

```

1 // ==UserScript==
2 // @name      自动屏蔽黑名单公司
3 // @namespace  https://freysu.github.io/
4 // @version   1.0
5 // @description F**k BlackList Company! Auto Blocking Company and Auto Send Resume!
6 // @author    Freysu
7 // @match     *://*.zhipin.com/*
8 // @match     *://*.zhaopin.com/*
9 // @match     *://*.liepin.com/*
10 // @match     *://*.lagou.com/*
11 // @run-at    document-idle
12 // @license   Apache Licence 2.0
13 // @require   https://cdn.staticfile.org/limonte-sweetalert2/8.19.0/sweetalert2.all.min.js
14 // @require   https://unpkg.com/axios/dist/axios.min.js
15 // @require   file:///D:/CodeStudy/fuckBlacklistCompany/config_CompanyBlackList.js
16 // @grant     GM_addStyle
17 // @grant     GM_xmlHttpRequest
18 // @grant     GM_setValue
19 // @grant     GM_getValue
20 // @grant     GM_listValues
21 // @grant     GM_registerMenuCommand
22 // @grant     GM_unregisterMenuCommand
23 // @grant     GM_addValueChangeListener
24 // @grant     unsafeWindow
25 // @connect   https://sa.lagou.com/collect.gif
26 // ==/UserScript==
27
28 /* eslint-disable no-inner-declarations */
29 /* eslint-disable no-undef */
30 /* eslint-disable no-lone-blocks */
31 /* eslint-disable new-cap */
32 /* eslint-disable no-extra-semi */
33 /* eslint-disable space-before-function-paren */
34 /* eslint-disable indent */
35 ;(async function () {
36     'use strict'
37     // 导入外部CSS样式
38     const style = document.createElement('link')
39     style.rel = 'stylesheet'
40     style.href =
41         'https://kjrate.com/wp-content/plugins/anspress-question-answer/templates/css/main.css?ver=4.3.2'
42     document.head.appendChild(style)
43     // -----
44     // 配置区
45     //
46     // #region
47     // -----
48     // prettier-ignore
49     const _config_kgCompanyBlackList = Config_kgCompanyBlackList
50     // 黑名单数组
51     const _config_blackListArr_1 = Config_blackListArr_1
52     // 白名单数组
53     // prettier-ignore
54     const _config_whiteListArr = []
55     // #endregion
56
57     /**
58      * 勿动
59      */
60     const _config_jobListDomSelectorArr = [
61
62         {
63             el: '#main > div.job-recommend > div.job-list.job-list-new > ul > li > a > div.name >
span.company',

```

```

64     siteType: 1
65 },
66 {
67     el: '.job-card-wrapper .job-card-right .company-name',
68     siteType: 1
69 },
70 {
71     el: ".job-list-new .item .name span[class='company']",
72     siteType: 1
73 },
74 {
75     el: '.cur.sub-ul-0.sub-ul-box > li > .sub-li > .sub-li-bottom > .user-info > p >
.name',
76     siteType: 1,
77     isList: [
78         '.hot-job-box.merge-city-job.common-tab-box > .cur > li > .sub-li > .sub-li-bottom >
.user-info > p > .name',
79         '.hot-company-wrapper ul li .company-info-top h3'
80     ]
81 },
82 {
83     el: '.school-body-wrapper .school-page.company-tab-box li .sub-li .company-title
.company-text h4',
84     siteType: 1,
85     isList: ['.cur > li > .sub-li > .user-info > p']
86 },
87 {
88     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-commpany-info > .name',
89     siteType: 1,
90     isList: [
91         '.practice-pos-buried.practice-pos.merge-city-job.common-tab-box > .position-list >
.cur > li > .sub-li > .sub-li-bottom > .sub-li-bottom-commpany-info > .name',
92         '.company-text > h4'
93     ]
94 },
95 {
96     el: '.positionlist .iteminfo__line1__compname__name',
97     siteType: 2
98 },
99 {
100     el: '.home-ad-a__card > .home-ad-a__company-info > .home-ad-a__company-info__name`,
101     siteType: 2,
102     isList: [
103         '.home-ad-b__card > .home-ad-b__company-info > .home-ad-b__company-info__name`,
104         '.home-ad-c__card > .home-ad-c__company-info--pl60.home-ad-c__company-info > .home-
ad-c__company-info__title`
105     ]
106 },
107 {
108     el: '.job-container__list__item > .list > div.list__item > a > .list__item__content >
.list__item__content--company',
109     siteType: 2,
110     isList: [
111         '.industry-mask__title',
112         '.zp-company-hiring-list__info__list--name`,
113         '.recommand-list__item__content--company.list__item__content--company`
114     ]
115 },
116 {
117     el: '.show-text-table-cell',
118     siteType: 2,
119     isList: [
120         '.industry-mask__title',
121         '.newest .newset-content ul li div:first-child .des-color`
122     ]
123 },

```

```

124     {
125         el: '.job-card-pc-container .company-name',
126         siteType: 3
127     },
128     {
129         el: '.seo-home-comp-card .home-comp-info .home-comp-info-content .comp-info-title',
130         siteType: 3
131     },
132     {
133         el: '.seo-home-comp-card > .home-comp-info > .home-comp-info-content > .ellipsis-
134         1.comp-info-title',
135         siteType: 3,
136         isList: [
137             '.seo-jobcard2-common .jobcard2-company-box .jobcard2-company-info .jobcard2-company-
138             title'
139         ]
140     },
141     {
142         el: '.company-list-item-content.jsx-3205826082 .title.jsx-3205826082',
143         siteType: 3,
144         isList: [
145             '.seo-jobcard2-common .jobcard2-company-box .jobcard2-company-info .jobcard2-company-
146             title',
147             '.campus-school-jobs-content .campus-school-jobs-item .school-jobs-company .company-
148             info .company-title'
149         ]
150     }
151 ]
152
153 const _config_detailListDomSelectorArr = [
154     {
155         el: '#__layout > div > div.company-info > div > div.name',
156         siteType: 1
157     },
158     {
159         el: '#main > div.job-detail > a > div.info-primary > div > div',
160         siteType: 1
161     },
162     {
163         el: '.sider-company .company-info',
164         siteType: 1
165     },
166     {
167         el: '.company-banner .name',
168         siteType: 1
169     },
170     {
171         el: '.page-gongsi .home-site .company-tab-box li .sub-li .company-text h4',
172         siteType: 1
173     },
174     {
175         el: '.mian-company .base-info__title h1',
176         siteType: 2
177     },
178     {
179         el: '.publisher .company__title',
180         siteType: 2
181     },
182     {
183         el: '.company-header-content-name .name-right .title-box .title',
184         siteType: 3
185     },
186     {
187         el: '.company-info-container .company-card .content .name',
188         siteType: 3
189     }
190 ]

```

```
187 const _config_jobListPageUrlArr = [  
188   {  
189     url: 'm.zhipin.com/',  
190     siteType: 1  
191   },  
192   {  
193     url: 'www.zhipin.com/web/geek/job',  
194     siteType: 1  
195   },  
196   {  
197     url: 'www.zhipin.com/',  
198     siteType: 1  
199   },  
200   {  
201     url: 'sou.zhaopin.com/',  
202     siteType: 2  
203   },  
204   {  
205     url: 'www.zhaopin.com/',  
206     siteType: 2  
207   },  
208   {  
209     url: 'xiaoyuan.zhaopin.com/',  
210     siteType: 2  
211   },  
212   {  
213     url: 'www.liepin.com/zhaopin/',  
214     siteType: 3  
215   },  
216   {  
217     url: 'www.liepin.com/',  
218     siteType: 3  
219   }  
220 ]  
221  
222 const _config_jobDetailPageUrlArr = [  
223   {  
224     url: 'www.zhipin.com/job_detail/',  
225     siteType: 1  
226   },  
227   {  
228     url: 'm.zhipin.com/job_detail/',  
229     siteType: 1  
230   },  
231   {  
232     url: 'jobs.zhaopin.com/',  
233     siteType: 2  
234   },  
235   {  
236     url: 'www.liepin.com/job/',  
237     siteType: 3  
238   },  
239   {  
240     url: 'www.liepin.com/lptjob/',  
241     siteType: 3  
242   }  
243 ]  
244  
245 const _config_companyDetailPageUrlArr = [  
246   {  
247     url: 'm.zhipin.com/gongsi/',  
248     siteType: 1  
249   },  
250   {  
251     url: 'm.zhipin.com/gongsir/',  
252     siteType: 1  
253   },
```

```
254     {
255         url: 'www.zhipin.com/gongsi/',
256         siteType: 1
257     },
258     {
259         url: 'www.zhipin.com/gongsir/',
260         siteType: 1
261     },
262     {
263         url: 'www.zhaopin.com/companydetail/',
264         siteType: 2
265     },
266     {
267         url: 'www.liepin.com/company/',
268         siteType: 3
269     }
270 ]
271 var isTimesExhausted_bs = false
272 var errMsg_bs = ''
273 var isTimesExhausted_lp = false
274 var errMsg_lp = ''
275 var hijacking_enabled = true
276
277 // prettier-ignore
```

