```
// ==UserScript==
// @name         自动屏蔽黑名单公司
// @namespace    https://freysu.github.io/
// @version      1.0
// @description  F**k BlackList Company! Auto Blocking Company and Auto Send Resume!
// @author       FreySu
// @match        *://*.zhipin.com/*
// @match        *://*.zhaopin.com/*
// @match        *://*.liepin.com/*
// @match        *://*.lagou.com/*
// @run-at       document-idle
// @license      Apache Licence 2.0
// @require      https://cdn.staticfile.org/limonte-sweetalert2/8.19.0/sweetalert2.all.min.js
// @require      https://unpkg.com/axios/dist/axios.min.js
// @require      file:///D:/CodeStudy/fuckBlacklistCompany/config_CompanyBlackList.js
// @grant        GM_addStyle
// @grant        GM_xmlhttpRequest
// @grant        GM_setValue
// @grant        GM_getValue
// @grant        GM_listValues
// @grant        GM_registerMenuCommand
// @grant        GM_unregisterMenuCommand
// @grant        GM_addValueChangeListener
// @grant        unsafeWindow
// @connect      https://sa.lagou.com/collect.gif
// ==/UserScript==

/* eslint-disable no-inner-declarations */
/* eslint-disable no-undef */
/* eslint-disable no-lone-blocks */
/* eslint-disable new-cap */
/* eslint-disable no-extra-semi */
/* eslint-disable space-before-function-paren */
/* eslint-disable indent */
;(async function () {
  'use strict'
  // 导入外部CSS样式
  const style = document.createElement('link')
  style.rel = 'stylesheet'
  style.href =
    'https://kjrate.com/wp-content/plugins/anspress-question-answer/templates/css/main.css?ver=4.3.2'
  document.head.appendChild(style)
  // -----------------------------------------------------
  // 配置区
  //
  // #region
  // -----------------------------------------------------
  // prettier-ignore
  const _config_kgCompanyBlackList = Config_kgCompanyBlackList
  // 黑名单数组
  const _config_blackListArr_1 = Config_blackListArr_1
  // 白名单数组
  // prettier-ignore
  const _config_whiteListArr = []
  // #endregion

  /**
   * 勿动
      */
    const _config_jobListDomSelectorArr = [

    {
      el: '#main > div.job-recommend > div.job-list.job-list-new > ul > li > a > div.name > span.company',
      siteType: 1
    },
    {
      el: '.job-card-wrapper .job-card-right .company-name',
      siteType: 1
    },
    {
      el: ".job-list-new .item .name span[class='company']",
      siteType: 1
    },
    {
      el: '.cur.sub-ul-0.sub-ul-box > li > .sub-li > .sub-li-bottom > .user-info > p > .name',
      siteType: 1,
      isList: [
        '.hot-job-box.merge-city-job.common-tab-box > .cur > li > .sub-li > .sub-li-bottom > .user-info > p > .name',
        '.hot-company-wrapper ul li .company-info-top h3'
      ]
    },
    {
      el: '.school-body-wrapper .school-page.company-tab-box li .sub-li .company-title .conpany-text h4',
      siteType: 1,
      isList: ['.cur > li > .sub-li > .user-info > p']
    },
    {
      el: '.hot-pos-buried.hot-pos-wrap.merge-city-job.common-tab-box > .position-list > .cur > li > .sub-li > .sub-li-bottom > .sub-li-bottom-commany-info > .name',
      siteType: 1,
      isList: [
        '.practice-pos-buried.practice-pos.merge-city-job.common-tab-box > .position-list > .cur > li > .sub-li > .sub-li-bottom > .sub-li-bottom-commany-info > .name',
        '.conpany-text > h4'
      ]
    },
    {
      el: '.positionlist .iteminfo__line1__compname__name',
      siteType: 2
    },
    {
      el: `.home-ad-a__card > .home-ad-a__company-info > .home-ad-a__company-info__name`,
      siteType: 2,
      isList: [
        `.home-ad-b__card > .home-ad-b__company-info > .home-ad-b__company-info__name`,
        `.home-ad-c__card > .home-ad-c__company-info--p160.home-ad-c__company-info > .home-ad-c__company-info__title`
      ]
    },
    {
      el: '.job-container__list__item > .list > div.list__item > a > .list__item__content > .list__item__content--company',
      siteType: 2,
      isList: [
        '.industry-mask__title',
        `.zp-company-hiring-list__info__list--name`,
        `.recommand-list__item__content--company.list__item__content--company`
      ]
    },
    {
      el: '.show-text-table-cell',
      siteType: 2,
      isList: [
        '.industry-mask__title',
        `.newest .newset-content ul li div:first-child .des-color`
      ]
    },
    {
      el: '.job-card-pc-container .company-name',
      siteType: 3
    },
    {
      el: '.seo-home-comp-card .home-comp-info .home-comp-info-content .comp-info-title',
      siteType: 3
    },
    {
      el: '.seo-home-comp-card > .home-comp-info > .home-comp-info-content > .ellipsis-1.comp-info-title',
      siteType: 3,
      isList: [
        '.seo-jobcard2-common .jobcard2-company-box .jobcard2-company-info .jobcard2-company-title'
      ]
    },
    {
      el: '.company-list-item-content.jsx-3205826082 .title.jsx-3205826082',
      siteType: 3,
      isList: [
        '.seo-jobcard2-common .jobcard2-company-box .jobcard2-company-info .jobcard2-company-title',
        '.campus-school-jobs-content .campus-school-jobs-item .school-jobs-company .company-info .company-title'
      ]
    }
  ]
  const _config_detailListDomSelectorArr = [
    {
      el: '#__layout > div > div.company-info > div > div.name',
      siteType: 1
    },
    {
      el: '#main > div.job-detail > a > div.info-primary > div > div',
      siteType: 1
    },
    {
      el: '.sider-company .company-info',
      siteType: 1
    },
    {
      el: '.company-banner .name',
      siteType: 1
    },
    {
      el: '.page-gongsi .home-site .company-tab-box li .sub-li .conpany-text h4',
      siteType: 1
    },
    {
      el: '.mian-company .base-info__title h1',
      siteType: 2
    },
    {
      el: '.publisher .company__title',
      siteType: 2
    },
    {
      el: '.company-header-content-name .name-right .title-box .title',
      siteType: 3
    },
    {
      el: '.company-info-container .company-card .content .name',
      siteType: 3
    }
  ]

```

```
const _config_jobListPageUrlArr = [
  {
    url: 'm.zhipin.com/',
    siteType: 1
  },
  {
    url: 'www.zhipin.com/web/geek/job',
    siteType: 1
  },
  {
    url: 'www.zhipin.com/',
    siteType: 1
  },
  {
    url: 'sou.zhaopin.com/',
    siteType: 2
  },
  {
    url: 'www.zhaopin.com/',
    siteType: 2
  },
  {
    url: 'xiaoyuan.zhaopin.com/',
    siteType: 2
  },
  {
    url: 'www.liepin.com/zhaopin/',
    siteType: 3
  },
  {
    url: 'www.liepin.com/',
    siteType: 3
  }
]

const _config_jobDetailPageUrlArr = [
  {
    url: 'www.zhipin.com/job_detail/',
    siteType: 1
  },
  {
    url: 'm.zhipin.com/job_detail/',
    siteType: 1
  },
  {
    url: 'jobs.zhaopin.com/',
    siteType: 2
  },
  {
    url: 'www.liepin.com/job/',
    siteType: 3
  },
  {
    url: 'www.liepin.com/lptjob/',
    siteType: 3
  }
]

const _config_companyDetailPageUrlArr = [
  {
    url: 'm.zhipin.com/gongsi/',
    siteType: 1
  },
  {
    url: 'm.zhipin.com/gongsir/',
    siteType: 1
  },
  {
    url: 'www.zhipin.com/gongsi/',
    siteType: 1
  },
  {
    url: 'www.zhipin.com/gongsir/',
    siteType: 1
  },
  {
    url: 'www.zhaopin.com/companydetail/',
    siteType: 2
  },
  {
    url: 'www.liepin.com/company/',
    siteType: 3
  }
]
var isTimesExhausted_bs = false
var errMsg_bs = ''
var isTimesExhausted_lp = false
var errMsg_lp = ''
var hijacking_enabled = true

// prettier-ignore
```

```
  const userAgents=["Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/86.0.4240.198 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.4577.63 Safari/537.36
Edg/93.0.961.38","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/93.0.4577.63 Safari/537.36 Edg/93.0.961.44","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.4577.63 Safari/537.36
Edg/93.0.961.47","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/93.0.4577.82 Safari/537.36 Edg/93.0.961.52","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.61 Safari/537.36
Edg/94.0.992.31","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/94.0.4606.61 Safari/537.36 Edg/94.0.992.37","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.71 Safari/537.36
Edg/94.0.992.38","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/94.0.4606.81 Safari/537.36 Edg/94.0.992.47","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36
Edg/94.0.992.50","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/95.0.4638.54 Safari/537.36 Edg/95.0.1020.30","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.54 Safari/537.36
Edg/95.0.1020.40","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/95.0.4638.69 Safari/537.36 Edg/95.0.1020.44","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69 Safari/537.36
Edg/95.0.1020.53","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/96.0.4664.45 Safari/537.36 Edg/96.0.1054.29","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.55 Safari/537.36
Edg/96.0.1054.34","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/96.0.4664.55 Safari/537.36 Edg/96.0.1054.41","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.55 Safari/537.36
Edg/96.0.1054.43","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/96.0.4664.93 Safari/537.36 Edg/96.0.1054.53","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.110 Safari/537.36
Edg/96.0.1054.62","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/97.0.4692.71 Safari/537.36 Edg/97.0.1072.55","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.71 Safari/537.36
Edg/97.0.1072.62","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/97.0.4692.99 Safari/537.36 Edg/97.0.1072.69","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.99 Safari/537.36
Edg/97.0.1072.76","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/98.0.4758.80 Safari/537.36 Edg/98.0.1108.43","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.80 Safari/537.36
Edg/98.0.1108.50","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/98.0.4758.102 Safari/537.36 Edg/98.0.1108.55","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102 Safari/537.36
Edg/98.0.1108.56","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/98.0.4758.102 Safari/537.36 Edg/98.0.1108.62","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51 Safari/537.36
Edg/99.0.1150.30","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/99.0.4844.51 Safari/537.36 Edg/99.0.1150.36","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51 Safari/537.36
Edg/99.0.1150.39","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/99.0.4844.74 Safari/537.36 Edg/99.0.1150.46","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74 Safari/537.36
Edg/99.0.1150.52","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/99.0.4844.74 Safari/537.36 Edg/99.0.1150.55","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.60 Safari/537.36
Edg/100.0.1185.29","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/100.0.4896.75 Safari/537.36 Edg/100.0.1185.36","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.75 Safari/537.36
Edg/100.0.1185.39","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/100.0.4896.127 Safari/537.36 Edg/100.0.1185.44","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.127 Safari/537.36
Edg/100.0.1185.50","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/101.0.4951.41 Safari/537.36 Edg/101.0.1210.32","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36
Edg/101.0.1210.39","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/101.0.4951.64 Safari/537.36 Edg/101.0.1210.47","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safari/537.36
Edg/101.0.1210.53","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/102.0.5005.63 Safari/537.36 Edg/102.0.1245.30","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.63 Safari/537.36
Edg/102.0.1245.33","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/102.0.5005.63 Safari/537.36 Edg/102.0.1245.39","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.124 Safari/537.36
Edg/102.0.1245.41","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/102.0.5005.124 Safari/537.36 Edg/102.0.1245.44","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5060.53 Safari/537.36
Edg/103.0.1264.37","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/103.0.5060.66 Safari/537.36 Edg/103.0.1264.44","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5060.114 Safari/537.36
Edg/103.0.1264.49","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/103.0.5060.114 Safari/537.36 Edg/103.0.1264.62","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5060.134 Safari/537.36
Edg/103.0.1264.71","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/103.0.5060.134 Safari/537.36 Edg/103.0.1264.77","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.81 Safari/537.36
Edg/104.0.1293.47","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/104.0.5112.81 Safari/537.36 Edg/104.0.1293.54","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.102 Safari/537.36
Edg/104.0.1293.63","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/104.0.5112.102 Safari/537.36 Edg/104.0.1293.70","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36
Edg/105.0.1343.27","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/105.0.0.0 Safari/537.36 Edg/105.0.1343.33","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36
Edg/105.0.1343.42","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/105.0.0.0 Safari/537.36 Edg/105.0.1343.50","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36
Edg/105.0.1343.53","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/106.0.0.0 Safari/537.36 Edg/106.0.1370.34","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36
Edg/106.0.1370.37","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/106.0.0.0 Safari/537.36 Edg/106.0.1370.42","Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.4577.63
Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/94.0.4606.54 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.61 Safari/537.36","Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.71
Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/94.0.4606.81 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.54 Safari/537.36","Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69
Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/96.0.4664.45 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 Safari/537.36","Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.110
Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/97.0.4692.71 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.99 Safari/537.36","Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.81
Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/98.0.4758.82 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102 Safari/537.36","Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51
Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/99.0.4844.74 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.82 Safari/537.36","Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.60
Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/100.0.4896.75 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.88 Safari/537.36","Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.127
Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/101.0.4951.41 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36","Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64
```

```
Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/101.0.4951.67 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.63 Safari/537.36","Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.115
Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/103.0.5060.53 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5060.66 Safari/537.36","Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5060.114
Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/103.0.5060.134 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.81 Safari/537.36","Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.102
Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/105.0.5195.54 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.5195.102 Safari/537.36","Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.5195.127
Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/106.0.5249.91 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.5249.103 Safari/537.36","Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.41
Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/95.0.4638.17 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.32 Safari/537.36","Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.40
Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/95.0.4638.49 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.18 Safari/537.36","Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.27
Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/96.0.4664.35 Safari/537.36","Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0)
Gecko/20100101 Firefox/92.0","Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:93.0)
Gecko/20100101 Firefox/93.0","Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0)
Gecko/20100101 Firefox/94.0","Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:95.0)
Gecko/20100101 Firefox/95.0","Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:96.0)
Gecko/20100101 Firefox/96.0","Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:97.0)
Gecko/20100101 Firefox/97.0","Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:98.0)
Gecko/20100101 Firefox/98.0","Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:99.0)
Gecko/20100101 Firefox/99.0","Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:100.0)
Gecko/20100101 Firefox/100.0","Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:101.0)
Gecko/20100101 Firefox/101.0","Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0)
Gecko/20100101 Firefox/102.0","Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:103.0)
Gecko/20100101 Firefox/103.0","Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:104.0)
Gecko/20100101 Firefox/104.0","Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0)
Gecko/20100101 Firefox/105.0","Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/39.0.2171.95 Safari/537.36 OPR/26.0.1656.60","Opera/8.0 (Windows NT 5.1;
U; en)","Mozilla/5.0 (Windows NT 5.1; U; en; rv:1.8.1) Gecko/20061208 Firefox/2.0.0 Opera
9.50","Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; en) Opera 9.50","Opera/9.80
(Macintosh; Intel Mac OS X 10.6.8; U; en) Presto/2.8.131 Version/11.11","Opera/9.80 (Windows
NT 6.1; U; en) Presto/2.8.131 Version/11.11","Opera/9.80 (Android 2.3.4; Linux; Opera
Mobi/build-1107180945; U; en-GB) Presto/2.8.149 Version/11.10","Mozilla/5.0 (Windows NT 6.1;
WOW64; rv:34.0) Gecko/20100101 Firefox/34.0","Mozilla/5.0 (X11; U; Linux x86_64; zh-CN;
rv:1.9.2.10) Gecko/20100922 Ubuntu/10.10 (maverick) Firefox/3.6.10","Mozilla/5.0 (Macintosh;
Intel Mac OS X 10.6; rv,2.0.1) Gecko/20100101 Firefox/4.0.1","Mozilla/5.0 (Windows NT 6.1;
rv,2.0.1) Gecko/20100101 Firefox/4.0.1","Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/534.57.2 (KHTML, like Gecko) Version/5.1.7 Safari/534.57.2","MAC：Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.122
Safari/537.36","Windows：Mozilla/5.0 (Windows; U; Windows NT 6.1; en-us) AppleWebKit/534.50
(KHTML, like Gecko) Version/5.1 Safari/534.50","Mozilla/5.0 (iPhone; U; CPU iPhone OS 4_3_3
like Mac OS X; en-us) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8J2
Safari/6533.18.5","Mozilla/5.0 (iPad; U; CPU OS 4_3_3 like Mac OS X; en-us)
AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8J2
Safari/6533.18.5","Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/39.0.2171.71 Safari/537.36","Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.11
(KHTML, like Gecko) Chrome/23.0.1271.64 Safari/537.11","Mozilla/5.0 (Windows; U; Windows NT
6.1; en-US) AppleWebKit/534.16 (KHTML, like Gecko) Chrome/10.0.648.133
Safari/534.16","Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_0) AppleWebKit/535.11 (KHTML,
like Gecko) Chrome/17.0.963.56 Safari/535.11","Mozilla/5.0 (Windows NT 6.1; WOW64)
```

```
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/30.0.1599.101 Safari/537.36","Mozilla/5.0
(Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko","Mozilla/4.0 (compatible; MSIE 7.0;
Windows NT 5.1; 360SE)","Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.11 (KHTML, like
Gecko) Chrome/20.0.1132.11 TaoBrowser/2.0 Safari/536.11","Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/537.1 (KHTML, like Gecko) Chrome/21.0.1180.71 Safari/537.1
LBBROWSER","Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; SLCC2;
.NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C;
.NET4.0E; LBBROWSER)","Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; QQDownload
732; .NET4.0C; .NET4.0E; LBBROWSER)","Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1;
WOW64; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media
Center PC 6.0; .NET4.0C; .NET4.0E; QQBrowser/7.0.3698.400)","Mozilla/4.0 (compatible; MSIE
6.0; Windows NT 5.1; SV1; QQDownload 732; .NET4.0C; .NET4.0E)","Mozilla/5.0 (Windows NT 5.1)
AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.84 Safari/535.11 SE 2.X MetaSr
1.0","Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; SV1; QQDownload 732;
.NET4.0C; .NET4.0E; SE 2.X MetaSr 1.0)","Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1;
Trident/4.0; SE 2.X MetaSr 1.0; SE 2.X MetaSr 1.0; .NET CLR 2.0.50727; SE 2.X MetaSr
1.0)","Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Maxthon/4.4.3.4000 Chrome/30.0.1599.101 Safari/537.36","Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/38.0.2125.122 UBrowser/4.0.3214.0
Safari/537.36","Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/55.0.2883.87 UBrowser/6.2.4094.1 Safari/537.36","Mozilla/5.0 (iPod; U; CPU iPhone OS
4_3_3 like Mac OS X; en-us) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8J2
Safari/6533.18.5","Mozilla/5.0 (iPad; U; CPU OS 4_2_1 like Mac OS X; zh-cn)
AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8C148
Safari/6533.18.5","Mozilla/5.0 (Linux; U; Android 2.2.1; zh-cn; HTC_Wildfire_A3333
Build/FRG83D) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile
Safari/533.1","Mozilla/5.0 (Linux; U; Android 2.3.7; en-us; Nexus One Build/FRF91)
AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1","MQQBrowser/26
Mozilla/5.0 (Linux; U; Android 2.3.7; zh-cn; MB200 Build/GRJ22; CyanogenMod-7)
AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1","Mozilla/5.0 (Linux;
U; Android 3.0; en-us; Xoom Build/HRI39) AppleWebKit/534.13 (KHTML, like Gecko) Version/4.0
Safari/534.13","Mozilla/5.0 (BlackBerry; U; BlackBerry 9800; en) AppleWebKit/534.1+ (KHTML,
like Gecko) Version/6.0.0.337 Mobile Safari/534.1+","Mozilla/5.0 (hp-tablet; Linux;
hpwOS/3.0.0; U; en-US) AppleWebKit/534.6 (KHTML, like Gecko) wOSBrowser/233.70 Safari/534.6
TouchPad/1.0","Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;","Mozilla/4.0
(compatible; MSIE 7.0; Windows NT 6.0)","Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0;
Trident/4.0)","Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)","Mozilla/4.0 (compatible;
MSIE 7.0; Windows NT 5.1)","Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; The
World)","Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; TencentTraveler
4.0)","Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Avant Browser)","Mozilla/5.0
(SymbianOS/9.4; Series60/5.0 NokiaN97-1/20.0.019; Profile/MIDP-2.1 Configuration/CLDC-1.1)
AppleWebKit/525 (KHTML, like Gecko) BrowserNG/7.1.18124","Mozilla/5.0 (compatible; MSIE 9.0;
Windows Phone OS 7.5; Trident/5.0; IEMobile/9.0; HTC;
Titan)","UCWEB7.0.2.37/28/999","NOKIA5700/ UCWEB7.0.2.37/28/999","Openwave/
UCWEB7.0.2.37/28/999"];

  const debug = true

  /**
   * 通知
      */
    const myToast = {

    /**
     * @description toast通知
     * @param {string} type "success", "error", "warning", "info" or "question"
     * @param {string} title 标题
     * @param {number} timer 显示时间
     * @param {string} html html类型内容，选填
     * @param {string} position 显示位置
     */
    normal: function (type, title, timer, position = 'center', html = null) {
      const Toast = Swal.mixin({
        toast: true,
        position: position,
        showConfirmButton: false,
        showCloseButton: true
      })
      Toast.fire({
        type,
        title,
        html,
        timer
      })
    },
    /**
     * @description 不控制的常规弹窗
     * @param {string} type "success","error","warning","info" or "question"
     * @param {string} title 标题
     * @param {number} timer 显示时间
     * @param {boolean} isHtml text or html
     * @param {string} content 内容
     */
    multipleNotControl: function (
      type = null,
      title,
      timer,
      isHtml,
      content,
      confirmCallBack
    ) {
      var text, html
      if (isHtml) {
        text = null
        html = content
      } else {
        text = content
        html = null
      }
      Swal.fire({
        type,
        title,
        text,
        timer,
        html,
        confirmButtonText: '我知道了'
      }).then((result) => {
        confirmCallBack(result)
      })
    },
    multipleControl: function (
      type = null,
      title,
      timer,
      isHtml,
      content,
      confirmCallBack
    ) {
      var text, html
      if (isHtml) {
        text = null
        html = content
      } else {
        text = content
        html = null
      }
      Swal.fire({
        type,
        title,
        text,
        timer,
        html,
        confirmButtonText: '我知道了',
        allowOutsideClick: false,
        allowEscapeKey: false,
        allowEnterKey: false
      }).then((result) => {
        confirmCallBack(result)
      })
    }
  }

  /**
   * 工具库
      */
    const myUtils = {

    /**
     *
     * @param {object} parent 被监听的元素的父节点，也可以是document
     * @param {Node} selector 被监听的dom元素
     * @param {Number} timeout 超时跳出时间，默认为0，则等到出现为止，大于0才超时跳出
     * @returns
     */
    getElement: function (parent, selector, timeout = 0) {
      return new Promise((resolve) => {
        let result = parent.querySelector(selector)
        if (result) return resolve(result)
        let timer
        const mutationObserver =
          window.MutationObserver ||
          window.WebkitMutationObserver ||
          window.MozMutationObserver
        if (mutationObserver) {
          const observer = new mutationObserver((mutations) => {
            for (const mutation of mutations) {
              for (const addedNode of mutation.addedNodes) {
                if (addedNode instanceof Element) {
                  result = addedNode.matches(selector)
                    ? addedNode
                    : addedNode.querySelector(selector)
                  if (result) {
                    observer.disconnect()
                    timer && clearTimeout(timer)
                    return resolve(result)
                  }
                }
              }
            }
          })
          observer.observe(parent, {
            childList: true,
            subtree: true
          })
          if (timeout > 0) {
            timer = setTimeout(() => {
              observer.disconnect()
              return resolve(null)
            }, timeout)
          }
        } else {
          const listener = (e) => {
            if (e.target instanceof Element) {
