader)) {

response.setHeader("Access-Control-Allow-Origin", originHeader);

}

response.setHeader("Access-Control-Allow-Credentials", "true");

response.setHeader("Access-Control-Allow-Headers", "Origin, No-Cache, X-Requested-With, If-Modified-Since, Pragma, Last-Modified, Cache-Control, Expires, Content-Type, X-E4M-With,userId,token");

response.setHeader("Access-Control-Allow-Methods", "POST, GET, OPTIONS, DELETE");

filterChain.doFilter(servletRequest, servletResponse);

}

public void destroy() {}

}

**java.lang.String.contains()**方法返回true，当且仅当此字符串包含指定的char值序列

String originHeader=((HttpServletRequest) servletRequest).getHeader("Origin"); 当前请求地址