[illegible]

[illegible]



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","MQQBrowsers/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 Nokia97-1/20.0.019; 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"];

```

```

279
280     const debug = true
281
282     /**
283      * 通知
284      */
285     const myToast = {
286
287         /**
288          * @description toast通知
289          * @param {string} type "success", "error", "warning", "info" or "question"
290          * @param {string} title 标题
291          * @param {number} timer 显示时间
292          * @param {string} html html类型内容, 选填
293          * @param {string} position 显示位置
294          */
295         normal: function (type, title, timer, position = 'center', html = null) {
296             const Toast = Swal.mixin({
297                 toast: true,
298                 position: position,
299                 showConfirmButton: false,
300                 showCloseButton: true

```

```

301     })
302     Toast.fire({
303         type,
304         title,
305         html,
306         timer
307     })
308 },
309 /**
310  * @description 不控制的常规弹窗
311  * @param {string} type "success","error","warning","info" or "question"
312  * @param {string} title 标题
313  * @param {number} timer 显示时间
314  * @param {boolean} isHtml text or html
315  * @param {string} content 内容
316  */
317 multipleNotControl: function (
318     type = null,
319     title,
320     timer,
321     isHtml,
322     content,
323     confirmCallback
324 ) {
325     var text, html
326     if (isHtml) {
327         text = null
328         html = content
329     } else {
330         text = content
331         html = null
332     }
333     Swal.fire({
334         type,
335         title,
336         text,
337         timer,
338         html,
339         confirmButtonText: '我知道了'
340     }).then((result) => {
341         confirmCallback(result)
342     })
343 },
344 multipleControl: function (
345     type = null,
346     title,
347     timer,
348     isHtml,
349     content,
350     confirmCallback
351 ) {
352     var text, html
353     if (isHtml) {
354         text = null
355         html = content
356     } else {
357         text = content
358         html = null
359     }
360     Swal.fire({
361         type,
362         title,
363         text,
364         timer,
365         html,
366         confirmButtonText: '我知道了',
367         allowOutsideClick: false,

```

```

368     allowEscapeKey: false,
369     allowEnterKey: false
370   }).then((result) => {
371     confirmCallback(result)
372   })
373 }
374 }
375
376 /**
377  * 工具库
378  */
379 const myUtils = {
380
381   /**
382    *
383    * @param {object} parent 被监听的元素的父节点，也可以是document
384    * @param {Node} selector 被监听的dom元素
385    * @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
386    * @returns
387    */
388   getElement: function (parent, selector, timeout = 0) {
389     return new Promise((resolve) => {
390       let result = parent.querySelector(selector)
391       if (result) return resolve(result)
392       let timer
393       const mutationObserver =
394         window.MutationObserver ||
395         window.WebkitMutationObserver ||
396         window.MozMutationObserver
397       if (mutationObserver) {
398         const observer = new mutationObserver((mutations) => {
399           for (const mutation of mutations) {
400             for (const addedNode of mutation.addedNodes) {
401               if (addedNode instanceof Element) {
402                 result = addedNode.matches(selector)
403                   ? addedNode
404                   : addedNode.querySelector(selector)
405                 if (result) {
406                   observer.disconnect()
407                   timer && clearTimeout(timer)
408                   return resolve(result)
409                 }
410             }
411           }
412         })
413       }
414       observer.observe(parent, {
415         childList: true,
416         subtree: true
417       })
418       if (timeout > 0) {
419         timer = setTimeout(() => {
420           observer.disconnect()
421           return resolve(null)
422         }, timeout)
423       }
424     } else {
425       const listener = (e) => {
426         if (e.target instanceof Element) {
427           result = e.target.matches(selector)
428             ? e.target
429             : e.target.querySelector(selector)
430           if (result) {
431             parent.removeEventListener('DOMNodeInserted', listener, true)
432             timer && clearTimeout(timer)
433             return resolve(result)
434           }
435         }
436       }

```

```

435     }
436   }
437   parent.addEventListener('DOMNodeInserted', listener, true)
438   if (timeout > 0) {
439     timer = setTimeout(() => {
440       parent.removeEventListener('DOMNodeInserted', listener, true)
441       return resolve(null)
442     }, timeout)
443   }
444 }
445 })
446 },
447 uniqueFunc: (arr, uniId) => {
448   // const res = new Map()
449   // return arr.filter(
450   //   (item) => !res.has(item[uniId]) && res.set(item[uniId], 1)
451   // )
452   if (!arr.length) return arr
453   const repeatArr = new Set()
454   if (uniId) {
455     return arr.reduce((pre, cur) => {
456       let item = null
457       if (typeof uniId === 'string' && uniId) {
458         item = cur[uniId]
459       } else {
460         item = typeof cur === 'object' ? JSON.stringify(cur) : cur
461       }
462       if (!repeatArr.has(item)) {
463         repeatArr.add(item)
464         pre.push(cur)
465       }
466       return pre
467     }, [])
468   }
469 },
470 uniqueArr: (arr) => Array.from(new Set(arr)),
471 sleep: function (duration) {
472   return new Promise((resolve) => {
473     setTimeout(resolve, duration)
474   })
475 },
476 /**
477  *
478  * @param {Array} toArr 要分割的数组
479  * @param {Number} eqLen 等长值
480  * @returns newArr
481  */
482 sliceArrLikeBeEqLengthGroup: (toArr, eqLen) => {
483   let index = 0
484   const newArr = []
485   while (index < toArr.length) {
486     newArr.push(toArr.slice(index, (index += eqLen)))
487   }
488   return newArr
489 },
490 /**
491  * async/await 的错误处理方法 1. try...catch 2. async/await 本质就是 promise 的语法糖
492  * @param {promise} promise 要执行的promise
493  * @returns {array} [err,rs] err:错误码, data:结果
494  */
495 awaitwrap: function (promise) {
496   return promise.then((data) => [null, data]).catch((err) => [err, null])
497 },
498 debounce: function (fn, delay) {
499   let timeout = null
500   return function () {
501     const args = arguments

```