              result = e.target.matches(selector)
                ? e.target
                : e.target.querySelector(selector)
              if (result) {
                parent.removeEventListener('DOMNodeInserted', listener, true)
                timer && clearTimeout(timer)
                return resolve(result)
              }
            }
          }
          parent.addEventListener('DOMNodeInserted', listener, true)
          if (timeout > 0) {
            timer = setTimeout(() => {
              parent.removeEventListener('DOMNodeInserted', listener, true)
              return resolve(null)
            }, timeout)
          }
        }
      })
    },
    uniqueFunc: (arr, uniId) => {
      // const res = new Map()
      // return arr.filter(
      //   (item) => !res.has(item[uniId]) && res.set(item[uniId], 1)
      // )
      if (!arr.length) return arr
      const repeatArr = new Set()
      if (uniId) {
        return arr.reduce((pre, cur) => {
          let item = null
          if (typeof uniId === 'string' && uniId) {
            item = cur[uniId]
          } else {
            item = typeof cur === 'object' ? JSON.stringify(cur) : cur
          }
          if (!repeatArr.has(item)) {
            repeatArr.add(item)
            pre.push(cur)
          }
          return pre
        }, [])
      }
    },
    uniqueArr: (arr) => Array.from(new Set(arr)),
    sleep: function (duration) {
      return new Promise((resolve) => {
        setTimeout(resolve, duration)
      })
    },
    /**
     *
     * @param {Array} toArr 要分割的数组
     * @param {Number} eqLen 等长值
     * @returns newArr
     */
    sliceArrLikeBeEQLengthGroup: (toArr, eqLen) => {
      let index = 0
      const newArr = []
      while (index < toArr.length) {
        newArr.push(toArr.slice(index, (index += eqLen)))
      }
      return newArr
    },
    /**
     * async/await 的错误处理方法 1. try...catch 2. async/await 本质就是 promise 的语法糖
     * @param {promise} promise 要执行的promise
     * @returns {array} [err,rs] err:错误码，data:结果
     */
    awaitWrap: function (promise) {
      return promise.then((data) => [null, data]).catch((err) => [err, null])
    },
    debounce: function (fn, delay) {
      let timeout = null
      return function () {
        const args = arguments
```

```
        if (timeout) window.clearTimeout(timeout)
        timeout = setTimeout(() => {
          fn.apply(this, args)
        }, delay)
      }
    },
    // 通过判断图片是否加载成功来判断网络通断
    onLine: function () {
      // 基于promise，可以用async和await
      var img = new Image()
      img.src = 'https://www.baidu.com/favicon.ico?_t=' + Date.now()
      return new Promise((resolve, reject) => {
        img.onload = function () {
          resolve(true)
        }
        img.onerror = function () {
          resolve(false)
        }
      })
    },
    /**
     * 从数组中查找出模糊或完全符合的内容
     * @param sth 要查找的内容
     * @param arr 被查找的数组
     * @param name 被查找的属性名
     * @returns Array
     */
    findSthFromObjArr: (sth, arr, name) => {
      return (
        arr.filter((item) => {
          return sth.indexOf(item[name]) !== -1 || sth === item[name]
        }) || []
      )
    },
    /**
     * @description: 正则表达式根据cookie的key匹配cookie的value
     * @param {string} key
     * @return {*}
     */
    getCookie: function (key) {
      const str = `(^| )${key}=([^;]*)(;|$)`
      const reg = new RegExp(str)
      const arr = document.cookie.match(reg)
      if (!arr) {
        return null
      }
      return arr[2] // 第2个分组匹配对应cookie的value
    },
    randomSleep: async function (number, number1, fn = null) {
      var a = Math.round(Math.random() * (number1 - number) + number) * 1000
      debug && console.log(`即将等待${+(a / 1000)}秒(${number}~${number1}秒)`)
      // GM_getValue('allSettings')['showHint_enabled'] &&
      //   myToast.normal(
      //     'success',
      //     `即将等待${+(a / 1000)}秒(${number}~${number1}秒)`,
      //     1000,
      //     'center'
      //   )
      await this.sleep(a)
      return fn ? fn() : null
    },
    checkName: function (name) {
      const reg = new RegExp(
        "[`~!@#$^&*()=|{}':;',\\[\\].<>/?~！@#￥……&*（）——|{}【】‘；：”“'。，、？]"
      )
      let newName = ''
      for (var i = 0, l = name.length; i < name.length; i++) {
```

```
      newName = newName + name.substr(i, 1).replace(reg, '')
    }
    return newName
  }
}

const _config_blackListArr = myUtils.uniqueArr(
  _config_kgCompanyBlackList.concat(_config_blackListArr_1)
)

/**
 * 悬浮窗样式
    */
  GM_addStyle(

  `#TManays{z-index:999999; position:absolute; left:0px; top:0px; width:100px; height:auto; border:0; margin:0;}#parseUl{position:fixed;top:80px; left:0px;}#parseUl li{list-style:none;}.TM1{opacity:0.3; position:relative;padding: 0 7px 0 0; min-width: 19px; cursor:pointer;}.TM1:hover{opacity:1;}.TM1 span{display:block; border-radius:0 5px 5px 0; background-color:#ffff00; border:0; font:bold 15px "微软雅黑" !important; color:#ff0000; margin:0; padding:15px 2px;}.TM3{position:absolute; top:0; left:19px; display:none; border-radius:5px; margin:0; padding:0;}.TM3 li{float:none; width:80px; margin:0; font-size:14px; padding:3px 10px 2px 15px; cursor:pointer; color:#3a3a3a !important; background:rgba(255,255,0,0.8)}.TM3 li:hover{color:white !important; background:rgba(0,0,0,0.8);}.TM3 li:last-child{border-radius: 0 0 5px 5px;}.TM3 li:first-child{border-radius: 5px 5px 0 0;}.TM1:hover .TM3{display:block}`
)

/**
 * 以下值，不要修改！
    */
  GM_setValue('_jobListJson_boss', null)
  GM_setValue('_zp_token', null)
  GM_setValue('_zl_currentJobNumbers', null)
  // GM_setValue('_zl_joblist', null)
  GM_setValue('_zl_params_details', null)
  GM_setValue('_zl_params_cityIds', null)
  GM_setValue('_lp_params_resId', null)
  GM_setValue('_lp_params_jobCardList', null)
  GM_setValue('_lg_params', null)
  GM_setValue('_lg_resumeId', null)
  GM_setValue('_lg_positionIds', null)

/**
 * 适配不同浏览器的匹配选择器
    */
  Element.prototype.matches = Element.prototype.matches || ``
  Element.prototype.matchesSelector ||

  Element.prototype.webkitMatchesSelector ||
  Element.prototype.msMatchesSelector ||
  Element.prototype.mozMatchesSelector

var curl = location.href

var isShowTip
if (GM_listValues().indexOf('allSettings') === -1) {
  GM_setValue('allSettings', {})
}

var curSettings = GM_getValue('allSettings')
// 设置自动识别初始值
if (curSettings['autoBlocking_enabled'] === undefined) {
  curSettings['autoBlocking_enabled'] = true
  GM_setValue('allSettings', curSettings)
}

```

```
  // 设置开启提示初始值
  if (curSettings['showHint_enabled'] === undefined) {
    curSettings['showHint_enabled'] = true
    GM_setValue('allSettings', curSettings)
    isShowTip = GM_getValue('allSettings')['showHint_enabled']
  }

  // 设置隐藏或提醒黑名单公司
  if (curSettings['isShowBlockingCompany'] === undefined) {
    curSettings['isShowBlockingCompany'] = true
    GM_setValue('allSettings', curSettings)
  }

  // 是否开启自动发送简历功能
  if (curSettings['autoSendResume_enabled'] === undefined) {
    curSettings['autoSendResume_enabled'] = false
    GM_setValue('allSettings', curSettings)
  }

  var autoBlocking_enabled_menuId
  var showHint_enabled_menuId
  var isShowBlockingCompany_menuId
  var autoSendResume_enabled_menuId
  // 刷新新菜单
  function registerMenu() {
    // 是否开启自动屏蔽黑名单功能
    GM_unregisterMenuCommand(autoBlocking_enabled_menuId)
    if (curSettings['autoBlocking_enabled']) {
      autoBlocking_enabled_menuId = GM_registerMenuCommand(
        '自动屏蔽当前页面公司 [状态：√ ]',
        function () {
          curSettings['autoBlocking_enabled'] = false
          GM_setValue('allSettings', curSettings)
          registerMenu()
          GM_getValue('allSettings')['showHint_enabled'] &&
            myToast.normal(
              'success',
              '已关闭自动屏蔽当前页面公司功能，遇到新网站请自行手动添加规则！',
              3000
            )
        }
      )
    } else {
      autoBlocking_enabled_menuId = GM_registerMenuCommand(
        '自动屏蔽当前页面公司 [状态：× ]',
        function () {
          curSettings['autoBlocking_enabled'] = true
          GM_setValue('allSettings', curSettings)
          registerMenu()
          GM_getValue('allSettings')['showHint_enabled'] &&
            myToast.normal(
              'success',
              '已开启自动屏蔽当前页面公司功能，请刷新网页！',
              3000
            )
        }
      )
    }

    // 各类提示
    GM_unregisterMenuCommand(showHint_enabled_menuId)
    if (curSettings['showHint_enabled']) {
      showHint_enabled_menuId = GM_registerMenuCommand(
        '提示信息 [状态：√ ]',
        function () {
          GM_getValue('allSettings')['showHint_enabled'] &&
            myToast.normal(
              'success',
              '提示功能已关闭，再次开启前将无任何提示！',
              3000
            )
          curSettings['showHint_enabled'] = false
          GM_setValue('allSettings', curSettings)
          isShowTip = GM_getValue('allSettings')['showHint_enabled']
          registerMenu()
        }
      )
    } else {
      showHint_enabled_menuId = GM_registerMenuCommand(
        '提示信息 [状态：× ]',
        function () {
          curSettings['showHint_enabled'] = true
          GM_setValue('allSettings', curSettings)
          isShowTip = GM_getValue('allSettings')['showHint_enabled']
          registerMenu()
          GM_getValue('allSettings')['showHint_enabled'] &&
            myToast.normal('success', '提示功能已开启！', 3000)
        }
      )
    }
    // 提示音
    GM_unregisterMenuCommand(isShowBlockingCompany_menuId)
    if (curSettings['isShowBlockingCompany']) {
      isShowBlockingCompany_menuId = GM_registerMenuCommand(
        '显示 [状态：√ ]',
        function () {
          GM_getValue('allSettings')['showHint_enabled'] &&
            myToast.normal('success', '隐藏！', 3000)
          curSettings['isShowBlockingCompany'] = false
          GM_setValue('allSettings', curSettings)
          registerMenu()
        }
      )
    } else {
      isShowBlockingCompany_menuId = GM_registerMenuCommand(
        '隐藏 [状态：× ]',
        function () {
          curSettings['isShowBlockingCompany'] = true
          GM_setValue('allSettings', curSettings)
          registerMenu()
          GM_getValue('allSettings')['showHint_enabled'] &&
            myToast.normal('success', '显示！', 3000)
        }
      )
    }
    // 自动投简历
    GM_unregisterMenuCommand(autoSendResume_enabled_menuId)
    if (curSettings['autoSendResume_enabled']) {
      autoSendResume_enabled_menuId = GM_registerMenuCommand(
        '自动投简历 [状态：√ ]',
        function () {
          GM_getValue('allSettings')['showHint_enabled'] &&
            myToast.normal('success', '自动投简历功能已关闭！', 3000)
          curSettings['autoSendResume_enabled'] = false
          GM_setValue('allSettings', curSettings)
          registerMenu()
        }
      )
    } else {
      autoSendResume_enabled_menuId = GM_registerMenuCommand(
        '自动投简历 [状态：× ]',
        function () {
          curSettings['autoSendResume_enabled'] = true
          GM_setValue('allSettings', curSettings)
          registerMenu()
          GM_getValue('allSettings')['showHint_enabled'] &&
            myToast.normal('success', '自动投简历功能已开启！', 3000)
        }
      )
    }
  }

  registerMenu()
  // var lg_isOpenGetCollectGif = true
  var curPageUrlArr_jobList = myUtils.findSthFromObjArr(
    curl,
    _config_jobListPageUrlArr,
    'url'
  )

  var curPageUrlArr_jobDetail = myUtils.findSthFromObjArr(
    curl,
    _config_jobDetailPageUrlArr,
    'url'
  )

  var curPageUrlArr_companyDetail = myUtils.findSthFromObjArr(
    curl,
    _config_companyDetailPageUrlArr,
    'url'
  )

  function addXMLRequestCallback(callback) {
    var oldSend, i, setRequestHeader

    if (XMLHttpRequest.callbacks) {
      XMLHttpRequest.callbacks.push(callback)
    } else {
      XMLHttpRequest.callbacks = [callback]

      oldSend = XMLHttpRequest.prototype.send

      XMLHttpRequest.prototype.send = function () {
        for (i = 0; i < XMLHttpRequest.callbacks.length; i++) {
          XMLHttpRequest.callbacks[i](this)
        }
        try {
          oldSend.apply(this, arguments)
        } catch (error) {
          console.log('error: ', error)
        }
      }
      setRequestHeader = XMLHttpRequest.prototype.setRequestHeader
      XMLHttpRequest.prototype.requestHeader = {}
      XMLHttpRequest.prototype.setRequestHeader = function (name, value) {
        XMLHttpRequest.prototype.requestHeader[name] = value
        try {
          setRequestHeader.call(this, name, value)
        } catch (error) {
          console.log('error: ', error)
        }
      }
    }
  }

  if (GM_getValue('allSettings')['autoSendResume_enabled']) {
    isShowTip &&
      myToast.normal('success', '你已开启自动投简历！', 3000, 'center') &&
      console.log('success', '你已开启自动投简历！', 3000, 'center')
  }

```

```
  function autoSend() {
    if (
      hijacking_enabled &&
      (curPageUrlArr_jobList.length ||
        curPageUrlArr_jobDetail.length ||
        curPageUrlArr_companyDetail.length)
    ) {
      console.log(1111)
      addXMLRequestCallback(function (xhr) {
        xhr.addEventListener('load', function () {
          if (xhr.readyState === 4 && xhr.status === 200) {
            var fRes
            var resUrl = new URL(xhr.responseURL)
            if (resUrl.origin.indexOf('apic.liepin.com') !== -1) {
              if (
                resUrl.pathname ===
                '/api/com.liepin.usercx.pc.user.base-property'
              ) {
                if (isTimesExhausted_lp) {
                  debug && console.log(errMsg_lp)
                  isShowTip &&
                    myToast.normal('error', errMsg_lp, 3000, 'center')
                  return false
                }
                fRes = JSON.parse(xhr.responseText)
