const cheerio = require('cheerio');

const superagen  = require('superagent');

// 1. 利用superagent 发起请求，用set设置请求头，

async function getHtml(url){

  const res = await superagen.get(url)

  .set('Accept', 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3')

  .set('Accept-Encoding','gzip, deflate, br')

  .set('User-Agent','Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.90 Safari/537.36')

  .set('Host','www.govz.cn')

  .set('Referer','https://www.govz.cn/div/4301')

  .set('Sec-Fetch-Mode','navigate')

  .set('Sec-Fetch-Site','same-origin')

  .set('Sec-Fetch-User','?1')

  return res.text;

}

async function main() {

  console.log("get HTML start:\n");

  // 1. 利用superagent 发起请求，用set设置请求头，

  // 注意：需要设置一些辨明身份的请求头，例如host，user-agent

  const html = await getHtml("https://www.govz.cn/div");

  // console.log(JSON.stringify(res));

  // 2. 利用cheerio对响应回来的html文本代码进行解析，加载为jquery能够解析的DOM结构

  let $ = cheerio.load(html);

  // 3. 利用css 选择器去获取html中的文本或数据

  let arr = $("#d\_qh ul li a");

  // 4. 想办法保存起来！

  let result = [];

  // console.log(arr);

  arr.each((idx, el) => {

    console.log("el,", el.attribs.href, el.children[0].data);

    result.push({

      name: el.children[0].data,

      url: el.attribs.href

    });

  });

  // 6. 当需要嵌套请求的时候，考虑使用Promise all

  const promises = result.map(el => {

    return getHtml(el.url);

  });

  const lastJiedaoReqArr = await Promise.all(promises);

  lastJiedaoReqArr.forEach(res => {

    res.text;

  });

  // console.log(JSON.stringify(result));

}

// 执行入口函数

main();