```

502         if (timeout) window.clearTimeout(timeout)
503         timeout = setTimeout(() => {
504             fn.apply(this, args)
505         }, delay)
506     }
507 },
508 // 通过判断图片是否加载成功来判断网络通断
509 onLine: function () {
510     // 基于promise, 可以用async和await
511     var img = new Image()
512     img.src = 'https://www.baidu.com/favicon.ico?t=' + Date.now()
513     return new Promise((resolve, reject) => {
514         img.onload = function () {
515             resolve(true)
516         }
517         img.onerror = function () {
518             resolve(false)
519         }
520     })
521 },
522 /**
523  * 从数组中查找出模糊或完全符合的内容
524  * @param sth 要查找的内容
525  * @param arr 被查找的数组
526  * @param name 被查找的属性名
527  * @returns Array
528  */
529 findSthFromObjArr: (sth, arr, name) => {
530     return (
531         arr.filter((item) => {
532             return sth.indexOf(item[name]) !== -1 || sth === item[name]
533         }) || []
534     )
535 },
536 /**
537  * @description: 正则表达式根据cookie的key匹配cookie的value
538  * @param {string} key
539  * @return {*}
540  */
541 getCookie: function (key) {
542     const str = `(\\ )${key}=(.*)?(;|$)`
543     const reg = new RegExp(str)
544     const arr = document.cookie.match(reg)
545     if (!arr) {
546         return null
547     }
548     return arr[2] // 第2个分组匹配对应cookie的value
549 },
550 randomSleep: async function (number, number1, fn = null) {
551     var a = Math.round(Math.random() * (number1 - number) + number) * 1000
552     debug && console.log(`即将等待${(a / 1000)}秒(${number}~${number1}秒)`)
553     // GM_getValue('allSettings')['showHint_enabled'] &&
554     // myToast.normal(
555     //     'success',
556     //     `即将等待${(a / 1000)}秒(${number}~${number1}秒)`,
557     //     1000,
558     //     'center'
559     // )
560     await this.sleep(a)
561     return fn ? fn() : null
562 },
563 checkName: function (name) {
564     const reg = new RegExp(
565         "[~!@#$%^&*()=|{}':;'\\"
566     )
567     let newName = ''
568     for (var i = 0, l = name.length; i < name.length; i++) {

```

```

569         newName = newName + name.substr(i, 1).replace(reg, '')
570     }
571     return newName
572 }
573 }
574
575 const _config_blackListArr = myUtils.uniqueArr(
576     _config_kgCompanyBlackList.concat(_config_blackListArr_1)
577 )
578
579 /**
580  * 悬浮窗样式
581  */
582 GM_addStyle(
583
584     `#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}`
585 )
586
587 /**
588  * 以下值，不要修改!
589  */
590 GM_setValue('_jobListJson_boss', null)
591 GM_setValue('_zp_token', null)
592 GM_setValue('_z1_currentJobNumbers', null)
593 // GM_setValue('_z1_joblist', null)
594 GM_setValue('_z1_params_details', null)
595 GM_setValue('_z1_params_cityIds', null)
596 GM_setValue('_lp_params_resId', null)
597 GM_setValue('_lp_params_jobCardList', null)
598 GM_setValue('_lg_params', null)
599 GM_setValue('_lg_resumeId', null)
600 GM_setValue('_lg_positionIds', null)
601
602 /**
603  * 适配不同浏览器的匹配选择器
604  */
605 Element.prototype.matches = Element.prototype.matches || ``
606 Element.prototype.matchesSelector ||
607
608 Element.prototype.webkitMatchesSelector ||
609 Element.prototype.msMatchesSelector ||
610 Element.prototype.mozMatchesSelector
611
612 var curl = location.href
613
614 var isShowTip
615 if (GM_listValues().indexOf('allSettings') === -1) {
616     GM_setValue('allSettings', {})
617 }
618
619 var curSettings = GM_getValue('allSettings')
620 // 设置自动识别初始值
621 if (curSettings['autoBlocking_enabled'] === undefined) {
622     curSettings['autoBlocking_enabled'] = true
623     GM_setValue('allSettings', curSettings)
624 }
625

```

```

626 // 设置开启提示初始值
627 if (curSettings['showHint_enabled'] === undefined) {
628     curSettings['showHint_enabled'] = true
629     GM_setValue('allSettings', curSettings)
630     isShowTip = GM_getValue('allSettings')['showHint_enabled']
631 }
632
633 // 设置隐藏或提醒黑名单公司
634 if (curSettings['isShowBlockingCompany'] === undefined) {
635     curSettings['isShowBlockingCompany'] = true
636     GM_setValue('allSettings', curSettings)
637 }
638
639 // 是否开启自动发送简历功能
640 if (curSettings['autoSendResume_enabled'] === undefined) {
641     curSettings['autoSendResume_enabled'] = false
642     GM_setValue('allSettings', curSettings)
643 }
644
645 var autoBlocking_enabled_menuId
646 var showHint_enabled_menuId
647 var isShowBlockingCompany_menuId
648 var autoSendResume_enabled_menuId
649 // 刷新新菜单
650 function registerMenu() {
651     // 是否开启自动屏蔽黑名单功能
652     GM_unregisterMenuCommand(autoBlocking_enabled_menuId)
653     if (curSettings['autoBlocking_enabled']) {
654         autoBlocking_enabled_menuId = GM_registerMenuCommand(
655             '自动屏蔽当前页面公司 [状态: √ ]',
656             function () {
657                 curSettings['autoBlocking_enabled'] = false
658                 GM_setValue('allSettings', curSettings)
659                 registerMenu()
660                 GM_getValue('allSettings')['showHint_enabled'] &&
661                 myToast.normal(
662                     'success',
663                     '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！',
664                     3000
665                 )
666             }
667         )
668     } else {
669         autoBlocking_enabled_menuId = GM_registerMenuCommand(
670             '自动屏蔽当前页面公司 [状态: x ]',
671             function () {
672                 curSettings['autoBlocking_enabled'] = true
673                 GM_setValue('allSettings', curSettings)
674                 registerMenu()
675                 GM_getValue('allSettings')['showHint_enabled'] &&
676                 myToast.normal(
677                     'success',
678                     '已开启自动屏蔽当前页面公司功能，请刷新网页！',
679                     3000
680                 )
681             }
682         )
683     }
684
685 // 各类提示
686 GM_unregisterMenuCommand(showHint_enabled_menuId)
687 if (curSettings['showHint_enabled']) {
688     showHint_enabled_menuId = GM_registerMenuCommand(
689         '提示信息 [状态: √ ]',
690         function () {
691             GM_getValue('allSettings')['showHint_enabled'] &&
692             myToast.normal(

```



```

693         'success',
694         '提示功能已关闭, 再次开启前将无任何提示! ',
695         3000
696     )
697     curSettings['showHint_enabled'] = false
698     GM_setValue('allSettings', curSettings)
699     isShowTip = GM_getValue('allSettings')['showHint_enabled']
700     registerMenu()
701 }
702 )
703 } else {
704     showHint_enabled_menuId = GM_registerMenuCommand(
705         '提示信息 [状态: x ]',
706         function () {
707             curSettings['showHint_enabled'] = true
708             GM_setValue('allSettings', curSettings)
709             isShowTip = GM_getValue('allSettings')['showHint_enabled']
710             registerMenu()
711             GM_getValue('allSettings')['showHint_enabled'] &&
712                 myToast.normal('success', '提示功能已开启! ', 3000)
713         }
714     )
715 }
716 // 提示音
717 GM_unregisterMenuCommand(isShowBlockingCompany_menuId)
718 if (curSettings['isShowBlockingCompany']) {
719     isShowBlockingCompany_menuId = GM_registerMenuCommand(
720         '显示 [状态: √ ]',
721         function () {
722             GM_getValue('allSettings')['showHint_enabled'] &&
723                 myToast.normal('success', '隐藏! ', 3000)
724             curSettings['isShowBlockingCompany'] = false
725             GM_setValue('allSettings', curSettings)
726             registerMenu()
727         }
728     )
729 } else {
730     isShowBlockingCompany_menuId = GM_registerMenuCommand(
731         '隐藏 [状态: x ]',
732         function () {
733             curSettings['isShowBlockingCompany'] = true
734             GM_setValue('allSettings', curSettings)
735             registerMenu()
736             GM_getValue('allSettings')['showHint_enabled'] &&
737                 myToast.normal('success', '显示! ', 3000)
738         }
739     )
740 }
741 // 自动投简历
742 GM_unregisterMenuCommand(autoSendResume_enabled_menuId)
743 if (curSettings['autoSendResume_enabled']) {
744     autoSendResume_enabled_menuId = GM_registerMenuCommand(
745         '自动投简历 [状态: √ ]',
746         function () {
747             GM_getValue('allSettings')['showHint_enabled'] &&
748                 myToast.normal('success', '自动投简历功能已关闭! ', 3000)
749             curSettings['autoSendResume_enabled'] = false
750             GM_setValue('allSettings', curSettings)
751             registerMenu()
752         }
753     )
754 } else {
755     autoSendResume_enabled_menuId = GM_registerMenuCommand(
756         '自动投简历 [状态: x ]',
757         function () {
758             curSettings['autoSendResume_enabled'] = true
759             GM_setValue('allSettings', curSettings)

```