                // GM_getValue('allSettings')['autoSendResume_enabled'] &&
                GM_sendMessage('_lp_params_resId', fRes.data.resId)
              } else if (
                resUrl.pathname ===
                '/api/com.liepin.searchfront4c.pc-search-job'
              ) {
                if (isTimesExhausted_lp) {
                  debug && console.log(errMsg_lp)
                  isShowTip &&
                    myToast.normal('error', errMsg_lp, 3000, 'center')
                  return false
                }
                fRes = JSON.parse(xhr.responseText)
                // GM_getValue('allSettings')['autoSendResume_enabled'] &&
                GM_sendMessage(
                  '_lp_params_jobCardList',
                  fRes.data.data.jobCardList
                )
              }
            } else if (resUrl.origin.indexOf('zhipin.com') !== -1) {
              if (isTimesExhausted_bs) {
                debug && console.log(errMsg_bs)
                isShowTip && myToast.normal('error', errMsg_bs, 3000, 'center')
                fRes = JSON.parse(xhr.responseText)
                return false
              } else if (
                resUrl.pathname == '/wapi/zpgeek/search/joblist.json'
              ) {
                fRes = JSON.parse(xhr.responseText)
                // GM_getValue('allSettings')['autoSendResume_enabled'] &&
                GM_sendMessage('_jobListJson_boss', fRes.zpData.jobList)
              } else if (resUrl.pathname == '/wapi/zppassport/get/zpToken') {
                console.log(222)
                try {
                  fRes = JSON.parse(xhr.responseText)
                  // GM_getValue('allSettings')['autoSendResume_enabled'] &&
                  GM_sendMessage('_zp_token', fRes.zpData.token)
                } catch (error) {
                  console.error('error: ', error)
                }
              } else if (
                resUrl.pathname == '/wapi/zpgeek/mobile/search/joblist.json'
              ) {
                fRes = JSON.parse(xhr.responseText)
                // GM_getValue('allSettings')['autoSendResume_enabled'] &&
                GM_sendMessage('_jobListJson_boss', fRes.zpData.html)
                handleNewPageElements(2, 1, 500)
              }
            } else if (resUrl.origin.indexOf('fe-api.zhaopin.com') !== -1) {
              if (
                resUrl.pathname === '/c/i/user/detail'
                // || resUrl.pathname === '/c/i/city-page/user-city'
              ) {
                fRes = JSON.parse(xhr.responseText)
                // GM_getValue('allSettings')['autoSendResume_enabled'] &&
                GM_sendMessage('_zl_params_details', {
                  _resumeNumber: fRes.data.Resume.ResumeNumber,
                  _at: resUrl.searchParams.get('at'),
                  _rt: resUrl.searchParams.get('rt'),
                  _x_zp_client_id: resUrl.searchParams.get('x-zp-client-id'),
                  _x_zp_page_request_id: resUrl.searchParams.get(
                    'x-zp-page-request-id'
                  )
                })
              } else if (
                resUrl.pathname === '/c/i/jobs/keyword-company-search' &&
                resUrl.searchParams.get('filter_c_workCity') !== null
              ) {
                fRes = JSON.parse(xhr.responseText)
                // GM_getValue('allSettings')['autoSendResume_enabled'] &&
                GM_sendMessage(
                  '_zl_params_cityIds',
                  resUrl.searchParams.get('filter_c_workCity')
                )
              }
            }
          }
        })
      })
    }
  }
  autoSend()

  console.log({
    msg: '注入完成',
    time: new Date().valueOf(),
    data: new Date()
  })

  /**
   * 监听翻页-history模式
      */
    const _historyWrap = function (type) {

    const orig = history[type]
    const e = new Event(type)
    return function () {
      const rv = orig.apply(this, arguments)
      e.arguments = arguments
      window.dispatchEvent(e)
      return rv
    }
  }
  history.pushState = _historyWrap('pushState')
  history.replaceState = _historyWrap('replaceState')

  window.addEventListener('pushState', handleNewPage)

  /**
```

```
   * 判断当前网络是否通畅
   * @returns Boolean
      */
    async function judgeNetState() {

    var tryAttemptTimes = 0
    const net_state = await myUtils.onLine()
    if (!net_state) {
      debug && console.log('网络中断，请检查当前网络！已为你结束当前操作！')
      isShowTip &&
        myToast.normal(
          'warning',
          '网络中断，请检查当前网络！已为你结束当前操作！',
          3000,
          'center'
        )
      if (tryAttemptTimes < 2) {
        setTimeout(async () => {
          await judgeNetState()
        }, 1500)
        tryAttemptTimes++
      } else {
        return false
      }
    }
    return true
  }

  /**
   * 监听值的改变的回调函数
   * @param {String} label 名字
   * @param {Function} callback 回调函数
      */
    function GM_onMessage(label, callback) {

    GM_addValueChangeListener(label, async function () {
      await callback.apply(undefined, arguments[2])
    })
  }
  /**
   * 设置值
   * @param {String} label 名字
      */
    function GM_sendMessage(label) {

    GM_setValue(label, Array.from(arguments).slice(1))
  }

  // var lgCollectGifData = {}
  // var lgCollectGifHeader = {}
  if (hijacking_enabled && curPageUrlArr_jobList.length) {
    if (curPageUrlArr_jobList[0].siteType === 1) {
      /**
       * 监听boss的岗位列表，用于发送简历
       */
      GM_onMessage('_jobListJson_boss', async function (src) {
        if (!GM_getValue('allSettings')['autoSendResume_enabled']) {
          isShowTip &&
            myToast.normal('error', '未开启自动投简历功能', 3000, 'center')
          return debug && console.log('未开启自动投简历功能')
        }
        if (isTimesExhausted_bs) {
          debug && console.log(errMsg_bs)
          isShowTip && myToast.normal('error', errMsg_bs, 3000, 'center')
          return false
        }
        var boss = new Boss()

        var webSelector = _config_jobListDomSelectorArr.filter((item) => {
          if (item.siteType === 1) {
            return true
          }
        })
        for (let i = 0; i < webSelector.length; i++) {
          var target = webSelector[i].el
          var a = await myUtils.getElement(document, target, 1500)
          if (a !== null) {
            var firstRes = document.querySelectorAll(target)
            if (!firstRes || firstRes === null) {
              var _targetarr = target.split(' ')

              firstRes = document.getElementsByClassName(
                _targetarr[_targetarr.length - 1].replace('.', '')
              )
            }
            const [hitsCompanyArr] = await autoBlock_getCompanyNameArrFromPage(
              myUtils.uniqueArr(_config_blackListArr),
              myUtils.uniqueArr(_config_whiteListArr),
              document.querySelectorAll(webSelector[i].el),
              2,
              1
            )
            var newJobListJson_Boss = boss.removeBlackList(src, hitsCompanyArr)

            if (GM_getValue('_zp_token') !== null) {
              boss.zp_token = GM_getValue('_zp_token')[0]
              boss._jobListJson_boss = newJobListJson_Boss
              debug && console.log('开始发起沟通!')
              isShowTip &&
                myToast.normal('success', '开始发起沟通', 1500, 'center')
              await boss.main()
            } else {
              debug && console.warn('没有获取到zp_token哦，尝试获取！')
              isShowTip &&
                myToast.normal(
                  'warning',
                  '没有获取到zp_token哦，尝试获取！',
                  1500,
                  'center'
                )
              var zp_token = await boss.get_zp_token()
              zp_token && (await boss.main(zp_token, newJobListJson_Boss))
              !zp_token && debug && console.error('你还未登录！')
              isShowTip &&
                myToast.normal('error', '你还未登录！', 3000, 'center')
            }
          }
        }
      })

      GM_onMessage('_zp_token', async (src) => {})
    } else if (curPageUrlArr_jobList[0].siteType === 2) {
      /**
       * 监听zl的城市id，用于发送简历
       */
      GM_onMessage('_zl_params_cityIds', async (src) => {
        if (!GM_getValue('allSettings')['autoSendResume_enabled']) {
          isShowTip &&
            myToast.normal('error', '未开启自动投简历功能', 1500, 'center')
          return debug && console.log('未开启自动投简历功能')
        }
        debug && console.log('[onMessage]', '_zl_params_cityIds', '=>', src)
        var zl = new ZhiLian(src)
        await zl.main(src)
      })

      GM_onMessage('_zl_params_details', async (src) => {
        debug && console.log('[onMessage]', '_zl_params_details', '=>', src)
      })

      GM_onMessage('_zl_currentJobNumbers', async (src) => {
        debug && console.log('[onMessage]', '_zl_currentJobNumbers', '=>', src)
      })
    } else if (curPageUrlArr_jobList[0].siteType === 3) {
      /**
       * 监听lp的岗位列表，用于发送简历
       */
      GM_onMessage('_lp_params_jobCardList', async function (src) {
        if (!GM_getValue('allSettings')['autoSendResume_enabled']) {
          isShowTip &&
            myToast.normal('error', '未开启自动投简历功能', 1500, 'center')
          return debug && console.log('未开启自动投简历功能')
        }
        if (isTimesExhausted_lp) {
          isShowTip && myToast.normal('error', errMsg_lp, 1500, 'center')
          debug && console.log(errMsg_lp)
          return false
        }
        var _resId
        var lp = new LiePin(src, _resId)
        var curResId = GM_getValue('_lp_params_resId')
        var firstRunFlag = document.getElementById('uselessDiv1')
        if (firstRunFlag == null) {
          debug &&
            console.log('[onMessage]', '_lp_params_jobCardList', '=>', src)
          if (
            curResId == null ||
            !curResId ||
            curResId !== '您的帐号信息已过期，请退出帐号重新登录'
          ) {
            _resId = await lp.get_resId()
            if (_resId !== '您的帐号信息已过期，请退出帐号重新登录') {
              GM_setValue('_lp_params_resId', _resId)
            } else {
              isShowTip && myToast.normal('error', _resId, 1500, 'center')
              return debug && console.error(_resId)
            }
          } else {
            _resId = curResId
          }
          debug && console.log(curResId)

          const uselessDiv1 = document.createElement('div')
          uselessDiv1.id = 'uselessDiv1'
          uselessDiv1.style.display = 'none'
          document.body.append(uselessDiv1)
          setTimeout(async () => {
            lp.resId = _resId
            GM_getValue('_lp_params_jobCardList') !== null && (await lp.main())
            setTimeout(() => {
              var temperUseless = document.getElementById('uselessDiv1')
              if (temperUseless !== null) {
                uselessDiv1.parentNode.removeChild(uselessDiv1)
              }
            }, 5)
          }, 3000)
        } else {
          firstRunFlag.remove()
        }
      })

```

```
    GM_onMessage('_lp_params_resId', async (src) => {
      debug && console.log('[onMessage]', '_lp_params_resId', '=>', src)
    })
  } else if (curPageUrlArr_jobList[0].siteType === 4) {
    GM_onMessage('_lg_params', async (src) => {
      debug && console.log('[onMessage]', '_lg_params', '=>', src)
      if (src !== undefined && src) {
        debug && console.log('----------------------')
        debug && console.log(src)
        debug && console.log(src['X-K-HEADER'])
        debug && console.log(src['X-SS-REQ-HEADER'])
        debug && console.log(src['aesKey'])
        debug && console.log(src['user_trace_token'])
        debug && console.log('-----------------------')
      }
    })
  }
}

/**
 * 第一次运行
 * @param {Array} blackListArr 黑名单列表
 * @param {Array} whiteListArr 白名单列表
    */
  async function firstRun(blackListArr, whiteListArr) {

  if (GM_getValue('allSettings')['autoBlocking_enabled']) {
    var pageType = null
    if (
      curPageUrlArr_jobDetail.length &&
      !curPageUrlArr_companyDetail.length
    ) {
      debug && console.info('检测到当前可能为岗位详情页')
      GM_getValue('allSettings')['showHint_enabled'] &&
        myToast.normal(
          'success',
          '检测到当前可能为岗位详情页',
          1500,
          'top-end'
        )
      pageType = 1
      autoBlock_mainFn(
        true,
        blackListArr,
        whiteListArr,
        pageType,
        _config_detailListDomSelectorArr,
        curPageUrlArr_jobDetail[0].siteType,
        true,
        2000
      )
    } else if (
      (curPageUrlArr_companyDetail.length &&
        !curPageUrlArr_jobDetail.length &&
        !curPageUrlArr_jobList.length) ||
      (curPageUrlArr_companyDetail.length &&
        curPageUrlArr_jobList.length &&
        location.href.indexOf('gongsi') !== -1) ||
      (curPageUrlArr_companyDetail.length &&
        !curPageUrlArr_jobDetail.length &&
        curPageUrlArr_jobList.length)
    ) {
      debug && console.info('检测到当前可能为公司详情页')
      GM_getValue('allSettings')['showHint_enabled'] &&
        myToast.normal(
          'success',
          '检测到当前可能为公司详情页',
          1500,
          'top-end'
        )
      pageType = 1
      autoBlock_mainFn(
        true,
        blackListArr,
        whiteListArr,
        curPageUrlArr_companyDetail.length &&
          curPageUrlArr_jobList.length &&
          document.title.indexOf('全国公司') !== -1
          ? 2
          : pageType,
        _config_detailListDomSelectorArr,
        curPageUrlArr_companyDetail[0].siteType,
        true,
        2000
      )
    } else if (
      curPageUrlArr_jobList.length &&
      !curPageUrlArr_jobDetail.length &&
      !curPageUrlArr_companyDetail.length
    ) {
      debug && console.info('检测到当前页面可能为岗位列表页')
      GM_getValue('allSettings')['showHint_enabled'] &&