```

760     registerMenu()
761     GM_getValue('allSettings')['showHint_enabled'] &&
762     myToast.normal('success', '自动投简历功能已开启! ', 3000)
763 }
764 )
765 }
766 }
767
768 registerMenu()
769 // var lg_isOpenGetCollectGif = true
770 var curPageUrlArr_jobList = myUtils.findSthFromObjArr(
771     curl,
772     _config_jobListPageUrlArr,
773     'url'
774 )
775
776 var curPageUrlArr_jobDetail = myUtils.findSthFromObjArr(
777     curl,
778     _config_jobDetailPageUrlArr,
779     'url'
780 )
781
782 var curPageUrlArr_companyDetail = myUtils.findSthFromObjArr(
783     curl,
784     _config_companyDetailPageUrlArr,
785     'url'
786 )
787
788 function addXMLHttpRequestCallback(callback) {
789     var oldSend, i, setRequestHeader
790
791     if (XMLHttpRequest.callbacks) {
792         XMLHttpRequest.callbacks.push(callback)
793     } else {
794         XMLHttpRequest.callbacks = [callback]
795     }
796
797     oldSend = XMLHttpRequest.prototype.send
798
799     XMLHttpRequest.prototype.send = function () {
800         for (i = 0; i < XMLHttpRequest.callbacks.length; i++) {
801             XMLHttpRequest.callbacks[i](this)
802         }
803         try {
804             oldSend.apply(this, arguments)
805         } catch (error) {
806             console.log('error: ', error)
807         }
808     }
809
810     setRequestHeader = XMLHttpRequest.prototype.setRequestHeader
811     XMLHttpRequest.prototype.setRequestHeader = {}
812     XMLHttpRequest.prototype.setRequestHeader = function (name, value) {
813         XMLHttpRequest.prototype.setRequestHeader[name] = value
814         try {
815             setRequestHeader.call(this, name, value)
816         } catch (error) {
817             console.log('error: ', error)
818         }
819     }
820 }
821
822 if (GM_getValue('allSettings')['autoSendResume_enabled']) {
823     isShowTip &&
824     myToast.normal('success', '你已开启自动投简历! ', 3000, 'center') &&
825     console.log('success', '你已开启自动投简历! ', 3000, 'center')
826 }

```

```

827 function autoSend() {
828     if (
829         hijacking_enabled &&
830         (curPageUrlArr_jobList.length ||
831          curPageUrlArr_jobDetail.length ||
832          curPageUrlArr_companyDetail.length)
833     ) {
834         console.log(1111)
835         addXMLRequestCallback(function (xhr) {
836             xhr.addEventListener('load', function () {
837                 if (xhr.readyState === 4 && xhr.status === 200) {
838                     var fRes
839                     var resUrl = new URL(xhr.responseURL)
840                     if (resUrl.origin.indexOf('apic.liepin.com') !== -1) {
841                         if (
842                             resUrl.pathname ===
843                             '/api/com.liepin.usercx.pc.user.base-property'
844                         ) {
845                             if (isTimesExhausted_lp) {
846                                 debug && console.log(errMsg_lp)
847                                 isShowTip &&
848                                 myToast.normal('error', errMsg_lp, 3000, 'center')
849                                 return false
850                             }
851                             fRes = JSON.parse(xhr.responseText)
852                             // GM_getValue('allSettings')['autoSendResume_enabled'] &&
853                             GM_sendMessage('_lp_params_resId', fRes.data.resId)
854                         } else if (
855                             resUrl.pathname ===
856                             '/api/com.liepin.searchfront4c.pc-search-job'
857                         ) {
858                             if (isTimesExhausted_lp) {
859                                 debug && console.log(errMsg_lp)
860                                 isShowTip &&
861                                 myToast.normal('error', errMsg_lp, 3000, 'center')
862                                 return false
863                             }
864                             fRes = JSON.parse(xhr.responseText)
865                             // GM_getValue('allSettings')['autoSendResume_enabled'] &&
866                             GM_sendMessage(
867                                 '_lp_params_jobCardList',
868                                 fRes.data.data.jobCardList
869                             )
870                         }
871                     } else if (resUrl.origin.indexOf('zhipin.com') !== -1) {
872                         if (isTimesExhausted_bs) {
873                             debug && console.log(errMsg_bs)
874                             isShowTip && myToast.normal('error', errMsg_bs, 3000, 'center')
875                             fRes = JSON.parse(xhr.responseText)
876                             return false
877                         } else if (
878                             resUrl.pathname === '/wapi/zpgeek/search/joblist.json'
879                         ) {
880                             fRes = JSON.parse(xhr.responseText)
881                             // GM_getValue('allSettings')['autoSendResume_enabled'] &&
882                             GM_sendMessage('_jobListJson_boss', fRes.zpData.jobList)
883                         } else if (resUrl.pathname === '/wapi/zppassport/get/zpToken') {
884                             console.log(222)
885                             try {
886                                 fRes = JSON.parse(xhr.responseText)
887                                 // GM_getValue('allSettings')['autoSendResume_enabled'] &&
888                                 GM_sendMessage('_zp_token', fRes.zpData.token)
889                             } catch (error) {
890                                 console.error('error: ', error)
891                             }
892                         } else if (
893                             resUrl.pathname === '/wapi/zpgeek/mobile/search/joblist.json'

```

```

894     ) {
895         fRes = JSON.parse(xhr.responseText)
896         // GM_getValue('allSettings')['autoSendResume_enabled'] &&
897         GM_sendMessage('_jobListJson_boss', fRes.zpData.html)
898         handleNewPageElements(2, 1, 500)
899     }
900 } else if (resUrl.origin.indexOf('fe-api.zhaopin.com') !== -1) {
901     if (
902         resUrl.pathname === '/c/i/user/detail'
903         // || resUrl.pathname === '/c/i/city-page/user-city'
904     ) {
905         fRes = JSON.parse(xhr.responseText)
906         // GM_getValue('allSettings')['autoSendResume_enabled'] &&
907         GM_sendMessage('_z1_params_details', {
908             _resumeNumber: fRes.data.Resume.ResumeNumber,
909             _at: resUrl.searchParams.get('at'),
910             _rt: resUrl.searchParams.get('rt'),
911             _x_zp_client_id: resUrl.searchParams.get('x-zp-client-id'),
912             _x_zp_page_request_id: resUrl.searchParams.get(
913                 'x-zp-page-request-id'
914             )
915         })
916     } else if (
917         resUrl.pathname === '/c/i/jobs/keyword-company-search' &&
918         resUrl.searchParams.get('filter_c_workCity') !== null
919     ) {
920         fRes = JSON.parse(xhr.responseText)
921         // GM_getValue('allSettings')['autoSendResume_enabled'] &&
922         GM_sendMessage(
923             '_z1_params_cityIds',
924             resUrl.searchParams.get('filter_c_workCity')
925         )
926     }
927 }
928 }
929 })
930 })
931 }
932 }
933 autoSend()
934
935 console.log({
936     msg: '注入完成',
937     time: new Date().valueOf(),
938     data: new Date()
939 })
940
941 /**
942  * 监听翻页-history模式
943  */
944 const _historyWrap = function (type) {
945
946     const orig = history[type]
947     const e = new Event(type)
948     return function () {
949         const rv = orig.apply(this, arguments)
950         e.arguments = arguments
951         window.dispatchEvent(e)
952         return rv
953     }
954 }
955 history.pushState = _historyWrap('pushState')
956 history.replaceState = _historyWrap('replaceState')
957
958 window.addEventListener('pushState', handleNewPage)
959
960 /**

```