        myToast.normal(
          'success',
          '检测到当前页面可能为岗位列表页',
          1500,
          'top-end'
        )
      pageType = 2
      setTimeout(
        () => {
          autoBlock_mainFn(
            true,
            blackListArr,
            whiteListArr,
            pageType,
            _config_jobListDomSelectorArr,
            curPageUrlArr_jobList[0].siteType,
            true,
            2000
          )
        },
        curPageUrlArr_jobList[0].siteType !== 4 ? 1500 : 4500
      )
    } else {
      setTimeout(() => {
        var curPageUrlArr_jobList1 = myUtils.findSthFromObjArr(
          location.href,
          _config_jobListPageUrlArr,
          'url'
        )
        var curPageUrlArr_jobDetail1 = myUtils.findSthFromObjArr(
          location.href,
          _config_jobDetailPageUrlArr,
          'url'
        )
        var curPageUrlArr_companyDetail1 = myUtils.findSthFromObjArr(
          location.href,
          _config_companyDetailPageUrlArr,
          'url'
        )
        if (
          (curPageUrlArr_jobList1.length &&
            curPageUrlArr_jobDetail1.length &&
            curPageUrlArr_companyDetail1.length) ||
          (curPageUrlArr_jobList1.length &&
            !curPageUrlArr_jobDetail1.length &&
            curPageUrlArr_companyDetail1.length) ||
          (curPageUrlArr_jobList1.length &&
            curPageUrlArr_jobDetail1.length &&
            !curPageUrlArr_companyDetail1.length) ||
          (!curPageUrlArr_jobList1.length &&
            curPageUrlArr_jobDetail1.length &&
            curPageUrlArr_companyDetail1.length)
        ) {
          debug && console.info('奇了个怪！快来给我反馈一下！')
          GM_getValue('allSettings')['showHint_enabled'] &&
            myToast.multipleNotControl(
              'error',
              '出错了！',
              1500,
              false,
              '奇了个怪！快来给我反馈一下！',
              () => {}
            )
        } else {
          debug &&
            console.log(
              '当前页面暂不支持！',
              location.origin + location.pathname
            )
        }
      }, 1500)
    }
  }
}

/**
 *
 * @param {Array} blackListArr 黑名单列表
 * @param {Array} whiteListArr 白名单列表
 * @param {Node} webSelector 网页dom选择器 querySelectAll
 * @param {Number} pageType 页面类型 type:1 公司信息页、招聘详情页 type:2 招聘列表
 * @param {Number} siteType 站点类型 1.boss 2.智联 3.猎聘 4.拉勾
 * @returns {Array} [hitsCompanyArr,allCompanyArr,siteType,notHitsCompanyArr]数组 命中的黑名单公司数组、从页面中读取到的公司名字数组，站点类型，未命中的数组
    */
  function autoBlock_getCompanyNameArrFromPage(

  blackListArr,
  whiteListArr,
  webSelector,
  pageType,
  siteType,
  webSelectorMore
) {
  return new Promise((resolve, reject) => {
    try {
      if (
        typeof blackListArr === 'object' &&
        typeof whiteListArr === 'object' &&
        typeof webSelector === 'object' &&
        typeof pageType === 'number' &&
        typeof siteType === 'number' &&
        webSelector.length > 0
      ) {
        var newArr = webSelectorMore
          ? webSelectorMore.map((item) => {
              var wbStor = document.querySelectorAll(item)
              var temp = [].slice.call(wbStor, 0)
              return temp
                ? temp.map(function (el, index) {
                    var txt = el.innerText
                    if (
                      (siteType === 1 &&
                        wbStor[0].parentNode.className === 'user-info') ||
                      (siteType === 2 &&
                        wbStor[0].parentNode.className ===
                          'base-info__title') ||
                      (siteType === 2 &&
                        wbStor[0].parentNode.className === 'industry-mask')
                    ) {
                      txt = txt.split('\n')[0]
                    } else if (
                      siteType === 3 &&
                      wbStor[0].parentNode.className === 'company-info'
                    ) {
                      txt = txt.split(' ')[0]
                    }
                    return [txt, index, wbStor[index]]
                  })
                : []
            })
          : []

        var allCompanyArr = [].slice.call(webSelector, 0)

        var _allCompanyNameArr = allCompanyArr
          ? allCompanyArr.map(function (el, index) {
              var txt = el.innerText
              if (
                siteType === 2 &&
                webSelector[0].parentNode.className === 'base-info__title'
              ) {
                txt = txt.split('\n')[0]
              }
              return [txt, index, webSelector[index]]
            })
          : []
        debug && console.log('_allCompanyNameArr: ', _allCompanyNameArr)
        // [...arr1,...arr2]
        var _newArr = []
        for (let i = 0; i <= newArr.length - 1; i++) {
          _newArr.push(...newArr[i])
        }
        var allCompanyNameArr = [..._newArr, ..._allCompanyNameArr]
        debug && console.log('allCompanyNameArr: ', allCompanyNameArr)

        var hitsCompanyArr = []
        var pattern = null

        const newBlackListArr = blackListArr.filter((blackCompany) => {
          return !whiteListArr.find((whiteCompany) => {
            return (
              whiteCompany.indexOf(blackCompany) !== -1 ||
              whiteCompany.match(
                new RegExp(myUtils.checkName(blackCompany), 'ig')
              ) ||
              whiteCompany === blackCompany
            )
          })
        })

        for (var index = 0; index < allCompanyNameArr.length; index++) {
          var curAllCompanyName = allCompanyNameArr[index][0]
          for (var i = 0; i < newBlackListArr.length; i++) {
            var curNewBlackListName = newBlackListArr[i]
            if (curAllCompanyName === curNewBlackListName) {
              pattern = 1
            } else if (
              curAllCompanyName.indexOf(curNewBlackListName) !== -1 ||
              myUtils
                .checkName(curAllCompanyName)
                .match(
                  new RegExp(myUtils.checkName(curNewBlackListName), 'ig')
                ) ||
              curNewBlackListName.indexOf(curAllCompanyName) !== -1 ||
              myUtils
                .checkName(curNewBlackListName)
                .match(new RegExp(myUtils.checkName(curAllCompanyName), 'ig'))
            ) {
              pattern = 2
            } else {
              pattern = null
            }

            pattern &&
              hitsCompanyArr.unshift({
                bName: curNewBlackListName,
                oName: pattern === 2 ? curAllCompanyName : null,
                documentIndex: allCompanyNameArr[index][1],
                elementNode: allCompanyNameArr[index][2],
                pattern
              })
          }
        }

        const notHitsCompanyArr = allCompanyNameArr.filter((item) => {
          return !hitsCompanyArr.find((blackCompany) => {
            return (
              blackCompany.elementNode === item[2] &&
              blackCompany.documentIndex === item[1]
            )
          })
        })

        debug &&
          console.log(
            '[hitsCompanyArr, allCompanyArr, siteType,notHitsCompanyArr]: ',
            [hitsCompanyArr, allCompanyArr, siteType, notHitsCompanyArr]
          )
        resolve([hitsCompanyArr, allCompanyArr, siteType, notHitsCompanyArr])
      } else {
        GM_getValue('allSettings')['showHint_enabled'] &&
          myToast.normal('warning', '暂不支持当前页面！', 3000)
      }
    } catch (e) {
      if (e.message.indexOf('The provided selector is empty.') !== -1) {
        GM_getValue('allSettings')['showHint_enabled'] &&
          myToast.normal('warning', '暂不支持当前页面！', 3000)
      }
      reject(e.message)
    }
  })
}

/**
 * 处理返回的结果
 * @param {Array} hitsCompanyArr 结果数组
 * @param {Array} allCompanyArr 从页面中读取到的公司名字数组
 * @param {Number} pageType 页面类型 type:1 公司信息页、招聘详情页 type:2 招聘列表
 * @param {Number} siteType 站点类型 1.boss 2.智联 3.猎聘 4.拉勾
    */
  async function autoBlock_handleHitsCompanyArr(

  hitsCompanyArr,
  allCompanyArr,
  pageType,
  siteType
) {
  if (typeof hitsCompanyArr !== 'object' || typeof pageType !== 'number') {
    return false
  }
  hitsCompanyArr = myUtils.uniqueFunc(hitsCompanyArr, 'documentIndex')
  if (hitsCompanyArr && hitsCompanyArr.length !== 0) {
    if (pageType === 1) {
      Swal.fire({
        title: '发现黑名单公司!!',
        type: 'error',
        html: `<span>公司名：<span style="color:red">${
          hitsCompanyArr[0].oName
            ? `${hitsCompanyArr[0].oName}`
            : `${hitsCompanyArr[0].bName}`
        }</span></span><br/>${
          hitsCompanyArr[0].bName
            ? `<span>黑名单中匹配到的公司名：<span style="color:green">${hitsCompanyArr[0].bName}</span></span>`
            : ''
        }<br/><span style="color:#6c757d;text-align: center">------------------------------------------<br/><span style="color:#6c757d">因为该公司在你填写的黑名单公司列表哦！<br/>如果有误判可以移除该公司！</span><br/><------------------------------------------<br/><iframe src="https://kjrate.com?post_type=question&s=${encodeURI(
          hitsCompanyArr[0].bName
        )}" style="width: 100%;height: 15rem"></iframe>`,
        cancelButtonText: '收到',
        confirmButtonText: '确定',
        showCancelButton: true,
        showConfirmButton: true,
        allowOutsideClick: false,
        allowEscapeKey: false,
        allowEnterKey: false
      }).then((result) => {})
    } else if (pageType === 2) {
      var tempString = ''
      console.log('hitsCompanyArr: ', hitsCompanyArr)
      for (const [index, item] of hitsCompanyArr.entries()) {
        tempString += `第${+(index + 1)}家：${
          item.oName ? `${item.oName} => ` : ''
        }${item.bName}\n`
        await autoBlock_switchBlackMode(
          GM_getValue('allSettings')['isShowBlockingCompany'],
          // allCompanyArr[item.documentIndex],
          item.elementNode,
          item.pattern,
          siteType
        )
      }
      GM_getValue('allSettings')['showHint_enabled'] &&
        myToast.normal(
          'info',
          `匹配到 ${
            hitsCompanyArr.length
          } 条黑名单公司！分别是：\n${tempString.substring(
            0,
            tempString.length - 1
          )}`,
          50000
        )
      debug &&
        console.info(
          'info',
          `匹配到 ${
```

```
            hitsCompanyArr.length
          } 条黑名单公司！分别是：\n${tempString.substring(
            0,
            tempString.length - 1
          )}`
        )
    }
  } else {
    GM_getValue('allSettings')['showHint_enabled'] &&
      myToast.normal(
        pageType !== 1 && pageType !== 2 ? 'error' : 'success',
        `${
          pageType === 1
            ? '这家不在黑名单列表哦！'
            : pageType === 2
            ? '当前页面没有黑名单公司！'
            : '奇了怪了！找我反馈一下吧'
        }`,
        3000
      )
    debug &&
      console.log(
        `${
          pageType === 1
            ? '这家不在黑名单列表哦！'
            : pageType === 2
            ? '当前页面没有黑名单公司！'
            : '奇了怪了！找我反馈一下吧'
        }`
      )
  }
  return true
}
/**
 *
 * @param {true} isShow 是否显示
 * @param {Node} target 要操作的dom节点
 * @param {Number} pattern 匹配模式 1.完全匹配 2.模糊匹配
 * @param {Number} siteType 站点类型 1.boss 2.智联 3.猎聘 4.拉勾
    */
  async function autoBlock_switchBlackMode(isShow, target, pattern, siteType) {

  var bgColor = pattern === 1 ? 'red' : pattern === 2 ? 'orange' : ''
  var curl = location.href
  var parentNodeTarget = target.parentNode
  var parentNodeTargetClassName = parentNodeTarget.className
  debug &&
    console.log('target:', target, 'parentNodeTarget: ', parentNodeTarget)
  debug && console.log('curl: ', curl)
  if (!isShow) {
    switch (siteType) {
      case 1:
        if (curl.match('https://www.zhipin.com/xiaoyuan/') !== null) {
          switch (parentNodeTargetClassName) {
            case 'conpany-text':
              parentNodeTarget.parentNode.parentNode.parentNode.parentNode.remove()
              break
            default:
              parentNodeTarget.parentNode.parentNode.remove()
              break
          }
        } else if (
          curl.match('https://www.zhipin.com/returnee_jobs/') !== null
        ) {
          switch (parentNodeTargetClassName) {
            case 'sub-li-bottom-commany-info':
              parentNodeTarget.parentNode.parentNode.parentNode.remove()
```

```
                  break
                case 'conpany-text':
                  parentNodeTarget.parentNode.parentNode.parentNode.remove()
                  break
                default:
                  parentNodeTarget.parentNode.parentNode.parentNode.remove()
                  break
              }
            } else if (curl.match('https://www.zhipin.com/(.*?)') !== null) {
              switch (parentNodeTargetClassName) {
                case 'company-info':
                  parentNodeTarget.parentNode.parentNode.remove()
                  break
                default:
                  parentNodeTarget.parentNode.parentNode.parentNode.parentNode.remove()
                  break
              }
            } else if (curl.indexOf('m.zhipin.com') !== -1) {
              parentNodeTarget.parentNode.parentNode.remove()
            } else {
              parentNodeTarget.parentNode.parentNode.parentNode.remove()
            }
            break
          case 2:
            if (curl.match('https://www.zhaopin.com/.*/') !== null) {
              switch (parentNodeTargetClassName) {
                case `list__item__content`:
                  parentNodeTarget.parentNode.parentNode.remove()
                  break
                case 'industry-mask':
                  parentNodeTarget.parentNode.parentNode.parentNode.remove()
                  break