```

961  * 判断当前网络是否通畅
962  * @returns Boolean
963  */
964  async function judgeNetState() {
965
966      var tryAttemptTimes = 0
967      const net_state = await myUtils.onLine()
968      if (!net_state) {
969          debug && console.log('网络中断, 请检查当前网络! 已为你结束当前操作! ')
970          isShowTip &&
971              myToast.normal(
972                  'warning',
973                  '网络中断, 请检查当前网络! 已为你结束当前操作! ',
974                  3000,
975                  'center'
976              )
977          if (tryAttemptTimes < 2) {
978              setTimeout(async () => {
979                  await judgeNetState()
980              }, 1500)
981              tryAttemptTimes++
982          } else {
983              return false
984          }
985      }
986      return true
987  }
988
989  /**
990   * 监听值的改变的回调函数
991   * @param {String} label 名字
992   * @param {Function} callback 回调函数
993   */
994   function GM_onMessage(label, callback) {
995
996       GM_addValueChangeListener(label, async function () {
997           await callback.apply(undefined, arguments[2])
998       })
999   }
1000  /**
1001   * 设置值
1002   * @param {String} label 名字
1003   */
1004   function GM_sendMessage(label) {
1005
1006       GM_setValue(label, Array.from(arguments).slice(1))
1007   }
1008
1009   // var lgCollectGifData = {}
1010   // var lgCollectGifHeader = {}
1011   if (hijacking_enabled && curPageUrlArr_jobList.length) {
1012       if (curPageUrlArr_jobList[0].siteType === 1) {
1013           /**
1014            * 监听boss的岗位列表, 用于发送简历
1015            */
1016           GM_onMessage('_jobListJson_boss', async function (src) {
1017               if (!GM_getValue('allSettings')['autoSendResume_enabled']) {
1018                   isShowTip &&
1019                       myToast.normal('error', '未开启自动投简历功能', 3000, 'center')
1020                   return debug && console.log('未开启自动投简历功能')
1021               }
1022               if (isTimesExhausted_bs) {
1023                   debug && console.log(errMsg_bs)
1024                   isShowTip && myToast.normal('error', errMsg_bs, 3000, 'center')
1025                   return false
1026               }
1027               var boss = new Boss()

```

```

1028
1029     var webselector = _config_jobListDomSelectorArr.filter((item) => {
1030         if (item.siteType === 1) {
1031             return true
1032         }
1033     })
1034     for (let i = 0; i < webselector.length; i++) {
1035         var target = webselector[i].el
1036         var a = await myUtils.getElement(document, target, 1500)
1037         if (a !== null) {
1038             var firstRes = document.querySelectorAll(target)
1039             if (!firstRes || firstRes === null) {
1040                 var _targetarr = target.split(' ')
1041
1042                 firstRes = document.getElementsByClassName(
1043                     _targetarr[_targetarr.length - 1].replace('.', '')
1044                 )
1045             }
1046             const [hitsCompanyArr] = await autoBlock_getCompanyNameArrFromPage(
1047                 myUtils.uniqueArr(_config_blackListArr),
1048                 myUtils.uniqueArr(_config_whiteListArr),
1049                 document.querySelectorAll(webselector[i].el),
1050                 2,
1051                 1
1052             )
1053             var newJobListJson_Boss = boss.removeBlackList(src, hitsCompanyArr)
1054
1055             if (GM_getValue('_zp_token') !== null) {
1056                 boss.zp_token = GM_getValue('_zp_token')[0]
1057                 boss._jobListJson_boss = newJobListJson_Boss
1058                 debug && console.log('开始发起沟通!')
1059                 isShowTip &&
1060                     myToast.normal('success', '开始发起沟通', 1500, 'center')
1061                 await boss.main()
1062             } else {
1063                 debug && console.warn('没有获取到zp_token哦, 尝试获取! ')
1064                 isShowTip &&
1065                     myToast.normal(
1066                         'warning',
1067                         '没有获取到zp_token哦, 尝试获取! ',
1068                         1500,
1069                         'center'
1070                     )
1071                 var zp_token = await boss.get_zp_token()
1072                 zp_token && (await boss.main(zp_token, newJobListJson_Boss))
1073                 !zp_token && debug && console.error('你还未登录! ')
1074                 isShowTip &&
1075                     myToast.normal('error', '你还未登录! ', 3000, 'center')
1076             }
1077         }
1078     }
1079 })
1080
1081 GM_onMessage('_zp_token', async (src) => {})
1082 } else if (curPageUrlArr_jobList[0].siteType === 2) {
1083     /**
1084     * 监听z1的城市id, 用于发送简历
1085     */
1086     GM_onMessage('_z1_params_cityIds', async (src) => {
1087         if (!GM_getValue('allSettings')['autoSendResume_enabled']) {
1088             isShowTip &&
1089                 myToast.normal('error', '未开启自动投简历功能', 1500, 'center')
1090             return debug && console.log('未开启自动投简历功能')
1091         }
1092         debug && console.log('[onMessage]', '_z1_params_cityIds', '=>', src)
1093         var z1 = new ZhiLian(src)
1094         await z1.main(src)

```

```

1095     })
1096
1097     GM_onMessage('_zl_params_details', async (src) => {
1098         debug && console.log('[onMessage]', '_zl_params_details', '=>', src)
1099     })
1100
1101     GM_onMessage('_zl_currentJobNumbers', async (src) => {
1102         debug && console.log('[onMessage]', '_zl_currentJobNumbers', '=>', src)
1103     })
1104 } else if (curPageUrlArr_jobList[0].siteType === 3) {
1105     /**
1106      * 监听lp的岗位列表，用于发送简历
1107      */
1108     GM_onMessage('_lp_params_jobCardList', async function (src) {
1109         if (!GM_getValue('allSettings')['autoSendResume_enabled']) {
1110             isShowTip &&
1111             myToast.normal('error', '未开启自动投简历功能', 1500, 'center')
1112             return debug && console.log('未开启自动投简历功能')
1113         }
1114         if (isTimesExhausted_lp) {
1115             isShowTip && myToast.normal('error', errMsg_lp, 1500, 'center')
1116             debug && console.log(errMsg_lp)
1117             return false
1118         }
1119         var _resId
1120         var lp = new LiePin(src, _resId)
1121         var curResId = GM_getValue('_lp_params_resId')
1122         var firstRunFlag = document.getElementById('uselessDiv1')
1123         if (firstRunFlag == null) {
1124             debug &&
1125             console.log('[onMessage]', '_lp_params_jobCardList', '=>', src)
1126             if (
1127                 curResId == null ||
1128                 !curResId ||
1129                 curResId !== '您的帐号信息已过期，请退出帐号重新登录'
1130             ) {
1131                 _resId = await lp.get_resId()
1132                 if (_resId !== '您的帐号信息已过期，请退出帐号重新登录') {
1133                     GM_setValue('_lp_params_resId', _resId)
1134                 } else {
1135                     isShowTip && myToast.normal('error', _resId, 1500, 'center')
1136                     return debug && console.error(_resId)
1137                 }
1138             } else {
1139                 _resId = curResId
1140             }
1141             debug && console.log(curResId)
1142
1143             const uselessDiv1 = document.createElement('div')
1144             uselessDiv1.id = 'uselessDiv1'
1145             uselessDiv1.style.display = 'none'
1146             document.body.append(uselessDiv1)
1147             setTimeout(async () => {
1148                 lp.resId = _resId
1149                 GM_getValue('_lp_params_jobCardList') !== null && (await lp.main())
1150                 setTimeout(() => {
1151                     var temperUseless = document.getElementById('uselessDiv1')
1152                     if (temperUseless !== null) {
1153                         uselessDiv1.parentNode.removeChild(uselessDiv1)
1154                     }
1155                 }, 5)
1156             }, 3000)
1157         } else {
1158             firstRunFlag.remove()
1159         }
1160     })
1161 }

```



```

1162     GM_onMessage('_lp_params_resId', async (src) => {
1163         debug && console.log('onMessage', '_lp_params_resId', '=>', src)
1164     })
1165 } else if (curPageUrlArr_jobList[0].siteType === 4) {
1166     GM_onMessage('_lg_params', async (src) => {
1167         debug && console.log('onMessage', '_lg_params', '=>', src)
1168         if (src !== undefined && src) {
1169             debug && console.log('-----')
1170             debug && console.log(src)
1171             debug && console.log(src['X-K-HEADER'])
1172             debug && console.log(src['X-SS-REQ-HEADER'])
1173             debug && console.log(src['aesKey'])
1174             debug && console.log(src['user_trace_token'])
1175             debug && console.log('-----')
1176         }
1177     })
1178 }
1179 }
1180
1181 /**
1182  * 第一次运行
1183  * @param {Array} blackListArr 黑名单列表
1184  * @param {Array} whiteListArr 白名单列表
1185  */
1186 async function firstRun(blackListArr, whiteListArr) {
1187
1188     if (GM_getValue('allSettings')['autoBlocking_enabled']) {
1189         var pageType = null
1190         if (
1191             curPageUrlArr_jobDetail.length &&
1192             !curPageUrlArr_companyDetail.length
1193         ) {
1194             debug && console.info('检测到当前可能为岗位详情页')
1195             GM_getValue('allSettings')['showHint_enabled'] &&
1196                 myToast.normal(
1197                     'success',
1198                     '检测到当前可能为岗位详情页',
1199                     1500,
1200                     'top-end'
1201                 )
1202             pageType = 1
1203             autoBlock_mainFn(
1204                 true,
1205                 blackListArr,
1206                 whiteListArr,
1207                 pageType,
1208                 _config_detailListDomSelectorArr,
1209                 curPageUrlArr_jobDetail[0].siteType,
1210                 true,
1211                 2000
1212             )
1213         } else if (
1214             (curPageUrlArr_companyDetail.length &&
1215                 !curPageUrlArr_jobDetail.length &&
1216                 !curPageUrlArr_jobList.length) ||
1217             (curPageUrlArr_companyDetail.length &&
1218                 curPageUrlArr_jobList.length &&
1219                 location.href.indexOf('gongsi') !== -1) ||
1220             (curPageUrlArr_companyDetail.length &&
1221                 !curPageUrlArr_jobDetail.length &&
1222                 curPageUrlArr_jobList.length)
1223         ) {
1224             debug && console.info('检测到当前可能为公司详情页')
1225             GM_getValue('allSettings')['showHint_enabled'] &&
1226                 myToast.normal(
1227                     'success',
1228                     '检测到当前可能为公司详情页',

```

```

1229         1500,
1230         'top-end'
1231     )
1232     pageType = 1
1233     autoBlock_mainFn(
1234         true,
1235         blackListArr,
1236         whiteListArr,
1237         curPageUrlArr_companyDetail.length &&
1238         curPageUrlArr_jobList.length &&
1239         document.title.indexOf('全国公司') !== -1
1240         ? 2
1241         : pageType,
1242         _config_detailListDomSelectorArr,
1243         curPageUrlArr_companyDetail[0].siteType,
1244         true,
1245         2000
1246     )
1247 } else if (
1248     curPageUrlArr_jobList.length &&
1249     !curPageUrlArr_jobDetail.length &&
1250     !curPageUrlArr_companyDetail.length
1251 ) {
1252     debug && console.info('检测到当前页面可能为岗位列表页')
1253     GM_getValue('allSettings')['showHint_enabled'] &&
1254     myToast.normal(
1255         'success',
1256         '检测到当前页面可能为岗位列表页',
1257         1500,
1258         'top-end'
1259     )
1260     pageType = 2
1261     setTimeout(
1262         () => {
1263             autoBlock_mainFn(
1264                 true,
1265                 blackListArr,
1266                 whiteListArr,
1267                 pageType,
1268                 _config_jobListDomSelectorArr,
1269                 curPageUrlArr_jobList[0].siteType,
1270                 true,
1271                 2000
1272             )
1273         },
1274         curPageUrlArr_jobList[0].siteType !== 4 ? 1500 : 4500
1275     )
1276 } else {
1277     setTimeout(() => {
1278         var curPageUrlArr_jobList1 = myUtils.findSthFromObjArr(
1279             location.href,
1280             _config_jobListPageUrlArr,
1281             'url'
1282         )
1283         var curPageUrlArr_jobDetail1 = myUtils.findSthFromObjArr(
1284             location.href,
1285             _config_jobDetailPageUrlArr,
1286             'url'
1287         )
1288         var curPageUrlArr_companyDetail1 = myUtils.findSthFromObjArr(
1289             location.href,
1290             _config_companyDetailPageUrlArr,
1291             'url'
1292         )
1293         if (
1294             (curPageUrlArr_jobList1.length &&
1295             curPageUrlArr_jobDetail1.length &&

```

```

1296         curPageUrlArr_companyDetail1.length) ||
1297         (curPageUrlArr_jobList1.length &&
1298         !curPageUrlArr_jobDetail1.length &&
1299         curPageUrlArr_companyDetail1.length) ||
1300         (curPageUrlArr_jobList1.length &&
1301         curPageUrlArr_jobDetail1.length &&
1302         !curPageUrlArr_companyDetail1.length) ||
1303         (!curPageUrlArr_jobList1.length &&
1304         curPageUrlArr_jobDetail1.length &&
1305         curPageUrlArr_companyDetail1.length)
1306     ) {
1307         debug && console.info('奇了个怪! 快来给我反馈一下! ')
1308         GM_getValue('allSettings')['showHint_enabled'] &&
1309         myToast.multipleNotControl(
1310             'error',
1311             '出错了! ',
1312             1500,
1313             false,
1314             '奇了个怪! 快来给我反馈一下! ',
1315             () => {}
1316         )
1317     } else {
1318         debug &&
1319         console.log(
1320             '当前页面暂不支持! ',
1321             location.origin + location.pathname
1322         )
1323     }
1324 }, 1500)
1325 }
1326 }
1327 }
1328
1329 /**
1330  *
1331  * @param {Array} blacklistArr 黑名单列表
1332  * @param {Array} whitelistArr 白名单列表
1333  * @param {Node} webSelector 网页dom选择器 querySelectAll
1334  * @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
1335  * @param {Number} siteType 站点类型 1.boss 2.智联 3.猎聘 4.拉勾
1336  * @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
1337  */
1338 function autoBlock_getCompanyNameArrFromPage(
1339     blacklistArr,
1340     whitelistArr,
1341     webSelector,
1342     pageType,
1343     siteType,
1344     webSelectorMore
1345 ) {
1346     return new Promise((resolve, reject) => {
1347         try {
1348             if (
1349                 typeof blacklistArr === 'object' &&
1350                 typeof whitelistArr === 'object' &&
1351                 typeof webSelector === 'object' &&
1352                 typeof pageType === 'number' &&
1353                 typeof siteType === 'number' &&
1354                 webSelector.length > 0
1355             ) {
1356                 var newArr = webSelectorMore
1357                 ? webSelectorMore.map((item) => {
1358                     var wbStor = document.querySelectorAll(item)
1359                     var temp = [].slice.call(wbStor, 0)
1360                     return temp

```