                case 'list__item__content recommand-list__item__content':
                  parentNodeTarget.parentNode.parentNode.remove()
                  break
                default:
                  if (
                    parentNodeTarget.parentNode.className ===
                    'zp-company-hiring-list__info__list'
                  ) {
                    parentNodeTarget.parentNode.parentNode.parentNode.remove()
                  } else {
                    parentNodeTarget.parentNode.parentNode.parentNode.remove()
                  }
                  break
              }
            } else if (curl.match('https://www.zhaopin.com/') !== null) {
              switch (parentNodeTargetClassName) {
                case 'home-ad-a__company-info':
                case 'home-ad-b__company-info':
                case 'home-ad-c__company-info home-ad-c__company-info--pl60':
                  parentNodeTarget.parentNode.parentNode.parentNode.remove()
                  break
                default:
                  parentNodeTarget.parentNode.parentNode.parentNode.remove()
                  break
              }
            } else if (curl.match('https://xiaoyuan.zhaopin.com/') !== null) {
              switch (parentNodeTargetClassName) {
                case `normal-show-text-table`:
                  parentNodeTarget.parentNode.parentNode.parentNode.remove()
                  break
                default:
                  if (
                    parentNodeTarget.parentNode.parentNode.parentNode.parentNode
                      .className === 'newset-content clearfix'
                  ) {
                    parentNodeTarget.parentNode.parentNode.remove()
                  } else {
                    parentNodeTarget.parentNode.parentNode.parentNode.remove()
                  }
                  break
              }
            } else {
              parentNodeTarget.parentNode.parentNode.parentNode.parentNode.remove()
            }
            break
          case 3:
            if (curl.match('https://www.liepin.com/campus/?t=1') !== null) {
              switch (parentNodeTargetClassName) {
                case 'jsx-3205826082 company-list-item-content':
                  parentNodeTarget.parentNode.parentNode.parentNode.remove()
                  break
                case 'jobcard2-company-info':
                  parentNodeTarget.parentNode.parentNode.parentNode.remove()
                  break
                case 'company-info':
                  parentNodeTarget.parentNode.parentNode.remove()
                  break
                default:
                  parentNodeTarget.parentNode.parentNode.parentNode.remove()
                  break
              }
            } else if (curl.match('https://www.liepin.com') !== null) {
              switch (parentNodeTargetClassName) {
                case 'jobcard2-company-info':
                  parentNodeTarget.parentNode.parentNode.parentNode.remove()
                  break
                case 'home-comp-info-content':
                  parentNodeTarget.parentNode.parentNode.parentNode.remove()
                  break
                default:
                  parentNodeTarget.parentNode.parentNode.parentNode.remove()
                  break
              }
            } else {
              parentNodeTarget.parentNode.parentNode.parentNode.parentNode.remove()
            }
            break
          default:
            debug && console.error('隐藏失败！')
            break
        }
      } else {
        switch (siteType) {
          case 1:
            if (curl.indexOf('m.zhipin.com') !== -1) {
              changeDom(parentNodeTarget.parentNode.parentNode, bgColor, isShow)
            } else if (curl.indexOf('zhipin.com/gongsi/') !== -1) {
              changeDom(parentNodeTarget.parentNode.parentNode, bgColor, isShow)
            } else if (curl.match('https://www.zhipin.com/(.*?)') !== null) {
              if (parentNodeTargetClassName === 'company-info') {
                changeDom(parentNodeTarget.parentNode.parentNode, bgColor, isShow)
              } else {
                changeDom(
                  parentNodeTarget.parentNode.parentNode.parentNode,
                  bgColor,
                  isShow
                )
              }
            } else if (curl.match('https://www.zhipin.com/xiaoyuan/') !== null) {
              if (parentNodeTargetClassName === 'conpany-text') {
                changeDom(
                  parentNodeTarget.parentNode.parentNode.parentNode,
                  bgColor,
                  isShow
                )
              } else if (parentNodeTargetClassName === 'user-info') {
                changeDom(parentNodeTarget.parentNode.parentNode, bgColor, isShow)
              }
            } else if (
              curl.match('https://www.zhipin.com/returnee_jobs/') !== null
            ) {
              if (parentNodeTargetClassName === 'sub-li-bottom-commany-info') {
                changeDom(
                  parentNodeTarget.parentNode.parentNode.parentNode,
                  bgColor,
                  isShow
                )
              } else if (parentNodeTargetClassName === 'conpany-text') {
                changeDom(
                  parentNodeTarget.parentNode.parentNode.parentNode,
                  bgColor,
                  isShow
                )
              } else {
                changeDom(
                  parentNodeTarget.parentNode.parentNode.parentNode,
                  bgColor,
                  isShow
                )
              }
            } else {
              changeDom(
                parentNodeTarget.parentNode.parentNode.parentNode,
                bgColor,
                isShow
              )
              parentNodeTarget.parentNode.parentNode.nextElementSibling.style.background =
                bgColor
            }
            break
          case 2:
            if (curl.match('https://www.zhaopin.com/(.*)/') !== null) {
              switch (parentNodeTargetClassName) {
                case `list__item__content`:
                  changeDom(
                    parentNodeTarget.parentNode.parentNode,
                    bgColor,
                    isShow
                  )
                  break
                case 'industry-mask':
                  break
                case 'list__item__content recommand-list__item__content':
                  changeDom(
                    parentNodeTarget.parentNode.parentNode,
                    bgColor,
                    isShow
                  )
                  break
                default:
                  if (
                    parentNodeTarget.parentNode.className ===
                    'zp-company-hiring-list__info__list'
                  ) {
                    changeDom(
                      parentNodeTarget.parentNode.parentNode.parentNode,
                      bgColor,
                      isShow
                    )
                  }
                  break
              }
            } else if (curl.match('https://www.zhaopin.com/') !== null) {
              switch (parentNodeTargetClassName) {
                case 'home-ad-a__company-info':
                case 'home-ad-b__company-info':
                case 'home-ad-c__company-info home-ad-c__company-info--p160':
                  changeDom(parentNodeTarget.parentNode, bgColor, isShow)
                  break
                default:
                  changeDom(parentNodeTarget.parentNode, bgColor, isShow)
                  break
              }
            } else if (curl.match('https://xiaoyuan.zhaopin.com/') !== null) {
              switch (parentNodeTargetClassName) {
                case `normal-show-text-table`:
                  changeDom(
                    parentNodeTarget.parentNode.parentNode.parentNode,
                    bgColor,
                    isShow
                  )
                  break
                default:
                  if (
                    parentNodeTarget.parentNode.parentNode.parentNode.parentNode
                      .className === 'newset-content clearfix'
                  ) {
                    changeDom(
                      parentNodeTarget.parentNode.parentNode,
                      bgColor,
                      isShow
                    )
                  }
                  break
              }
            } else {
              changeDom(
                parentNodeTarget.parentNode.parentNode.parentNode,
                bgColor,
                isShow
              )
              parentNodeTarget.parentNode.parentNode.nextElementSibling.style.background =
                bgColor
            }
            break
          case 3:
            if (curl.match('https://www.liepin.com/campus/?t=1') !== null) {
              switch (parentNodeTargetClassName) {
                case 'jsx-3205826082 company-list-item-content':
                  changeDom(
                    parentNodeTarget.parentNode.parentNode,
                    bgColor,
                    isShow
                  )
                  break
                case 'jobcard2-company-info':
                  changeDom(
                    parentNodeTarget.parentNode.parentNode,
                    bgColor,
                    isShow
                  )
                  break
                case 'company-info':
                  changeDom(
                    parentNodeTarget.parentNode.parentNode,
                    bgColor,
                    isShow
                  )
                  break
                default:
                  changeDom(
                    parentNodeTarget.parentNode.parentNode,
                    bgColor,
                    isShow
                  )
                  break
              }
            } else if (curl.indexOf('liepin.com/city') !== -1) {
              parentNodeTarget.parentNode.parentNode.parentNode.style.border = `10px solid ${bgColor}`
            } else if (curl.match('https://www.liepin.com') !== null) {
              switch (parentNodeTargetClassName) {
                case 'jobcard2-company-info':
                  changeDom(
                    parentNodeTarget.parentNode.parentNode,
                    bgColor,
                    isShow
                  )
                  break
                case 'home-comp-info-content':
                  changeDom(
                    target.parentNode.parentNode.parentNode,
                    bgColor,
                    isShow
                  )
                  break
                default:
                  changeDom(
                    parentNodeTarget.parentNode.parentNode.parentNode.parentNode,
                    bgColor,
                    isShow
                  )
                  break
              }
            } else {
              parentNodeTarget.parentNode.parentNode.parentNode.parentNode.style.backgroundColor =
                bgColor
            }
            break
          default:
            console.error('操作样式出问题啦！')
            isShowTip &&
              myToast.multipleNotControl(
                'error',
                '出错了！',
                1500,
                false,
                '操作样式出问题啦！',
                () => {}
              )
            break
        }
      }
      function changeDom(
        target,
        bgColor,
        isShow,
        msg = '此家公司在黑名单列表中！'
      ) {
        if (!isShow) {
          target.remove()
        } else {
```

```
      target.style.backgroundColor = bgColor
      target.title = msg
    }
  }
  return 'end'
}

/**
 * 屏蔽功能的主函数
 * @param {Boolean} isFirstRun 是否是第一次运行
 * @param {Array} blackListArr 黑名单列表
 * @param {Array} whiteListArr 白名单列表
 * @param {Number} pageType 页面类型 type:1 公司信息页、招聘详情页 type:2 招聘列表
 * @param {Node} listDomSelectorArr 用于判断网站的dom元素类名
 * @param {Number} delay 延迟获取的毫秒数
    */
  async function autoBlock_mainFn(

  isFirstRun,
  blackListArr,
  whiteListArr,
  pageType,
  listDomSelectorArr,
  curSiteType,
  isHandleHits,
  delay = 1000
) {
  var curNetState = await judgeNetState()
  if (!curNetState) return false
  var res = []
  for (let i = 0; i < listDomSelectorArr.length; i++) {
    if (curSiteType === listDomSelectorArr[i].siteType) {
      const webSelector = listDomSelectorArr[i].el
      const isList = listDomSelectorArr[i].isList
      const b = await myUtils.getElement(document, webSelector, delay)
      debug && console.log('第一条招聘信息的dom元素节点: ', b)
      if (b !== null) {
        const [hitsCompanyArr, allCompanyArr] =
          await autoBlock_getCompanyNameArrFromPage(
            myUtils.uniqueArr(blackListArr),
            myUtils.uniqueArr(whiteListArr),
            document.querySelectorAll(webSelector),
            pageType,
            curSiteType,
            isList || null
          )
        isHandleHits &&
          (await autoBlock_handleHitsCompanyArr(
            hitsCompanyArr,
            allCompanyArr,
            isList === 2 ? isList : pageType,
            curSiteType
          ))
        break
      } else {
        res.push(b)
      }
    }
    if (i === listDomSelectorArr.length - 1 && res[res.length - 1] === null) {
      debug && console.log(res)
      GM_getValue('allSettings')['showHint_enabled'] &&
        myToast.normal('warning', '暂不支持当前页面！', 3000, 'top-end')

      console.log('暂不支持当前页面！')
    }
  }
  if (
    isFirstRun &&
    res.length === listDomSelectorArr.length &&
    res[res.length - 1] === null
  ) {
    GM_getValue('allSettings')['showHint_enabled'] &&
      myToast.normal('warning', '暂不支持当前页面！', 3000, 'top-end')

    console.log('暂不支持当前页面！')
  }
}