```

1362         ? temp.map(function (e1, index) {
1363             var txt = e1.innerText
1364             if (
1365                 (siteType === 1 &&
1366                     wbStor[0].parentNode.className === 'user-info') ||
1367                 (siteType === 2 &&
1368                     wbStor[0].parentNode.className ===
1369                     'base-info__title') ||
1370                 (siteType === 2 &&
1371                     wbStor[0].parentNode.className === 'industry-mask')
1372             ) {
1373                 txt = txt.split('\n')[0]
1374             } else if (
1375                 siteType === 3 &&
1376                 wbStor[0].parentNode.className === 'company-info'
1377             ) {
1378                 txt = txt.split(' ')[0]
1379             }
1380             return [txt, index, wbStor[index]]
1381         })
1382         : []
1383     })
1384     : []
1385
1386     var allCompanyArr = [].slice.call(webSelector, 0)
1387
1388     var _allCompanyNameArr = allCompanyArr
1389     ? allCompanyArr.map(function (e1, index) {
1390         var txt = e1.innerText
1391         if (
1392             siteType === 2 &&
1393             webSelector[0].parentNode.className === 'base-info__title'
1394         ) {
1395             txt = txt.split('\n')[0]
1396         }
1397         return [txt, index, webSelector[index]]
1398     })
1399     : []
1400     debug && console.log('_allCompanyNameArr: ', _allCompanyNameArr)
1401     // [...arr1,...arr2]
1402     var _newArr = []
1403     for (let i = 0; i <= newArr.length - 1; i++) {
1404         _newArr.push(...newArr[i])
1405     }
1406     var allCompanyNameArr = [..._newArr, ..._allCompanyNameArr]
1407     debug && console.log('allCompanyNameArr: ', allCompanyNameArr)
1408
1409     var hitsCompanyArr = []
1410     var pattern = null
1411
1412     const newBlackListArr = blackListArr.filter((blackCompany) => {
1413         return !whiteListArr.find((whiteCompany) => {
1414             return (
1415                 whiteCompany.indexOf(blackCompany) !== -1 ||
1416                 whiteCompany.match(
1417                     new RegExp(myUtils.checkName(blackCompany), 'ig')
1418                 ) ||
1419                 whiteCompany === blackCompany
1420             )
1421         })
1422     })
1423
1424     for (var index = 0; index < allCompanyNameArr.length; index++) {
1425         var curAllCompanyName = allCompanyNameArr[index][0]
1426         for (var i = 0; i < newBlackListArr.length; i++) {
1427             var curNewBlackListName = newBlackListArr[i]
1428             if (curAllCompanyName === curNewBlackListName) {

```

```

1429         pattern = 1
1430     } else if (
1431         curAllCompanyName.indexOf(curNewBlackListName) !== -1 ||
1432         myUtils
1433             .checkName(curAllCompanyName)
1434             .match(
1435                 new RegExp(myUtils.checkName(curNewBlackListName), 'ig')
1436             ) ||
1437         curNewBlackListName.indexOf(curAllCompanyName) !== -1 ||
1438         myUtils
1439             .checkName(curNewBlackListName)
1440             .match(new RegExp(myUtils.checkName(curAllCompanyName), 'ig'))
1441     ) {
1442         pattern = 2
1443     } else {
1444         pattern = null
1445     }
1446
1447     pattern &&
1448         hitsCompanyArr.unshift({
1449             bName: curNewBlackListName,
1450             oName: pattern === 2 ? curAllCompanyName : null,
1451             documentIndex: allCompanyNameArr[index][1],
1452             elementNode: allCompanyNameArr[index][2],
1453             pattern
1454         })
1455     }
1456 }
1457
1458 const notHitsCompanyArr = allCompanyNameArr.filter((item) => {
1459     return !hitsCompanyArr.find((blackCompany) => {
1460         return (
1461             blackCompany.elementNode === item[2] &&
1462             blackCompany.documentIndex === item[1]
1463         )
1464     })
1465 })
1466
1467 debug &&
1468     console.log(
1469         '[hitsCompanyArr, allCompanyArr, siteType, notHitsCompanyArr]: ',
1470         [hitsCompanyArr, allCompanyArr, siteType, notHitsCompanyArr]
1471     )
1472     resolve([hitsCompanyArr, allCompanyArr, siteType, notHitsCompanyArr])
1473 } else {
1474     GM_getValue('allSettings')['showHint_enabled'] &&
1475         myToast.normal('warning', '暂不支持当前页面!', 3000)
1476 }
1477 } catch (e) {
1478     if (e.message.indexOf('The provided selector is empty.') !== -1) {
1479         GM_getValue('allSettings')['showHint_enabled'] &&
1480             myToast.normal('warning', '暂不支持当前页面!', 3000)
1481     }
1482     reject(e.message)
1483 }
1484 })
1485 }
1486
1487 /**
1488  * 处理返回的结果
1489  * @param {Array} hitsCompanyArr 结果数组
1490  * @param {Array} allCompanyArr 从页面中读取到的公司名字数组
1491  * @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
1492  * @param {Number} siteType 站点类型 1.boss 2.智联 3.猎聘 4.拉勾
1493  */
1494 async function autoBlock_handleHitsCompanyArr(
1495

```

```

1496     hitsCompanyArr,
1497     allCompanyArr,
1498     pageType,
1499     siteType
1500   ) {
1501     if (typeof hitsCompanyArr !== 'object' || typeof pageType !== 'number') {
1502       return false
1503     }
1504     hitsCompanyArr = myUtils.uniqueFunc(hitsCompanyArr, 'documentIndex')
1505     if (hitsCompanyArr && hitsCompanyArr.length !== 0) {
1506       if (pageType === 1) {
1507         Swal.fire({
1508           title: '发现黑名单公司!!',
1509           type: 'error',
1510           html: `<span>公司名: <span style="color:red">${
1511             hitsCompanyArr[0].oName
1512               ? `${hitsCompanyArr[0].oName}`
1513               : `${hitsCompanyArr[0].bName}`
1514           }</span></span><br/>${
1515             hitsCompanyArr[0].bName
1516               ? `<span>黑名单中匹配到的公司名: <span
1517 style="color:green">${hitsCompanyArr[0].bName}</span></span>`
1518               : ''
1519             }<br/><span style="color:#6c757d;text-align: center">-----
1520 -----<br/><span style="color:#6c757d">因为该公司在你填写的黑名单公司列表哦! <br/>如果有误判可
1521 以移除该公司! </span><br/><-----<br/><iframe
1522 src="https://kjrate.com?post_type=question&s=${encodeURIComponent(
1523   hitsCompanyArr[0].bName
1524   )}" style="width: 100%;height: 15rem"></iframe>`,
1525           cancelButtonText: '收到',
1526           confirmButtonText: '确定',
1527           showCancelButton: true,
1528           showConfirmButton: true,
1529           allowOutsideClick: false,
1530           allowEscapeKey: false,
1531           allowEnterKey: false
1532         }).then((result) => {}))
1533       } else if (pageType === 2) {
1534         var tempString = ''
1535         console.log('hitsCompanyArr: ', hitsCompanyArr)
1536         for (const [index, item] of hitsCompanyArr.entries()) {
1537           tempString += `第${index + 1}家: ${
1538             item.oName ? `${item.oName} => ` : ''
1539           }${item.bName}\\n`
1540           await autoBlock_switchBlackMode(
1541             GM_getValue('allSettings')['isShowBlockingCompany'],
1542             // allCompanyArr[item.documentIndex],
1543             item.elementNode,
1544             item.pattern,
1545             siteType
1546           )
1547         }
1548         GM_getValue('allSettings')['showHint_enabled'] &&
1549         myToast.normal(
1550           'info',
1551           `匹配到 ${
1552             hitsCompanyArr.length
1553           } 条黑名单公司! 分别是: \\n${tempString.substring(
1554             0,
1555             tempString.length - 1
1556           )}`,
1557           50000
1558         )
1559       }
1560       debug &&
1561       console.info(
1562         'info',
1563         `匹配到 ${

```