class Boss {
  constructor(token, jobJson) {
    this.zp_token = token
    this._jobListJson_boss = jobJson
  }
  /**
   * 发送简历-boss-主函数
   * @param {String} token boss身份验证zp_token
   * @param {Array} jobListJson 岗位列表数组
   * @returns
   */
  async main(token = this.zp_token, jobListJson = this._jobListJson_boss) {
    debugger
    try {
      if (isTimesExhausted_bs) {
        debug && console.log(errMsg_bs)
        GM_getValue('allSettings')['showHint_enabled'] &&
          myToast.normal('error', errMsg_bs, 5000, 'center')
        return false
      }
      var newArr = jobListJson.map((item, index) => {
        return Object.assign(
          {},
          {
            jobId: item.encryptJobId,
            securityId: item.securityId,
            lid: item.lid,
            brandName: item.brandName, // 公司名
            brandIndustry: item.brandIndustry, // 公司所属行业
            brandScaleName: item.brandScaleName, // 公司规模
            brandStageName: item.brandStageName // 公司阶段
          }
        )
      })
      var curNetState = await judgeNetState()
      if (!curNetState) return false
      debug && console.log(`总共要和${newArr.length}家公司发起沟通`)
      var err, rs, templateText
      var errorTimes = 0
      var successTimes = 0
      for (let i = 0; i < newArr.length; i++) {
        var {
          securityId,
          jobId,
          lid,
          brandName,
          brandIndustry,
          brandScaleName,
          brandStageName
        } = newArr[i]
        if (brandIndustry && brandScaleName && brandStageName) {
          templateText = `(${brandIndustry}/${brandScaleName}/${brandStageName})`
        } else if (!brandScaleName && brandStageName) {
          templateText = `(${brandIndustry}/${brandStageName})`
        } else if (brandScaleName && !brandStageName) {
          templateText = `(${brandIndustry}/${brandScaleName})`
        } else {
          templateText = ''
        }
        ;[err, rs] = await myUtils.awaitWrap(
          this.try2Contact(securityId, jobId, lid, token)
        )
        if (rs) {
          debug &&
            console.log(
              '[success]：',
              `第${+(i + 1)}家\n${brandName}${templateText}\n问候语：${rs}`
            )
          // document
          //   .querySelectorAll('.res')[0]
          //   .insertAdjacentHTML(
          //     'beforeend',
          //     `</br><span class="res" style="color:green">[success]：第${+(
          //       i + 1
          //     )}家</br>${brandName}${templateText}</br>问候语：${rs}</span>`
          //   )
          successTimes++
        }
        if (err) {
          debug &&
            console.log(
              '[error]：',
              `第${+(i + 1)}家\n${brandName}${templateText}\n${err[0]}：${
                err[1]
              }`
            )
          // document
          //   .querySelectorAll('.res')[0]
          //   .insertAdjacentHTML(
          //     'beforeend',
          //     `</br><span class="res" style="color:red">[error]：第${+(
          //       i + 1
          //     )}家</br>${brandName}${templateText}</br>${err[0]}：${
          //       err[1]
          //     }</span>`
          //   )
          errorTimes++

          if (err[1] === '今日沟通人数已达上限，请明天再试') {
            errMsg_bs = '今日沟通人数已达上限，请明天再试'
            debug && console.log('已达上限啦，为你退出本次脚本！等明天再来吧')
            isTimesExhausted_bs = true
            break
          }
        }
        await myUtils.randomSleep(3, 6)
      }
      // document
      //   .querySelectorAll('.res')[0]
      //   .insertAdjacentHTML(
      //     'beforeend',
      //     `</br><span class="res">-- 结束 --</span>`
      //   )
      debug &&
        console.log(
          `与当前页面的全部公司结束沟通！\n成功：${successTimes}家；失败：${errorTimes}家`
        )
    } catch (error) {
      debug && console.error(error.message)
    }
  }

  removeBlackList(arr, drr) {
    return arr.filter((item) => {
      var target = item['brandName']
      return !drr.find((element) => {
        var name = element['bName'] ?? element['oName']
        return (
          name.indexOf(target) !== -1 ||
          name.match(new RegExp(target, 'ig')) ||
          name === target
        )
      })
    })
  }

  /**
   * 获取boss的zp_token
   * @returns zp_token
   */
  get_zp_token() {
    return new Promise((resolve, reject) => {
      try {
        axios
          .get('https://www.zhipin.com/wapi/zppassport/get/zpToken')
          .then((result) => {
            var res = result.data
            if (res.zpData) {
              resolve(res.zpData.token)
            } else {
              reject('error?:', res)
            }
          })
      } catch (error) {
        debug && console.log('error: ', error.message)
      }
    })
  }

  /**
   * 发起沟通-boss，因为boss不能直接发简历
   * @param {String} securityId boss身份验证securityId
   * @param {String} jobId jobid
   * @param {String} lid boss身份验证lid
   * @param {String} zp_token boss身份验证zp_token
   * @returns
   */
  try2Contact(securityId, jobId, lid, zp_token = this.zp_token) {
    return new Promise((resolve, reject) => {
      try {
        var config = {
          method: 'post',
          url: `https://www.zhipin.com/wapi/zpgeek/friend/add.json?securityId=${securityId}&jobId=${jobId}&lid=${lid}`,
          headers: {
            zp_token: zp_token,
            'User-Agent':
              userAgents[parseInt(Math.random() * userAgents.length)]
          }
        }
        axios(config)
          .then(function (result) {
            try {
              var res = result.data
              if (res && res.zpData && res.zpData !== {}) {
                if (res.zpData.greeting) {
                  resolve(res.zpData.greeting)
                } else if (res.zpData.showGreeting) {
                  resolve('已发起沟通过！')
                } else if (res.zpData.bizData) {
                  reject([
                    res.zpData.bizData.chatRemindDialog.title,
                    res.zpData.bizData.chatRemindDialog.content
                  ])
                }
              } else {
                reject('api失效了')
              }
            } catch (error) {
              debug && console.log('error: ', error)
            }
          })
          .catch(function (error) {
            debug && console.log(error)
            reject('error: ', error)
          })
      } catch (error) {
        debug && console.log('error: ', error)
      }
    })
  }
}

class ZhiLian {
  constructor(cityId) {
    this.cityId = cityId
  }
  /**
   * 发送简历-zl-提取页面中所有岗位的jobNumbers
   * @param {String} str 岗位招聘信息的链接
   * @returns jobNumbers
   */
  extract_jobNumbers(str) {
    var nurl = new URL(str)
    var reg = /\/([^&#]+)\.htm/
    var query = nurl.pathname.match(reg)
    return query[1] || null
  }
  /**
   * 发送简历-zl-主函数
   * @param {Number} cityId 岗位所在城市的代码，其实也是所在地区的代码
   */
  async main(cityId = this.cityId) {
    var _self = this
    try {
      const allHrefNodeArr = document.querySelectorAll(
        'a.joblist-box__iteminfo'
      )
      var allHrefArr = [].slice.call(allHrefNodeArr, 0)
      const jobNumbersArr = allHrefArr
        ? allHrefArr.map(function (el, index) {
            var txt = _self.extract_jobNumbers(el.href)
            return { jobNumber: txt, index: index }
          })
        : []
      var webSelector = _config_jobListDomSelectorArr.filter((item) => {
        if (item.siteType === 2) {
          return true
        }
      })
      for (let i = 0; i < webSelector.length; i++) {
        var target = webSelector[i].el
        var a = await myUtils.getElement(document, target, 1500)
        if (a !== null) {
          var firstRes = document.querySelectorAll(target)
          if (!firstRes || firstRes === null) {
            var _targetarr = target.split(' ')
            debug && console.log('resume-getElementsByClassName')
```

```
              firstRes = document.getElementsByClassName(
                _targetarr[_targetarr.length - 1].replace('.', '')
              )
            }
            const rr = await autoBlock_getCompanyNameArrFromPage(
              myUtils.uniqueArr(_config_blackListArr),
              myUtils.uniqueArr(_config_whiteListArr),
              firstRes,
              2,
              3
            )
            var _jobNumbersArr = jobNumbersArr.filter((item) => {
              return !rr[0].find((comp) => {
                var target = this.extract_jobNumbers(
                  comp['elementNode'].parentNode.parentNode.parentNode.href
                )
                return (
                  target === item.jobNumber &&
                  comp['documentIndex'] === item.index
                )
              })
            })
            debug && console.log('_jobNumbersArr: ', _jobNumbersArr)
            var jobNumbers = _jobNumbersArr
              ? _jobNumbersArr.map(function (el) {
                  return el.jobNumber
                })
              : []
          }
          var cityIds = Array(jobNumbers.length).fill(cityId)
          var curNetState = await judgeNetState()
          if (!curNetState) return false
          await myUtils.randomSleep(3, 6)
          var obj = {}
          try {
            obj = GM_getValue('_zl_params_details')[0]
            if (JSON.stringify(obj) === '{}' || obj === null) {
              isShowTip &&
                myToast.normal('error', '你还未登录！', 1500, 'center')
              return console.error('你还未登录！')
            }
            const [err, rs] = await myUtils.awaitWrap(
              this.send(
                jobNumbers,
                cityIds,
                obj['_resumeNumber'],
                obj['_at'],
                obj['_rt']
              )
            )
            debug && console.log('rs: ', rs)
            debug && console.log('err: ', err)
            if (rs) {
              debug &&
                console.log(
                  `已向当前页面的全部公司发送完简历！\n共尝试投了${jobNumbers.length}家！`
                )
            }
          } catch (error) {
            debug && console.log('error: ', error, '你还未登录！')
          }
        }
      } catch (error) {
        debug && console.log('error: ', error)
      }
    }

```

```
  /**
   * 发送简历-zl
   * @param {Array} jobNumbers 所有岗位的jobNumbers数组
   * @param {Array} cityIds 岗位所在城市的数组
   * @param {String} resumeNumber 简历id
   * @param {String} at zl的身份验证at
   * @param {String} rt zl的身份验证rt
   * @returns
   */
  send(jobNumbers, cityIds = this.cityId, resumeNumber, at, rt) {
    return new Promise((resolve, reject) => {
      try {
        if (resumeNumber === null || !resumeNumber) {
          isShowTip && myToast.normal('error', '你还未登录! ', 1500, 'center')
          return reject('你还未登录! ')
        }
        debug && console.log('1.5秒后将开始自动投! ')
        setTimeout(() => {
          var config = {
            url: `https://fe-api.zhaopin.com/c/pc/alan/jobs/application`,
            method: 'POST',
            headers: {
              'User-Agent':
                userAgents[parseInt(Math.random() * userAgents.length)]
            },
            data: JSON.stringify({
              jobNumbers,
              cityIds,
              resumeNumber: String(resumeNumber),
              at: String(at),
              rt: String(rt),
              language: 3,
              batched: true,
              inviteCode: '',
              ignoreIntention: 1,
              ignoreBlackType: '',
              deliveryChannelType: 1,
              pageCode: 0,
              jobSource: 'RECOMMENDATION'
            })
          }
          axios(config)
            .then((result) => {
              try {
                var res = result.data
                !res.error ? resolve(res.message) : reject(res.message)
              } catch (e) {
                debug && console.error('[autoSendResume_zl_send]', e.message)
              }
            })
            .catch(function (error) {
              debug && console.log(error)
            })
        }, 1500)
      } catch (error) {
        debug && console.log('error: ', error)
      }
    })
  }
}

class LiePin {
  constructor(originJobList, resId) {
    this.originJobList = originJobList
    this.resId = resId
  }
  /**
```

```
     * 发送简历-lp-提取工作列表（关于公司信息）
     * @param {Array} originJobList 工作列表 jobKind、jobId、jobIndex
     * @param {Boolean} isSaveCompanyInfo  是否要保存公司信息
     * @returns
     */
    extract_JobList(originJobList, isSaveCompanyInfo = false) {
      return originJobList
        ? originJobList.map(function (el, index) {
            var job = el.job
            var comp = el.comp
            var obj = {
              jobIndex: String(+(index % 10)),
              jobId: String(job.jobId),
              jobKind: job.jobKind !== 2 ? String(2) : String(job.jobKind),
              info: `第${+(index + 1)}家公司\n${job.title}【${job.dq}】${
                job.salary
              }\n${comp.compName} | ${comp.compIndustry} ${
                comp.compStage ? comp.compStage : ''
              } ${comp.compScale}`
            }
            if (!isSaveCompanyInfo) {
              delete obj.info
            }
            return obj
          })
        : []
    }

    /**
     * 发送简历-lp-主函数
     * @param {Array} originJobList 工作列表 jobKind、jobId、jobIndex
     * @param {String} resId 简历id
     * @returns
     */
    async main(originJobList = this.originJobList, resId = this.resId) {
      if (isTimesExhausted_lp) {
        debug && console.log(errMsg_lp)
        GM_getValue('allSettings')['showHint_enabled'] &&
          myToast.normal('error', errMsg_lp, 5000, 'center')
        return false
      }
      if (!resId || resId === undefined || resId === null) {
        debug && console.log('!!!')
        resId = await this.get_resId()
        if (resId === '您的帐号信息已过期，请退出帐号重新登录') {
          isShowTip && myToast.normal('error', '请重新登录！', 1500, 'center')
          return debug && console.error('请重新登录！')
        }
      }
      var webSelector = _config_jobListDomSelectorArr.filter((item) => {
        if (item.siteType === 3) {
          return true
        }
      })
      for (let i = 0; i < webSelector.length; i++) {
        var target = webSelector[i].el
        var a = await myUtils.getElement(document, target, 1500)
        if (a !== null) {
          var firstRes = document.querySelectorAll(target)
          if (!firstRes || firstRes === null) {
            var _targetarr = target.split(' ')
            debug && console.log('resume-getElementsByClassName')
            firstRes = document.getElementsByClassName(
              _targetarr[_targetarr.length - 1].replace('.', '')
            )
          }
          const rr = await autoBlock_getCompanyNameArrFromPage(
            myUtils.uniqueArr(_config_blackListArr),
            myUtils.uniqueArr(_config_whiteListArr),
            firstRes,
            2,
            3
          )
          var _originJobList = originJobList.filter((item) => {
            return !rr[0].find((comp) => {
              return (
                comp['oName'] === item.comp.compName ||
                comp['bName'] === item.comp.compName ||
                comp['bName'].indexOf(item.comp.compName) !== -1 ||
                comp['bName'].match(new RegExp(item.comp.compName, 'ig')) ||
                comp['bName'] === item.comp.compName
              )
            })
          })
          var newJobList = myUtils.sliceArrLikeBeEQLengthGroup(
            this.extract_JobList(_originJobList, false),
            10
          )

          var _newJobList = myUtils.sliceArrLikeBeEQLengthGroup(
            this.extract_JobList(_originJobList, true),
            10
          )

          var curNetState = await judgeNetState()
          if (!curNetState) return false
          var err, rs
          var errorTimes = 0
          var successTimes = 0

          for (let i = 0; i < newJobList.length; i++) {
            ;[err, rs] = await myUtils.awaitWrap(
              this.send(resId, newJobList[i])
            )
            if (rs) {
              successTimes++
              debug && console.log(rs)
              var templateText = ''
              for (var j = 0; j < _newJobList[i].length; j++) {
                templateText += `\n` + _newJobList[i][j].info
              }
              debug && console.log('templateText: ', templateText)
            }
            if (err) {
              errorTimes++
              if (err.code === '20010') {
                errMsg_lp = err.msg
                debug && console.info(errMsg_lp)
                isTimesExhausted_lp = true
                break
              } else if (err.code === '-1401') {
                GM_getValue('allSettings')['showHint_enabled'] &&
                  myToast.normal('warning', err.msg, 3000, 'center')
                debug && console.log(err.msg)
                break
              }
            }
            await myUtils.randomSleep(3, 6)
          }
          debug &&
            console.log(
              `已向当前页面的全部公司发送完简历！\n成功：${successTimes}家；失败：${errorTimes}家`
            )
        }
```

```
      }
    }

    /**
     *  发送简历-lp
     * @param {String} resId 简历id
     * @param {Array} jobInfo 工作的jobKind、jobId、jobIndex
     * @returns code || msg
     */
    send(resId = this.resId, jobInfo) {
      return new Promise((resolve, reject) => {
        if (jobInfo) return resolve('not send:', jobInfo)
        var myHeaders = new Headers()
        myHeaders.append('Accept', 'application/json, text/plain, */*')
        myHeaders.append('Accept-Language', 'zh-CN,zh;q=0.9')
        myHeaders.append('Connection', 'keep-alive')
        myHeaders.append('Content-Type', 'application/x-www-form-urlencoded')
        myHeaders.append('Origin', 'https://c.liepin.com')
        myHeaders.append('Referer', 'https://c.liepin.com/')
        myHeaders.append('Sec-Fetch-Dest', 'empty')
        myHeaders.append('Sec-Fetch-Mode', 'cors')
        myHeaders.append('Sec-Fetch-Site', 'same-site')
        myHeaders.append(
          'User-Agent',
          userAgents[parseInt(Math.random() * userAgents.length)]
        )
        myHeaders.append('X-Client-Type', 'web')
        myHeaders.append('X-Fscp-Fe-Version', '6f6ae4a')
        myHeaders.append('X-Fscp-Std-Info', '{"client_id": "40106"}')
        myHeaders.append(
          'X-Fscp-Trace-Id',
          '9751cb68-507c-4580-ad2c-b3a6c295aca3'
        )
        myHeaders.append('X-Fscp-Version', '1.1')
        myHeaders.append('X-Requested-With', 'XMLHttpRequest')
        var urlencoded = new URLSearchParams()
        urlencoded.append('resId', resId)
        urlencoded.append('jobInfo', JSON.stringify(jobInfo))

        var requestOptions = {
          method: 'POST',
          headers: myHeaders,
          body: urlencoded,
          redirect: 'follow',
          credentials: 'include'
        }

        fetch(
          'https://apic.liepin.com/api/com.liepin.capply.platform.apply.batch-apply',
          requestOptions
        )
          .then((response) => response.json())
          .then((result) => {
            if (
              result.flag === 0 &&
              (result.code === '20010' || result.code === '-1401')
            ) {
              reject({
                code: result.code,
                msg: result.msg
              })
            } else resolve(result)
          })
          .catch((error) => reject('error', error))
      })
    }

    /**
     * 获取lp的resId
     * @returns resId || message
     */
    get_resId() {
      return new Promise((resolve, reject) => {
        var myHeaders = new Headers()
        myHeaders.append('Accept', 'application/json, text/plain, */*')
        myHeaders.append(
          'Accept-Language',
          'zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6'
        )
        myHeaders.append('Connection', 'keep-alive')
        myHeaders.append('Content-Length', '0')
        myHeaders.append('Content-Type', 'application/x-www-form-urlencoded')
        myHeaders.append('DNT', '1')
        myHeaders.append('Origin', 'https://www.liepin.com')
        myHeaders.append('Referer', 'https://www.liepin.com/')
        myHeaders.append('Sec-Fetch-Dest', 'empty')
        myHeaders.append('Sec-Fetch-Mode', 'cors')
        myHeaders.append('Sec-Fetch-Site', 'same-site')
        myHeaders.append(
          'User-Agent',
          'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5060.134 Safari/537.36 Edg/103.0.1264.77'
        )
        myHeaders.append('X-Client-Type', 'web')
        myHeaders.append(
          'X-Fscp-Bi-Stat',
          '{"location": "https://www.liepin.com/zhaopin/?city=410&dq=410&pubTime=&currentPage=0&pageSize=40&key=%E5%89%8D%E7%AB%AF&workYearCode=0&compId=&compName=&compTag=&industry=&salary=&jobKind=&compScale=&compKind=&compStage=&eduLevel=&otherCity=&scene=input&suggestId="}'
        )
        myHeaders.append('X-Fscp-Fe-Version', '2e3a0e1')
        myHeaders.append('X-Fscp-Std-Info', '{"client_id": "40108"}')
        myHeaders.append(
          'X-Fscp-Trace-Id',
          'fb6d6c4e-8618-4ae1-a799-b81eae2e09c6'
        )
        myHeaders.append('X-Fscp-Version', '1.1')
        myHeaders.append('X-Requested-With', 'XMLHttpRequest')

        var requestOptions = {
          method: 'POST',
          headers: myHeaders,
          redirect: 'follow',
          credentials: 'include'
        }

        fetch(
          'https://apic.liepin.com/api/com.liepin.usercx.pc.user.base-property',
          requestOptions
        )
          .then((response) => response.json())
          .then((result) => {
            return resolve(result.data ? result.data.resId : result.msg)
          })
          .catch((error) => reject('error', error))
      })
    }
  }

  await firstRun(_config_blackListArr, _config_whiteListArr)
  // #endregion

  function getUserAgentObj() {
    let browserReg = {
      Chrome: /Chrome/,
      IE: /MSIE/,
      Firefox: /Firefox/,
      Opera: /Presto/,
      Safari: /Version\/([\d.]+).*Safari/,
      360: /360SE/,
      QQBrowswe: /QQ/
    }

    let deviceReg = {
      iPhone: /iPhone/,
      iPad: /iPad/,
      Android: /Android/,
      Windows: /Windows/,
      Mac: /Macintosh/
    }
    let userAgentStr = navigator.userAgent
    const userAgentObj = {
      browserName: '', // 浏览器名称
      browserVersion: '', // 浏览器版本
      osName: '', // 操作系统名称
      osVersion: '', // 操作系统版本
      deviceName: '' // 设备名称
    }
    console.log(userAgentStr)

    for (let key in browserReg) {
      if (browserReg[key].test(userAgentStr)) {
        userAgentObj.browserName = key
        if (key === 'Chrome') {
          userAgentObj.browserVersion = userAgentStr
            .split('Chrome/')[1]
            .split(' ')[0]
        } else if (key === 'IE') {
          userAgentObj.browserVersion = userAgentStr
            .split('MSIE ')[1]
            .split(' ')[1]
        } else if (key === 'Firefox') {
          userAgentObj.browserVersion = userAgentStr.split('Firefox/')[1]
        } else if (key === 'Opera') {
          userAgentObj.browserVersion = userAgentStr.split('Version/')[1]
        } else if (key === 'Safari') {
          userAgentObj.browserVersion = userAgentStr
            .split('Version/')[1]
            .split(' ')[0]
        } else if (key === '360') {
          userAgentObj.browserVersion = ''
        } else if (key === 'QQBrowswe') {
          userAgentObj.browserVersion = userAgentStr
            .split('Version/')[1]
            .split(' ')[0]
        }
      }
    }

    for (let key in deviceReg) {
      if (deviceReg[key].test(userAgentStr)) {
        userAgentObj.osName = key
        if (key === 'Windows') {
          userAgentObj.osVersion = userAgentStr
            .split('Windows NT ')[1]
            .split(';')[0]
        } else if (key === 'Mac') {
          userAgentObj.osVersion = userAgentStr
            .split('Mac OS X ')[1]
            .split(')')[0]
        } else if (key === 'iPhone') {
          userAgentObj.osVersion = userAgentStr
            .split('iPhone OS ')[1]
            .split(' ')[0]
        } else if (key === 'iPad') {
          userAgentObj.osVersion = userAgentStr
            .split('iPad; CPU OS ')[1]
            .split(' ')[0]
        } else if (key === 'Android') {
          userAgentObj.osVersion = userAgentStr
            .split('Android ')[1]
            .split(';')[0]
          userAgentObj.deviceName = userAgentStr
            .split('(Linux; Android ')[1]
            .split('; ')[1]
            .split(' Build')[0]
        }
      }
    }

    return userAgentObj
  }

  function handleNewPageElements(pageType, siteType, delayMS) {
    const uselessDiv = document.createElement('div')
    uselessDiv.id = 'uselessDiv'
    uselessDiv.style.display = 'none'
    document.body.append(uselessDiv)
    setTimeout(() => {
      autoBlock_mainFn(
        false,
        myUtils.uniqueArr(_config_blackListArr),
        myUtils.uniqueArr(_config_whiteListArr),
        pageType,
        _config_jobListDomSelectorArr,
        siteType,
        true,
        delayMS
      )
      setTimeout(() => {
        var sss = document.getElementById('uselessDiv')
        if (sss !== null) {
          document.body.removeChild(uselessDiv)
        }
      }, 5)
    }, 3000)
  }

  // 定义一个函数来处理新元素
  function handleNewPage() {
    // newElements.forEach((element) => {
    //   // 在这里处理每个新元素
    //   // 例如，打印新元素的内容
    //   console.log('New element:', element.textContent)
    // })
    var firstRunFlag = document.getElementById('uselessDiv')
    if (firstRunFlag === null) {
      debug && console.info('检测到翻页！')
      isShowTip && myToast.normal('success', '检测到翻页！', 1000, 'top-end')
      if (GM_getValue('allSettings')['autoBlocking_enabled']) {
        var curPageUrlArr_jobList = myUtils.findSthFromObjArr(
          location.href,
          _config_jobListPageUrlArr,
          'url'
        )
        var curPageUrlArr_jobDetail = myUtils.findSthFromObjArr(
          location.href,
          _config_jobDetailPageUrlArr,
          'url'
        )
        var curPageUrlArr_companyDetail = myUtils.findSthFromObjArr(
          location.href,
          _config_companyDetailPageUrlArr,
          'url'
        )
        var pageType = null
        if (
          curPageUrlArr_jobDetail.length &&
          !curPageUrlArr_jobList.length &&
          !curPageUrlArr_companyDetail.length
        ) {
          isShowTip &&
            myToast.normal(
              'success',
              '检测到当前可能为岗位详情页!',
              1000,
              'top-end'
            )
          return debug && console.info('检测到当前可能为岗位详情页')
        } else if (
          curPageUrlArr_companyDetail.length &&
          !curPageUrlArr_jobDetail.length &&
          !curPageUrlArr_jobList.length
        ) {
          isShowTip &&
            myToast.normal(
              'success',
              '检测到当前可能为公司详情页!',
              1000,
              'top-end'
            )
          return debug && console.info('检测到当前可能为公司详情页')
        } else if (
          curPageUrlArr_jobList.length &&
          !curPageUrlArr_jobDetail.length &&
          !curPageUrlArr_companyDetail.length
        ) {
          isShowTip &&
            myToast.normal(
              'success',
              '检测到当前页面可能为岗位列表页!',
              1000,
              'top-end'
            )
          debug && console.info('检测到当前页面可能为岗位列表页')
          pageType = 2
          handleNewPageElements(
            pageType,
            curPageUrlArr_jobList[0].siteType,
            1500
          )
        } else {
          setTimeout(() => {
            var curPageUrlArr_jobList1 = myUtils.findSthFromObjArr(
              location.href,
              _config_jobListPageUrlArr,
              'url'
            )
            var curPageUrlArr_jobDetail1 = myUtils.findSthFromObjArr(
              location.href,
              _config_jobDetailPageUrlArr,
              'url'
            )
            var curPageUrlArr_companyDetail1 = myUtils.findSthFromObjArr(
              location.href,
              _config_companyDetailPageUrlArr,
              'url'
            )
            console.log(
              curPageUrlArr_jobList1,
              curPageUrlArr_jobDetail1,
              curPageUrlArr_companyDetail1
            )
            if (
              curPageUrlArr_jobList1.length &&
              curPageUrlArr_jobDetail1.length &&
              curPageUrlArr_companyDetail1.length
            ) {
              debug && console.info('奇了个怪！快来给我反馈一下！')
              isShowTip &&
                myToast.multipleNotControl(
                  'error',
                  '出错了！',
                  1500,
                  false,
                  '奇了个怪！快来给我反馈一下！',
                  () => {}
                )
            } else {
              debug &&
                console.log(
                  '当前页面暂不支持！',
                  location.origin + location.pathname
                )
              isShowTip &&
                myToast.normal(
                  'info',
                  `当前页面暂不支持！\n当前网页链接：${location.origin}${location.pathname}`,
                  1000,
                  'top-end'
                )
            }
          }, 1500)
        }
      }
      if (GM_getValue('allSettings')['autoSendResume_enabled']) {
        console.log('success', '你已开启自动投简历！', 3000, 'center')
        autoSend()
      }
    } else {
      firstRunFlag.remove()
    }
  }
})()
```