```

1559         hitsCompanyArr.length
1560     } 条黑名单公司! 分别是: \n${tempString.substring(
1561         0,
1562         tempString.length - 1
1563     )}`
1564     )
1565 }
1566 } else {
1567     GM_getValue('allSettings')['showHint_enabled'] &&
1568     myToast.normal(
1569         pageType !== 1 && pageType !== 2 ? 'error' : 'success',
1570         `${
1571             pageType === 1
1572             ? '这家不在黑名单列表哦!'
1573             : pageType === 2
1574             ? '当前页面没有黑名单公司!'
1575             : '奇了怪了! 找我反馈一下吧'
1576         }`,
1577         3000
1578     )
1579     debug &&
1580     console.log(
1581         `${
1582             pageType === 1
1583             ? '这家不在黑名单列表哦!'
1584             : pageType === 2
1585             ? '当前页面没有黑名单公司!'
1586             : '奇了怪了! 找我反馈一下吧'
1587         }`
1588     )
1589 }
1590 return true
1591 }
1592 /**
1593  *
1594  * @param {true} isShow 是否显示
1595  * @param {Node} target 要操作的dom节点
1596  * @param {Number} pattern 匹配模式 1.完全匹配 2.模糊匹配
1597  * @param {Number} siteType 站点类型 1.boss 2.智联 3.猎聘 4.拉勾
1598  */
1599 async function autoBlock_switchBlackMode(isShow, target, pattern, siteType) {
1600
1601     var bgColor = pattern === 1 ? 'red' : pattern === 2 ? 'orange' : ''
1602     var curl = location.href
1603     var parentNodeTarget = target.parentNode
1604     var parentNodeTargetClassName = parentNodeTarget.className
1605     debug &&
1606     console.log('target:', target, 'parentNodeTarget: ', parentNodeTarget)
1607     debug && console.log('curl: ', curl)
1608     if (!isShow) {
1609         switch (siteType) {
1610             case 1:
1611                 if (curl.match('https://www.zhipin.com/xiaoyuan/') !== null) {
1612                     switch (parentNodeTargetClassName) {
1613                         case 'company-text':
1614                             parentNodeTarget.parentNode.parentNode.parentNode.remove()
1615                             break
1616                         default:
1617                             parentNodeTarget.parentNode.parentNode.remove()
1618                             break
1619                     }
1620                 } else if (
1621                     curl.match('https://www.zhipin.com/returnee_jobs/') !== null
1622                 ) {
1623                     switch (parentNodeTargetClassName) {
1624                         case 'sub-li-bottom-commany-info':
1625                             parentNodeTarget.parentNode.parentNode.remove()

```



```

1626         break
1627     case 'company-text':
1628         parentNodeTarget.parentNode.parentNode.parentNode.remove()
1629         break
1630     default:
1631         parentNodeTarget.parentNode.parentNode.parentNode.remove()
1632         break
1633 }
1634 } else if (curl.match('https://www.zhipin.com/(.*?)') !== null) {
1635     switch (parentNodeTargetClassName) {
1636         case 'company-info':
1637             parentNodeTarget.parentNode.parentNode.remove()
1638             break
1639         default:
1640             parentNodeTarget.parentNode.parentNode.parentNode.parentNode.remove()
1641             break
1642     }
1643 } else if (curl.indexOf('m.zhipin.com') !== -1) {
1644     parentNodeTarget.parentNode.parentNode.remove()
1645 } else {
1646     parentNodeTarget.parentNode.parentNode.parentNode.remove()
1647 }
1648 break
1649 case 2:
1650     if (curl.match('https://www.zhaopin.com/.*/') !== null) {
1651         switch (parentNodeTargetClassName) {
1652             case `list__item__content`:
1653                 parentNodeTarget.parentNode.parentNode.remove()
1654                 break
1655             case 'industry-mask':
1656                 parentNodeTarget.parentNode.parentNode.parentNode.remove()
1657                 break
1658             case 'list__item__content recomand-list__item__content':
1659                 parentNodeTarget.parentNode.parentNode.remove()
1660                 break
1661             default:
1662                 if (
1663                     parentNodeTarget.parentNode.className ===
1664                     'zp-company-hiring-list__info__list'
1665                 ) {
1666                     parentNodeTarget.parentNode.parentNode.parentNode.remove()
1667                 } else {
1668                     parentNodeTarget.parentNode.parentNode.parentNode.remove()
1669                 }
1670                 break
1671             }
1672         } else if (curl.match('https://www.zhaopin.com/') !== null) {
1673             switch (parentNodeTargetClassName) {
1674                 case 'home-ad-a__company-info':
1675                 case 'home-ad-b__company-info':
1676                 case 'home-ad-c__company-info home-ad-c__company-info--pl60':
1677                     parentNodeTarget.parentNode.parentNode.parentNode.remove()
1678                     break
1679                 default:
1680                     parentNodeTarget.parentNode.parentNode.parentNode.remove()
1681                     break
1682             }
1683         } else if (curl.match('https://xiaoyuan.zhaopin.com/') !== null) {
1684             switch (parentNodeTargetClassName) {
1685                 case `normal-show-text-table`:
1686                     parentNodeTarget.parentNode.parentNode.parentNode.remove()
1687                     break
1688                 default:
1689                     if (
1690                         parentNodeTarget.parentNode.parentNode.parentNode.parentNode
1691                         .className === 'newset-content clearfix'
1692                     ) {

```

```

1693         parentNodeTarget.parentNode.parentNode.remove()
1694     } else {
1695         parentNodeTarget.parentNode.parentNode.parentNode.remove()
1696     }
1697     break
1698 }
1699 } else {
1700     parentNodeTarget.parentNode.parentNode.parentNode.parentNode.remove()
1701 }
1702 break
1703 case 3:
1704     if (curl.match('https://www.liepin.com/campus/?t=1') !== null) {
1705         switch (parentNodeTargetClassName) {
1706             case 'jsx-3205826082 company-list-item-content':
1707                 parentNodeTarget.parentNode.parentNode.parentNode.remove()
1708                 break
1709             case 'jobcard2-company-info':
1710                 parentNodeTarget.parentNode.parentNode.parentNode.remove()
1711                 break
1712             case 'company-info':
1713                 parentNodeTarget.parentNode.parentNode.remove()
1714                 break
1715             default:
1716                 parentNodeTarget.parentNode.parentNode.parentNode.remove()
1717                 break
1718         }
1719     } else if (curl.match('https://www.liepin.com') !== null) {
1720         switch (parentNodeTargetClassName) {
1721             case 'jobcard2-company-info':
1722                 parentNodeTarget.parentNode.parentNode.parentNode.remove()
1723                 break
1724             case 'home-comp-info-content':
1725                 parentNodeTarget.parentNode.parentNode.parentNode.remove()
1726                 break
1727             default:
1728                 parentNodeTarget.parentNode.parentNode.parentNode.remove()
1729                 break
1730         }
1731     } else {
1732         parentNodeTarget.parentNode.parentNode.parentNode.parentNode.remove()
1733     }
1734     break
1735 default:
1736     debug && console.error('隐藏失败! ')
1737     break
1738 }
1739 } else {
1740     switch (siteType) {
1741         case 1:
1742             if (curl.indexOf('m.zhipin.com') !== -1) {
1743                 changeDom(parentNodeTarget.parentNode.parentNode, bgColor, isShow)
1744             } else if (curl.indexOf('zhipin.com/gongsi/') !== -1) {
1745                 changeDom(parentNodeTarget.parentNode.parentNode, bgColor, isShow)
1746             } else if (curl.match('https://www.zhipin.com/(.*)') !== null) {
1747                 if (parentNodeTargetClassName === 'company-info') {
1748                     changeDom(parentNodeTarget.parentNode.parentNode, bgColor, isShow)
1749                 } else {
1750                     changeDom(
1751                         parentNodeTarget.parentNode.parentNode.parentNode,
1752                         bgColor,
1753                         isShow
1754                     )
1755                 }
1756             } else if (curl.match('https://www.zhipin.com/xiaoyuan/') !== null) {
1757                 if (parentNodeTargetClassName === 'company-text') {
1758                     changeDom(
1759                         parentNodeTarget.parentNode.parentNode.parentNode,

```



```

1827         }
1828         break
1829     }
1830     } else if (curl.match('https://www.zhaopin.com/') !== null) {
1831         switch (parentNodeTargetClassName) {
1832             case 'home-ad-a__company-info':
1833             case 'home-ad-b__company-info':
1834             case 'home-ad-c__company-info home-ad-c__company-info--pl60':
1835                 changeDom(parentNodeTarget.parentNode, bgColor, isShow)
1836                 break
1837             default:
1838                 changeDom(parentNodeTarget.parentNode, bgColor, isShow)
1839                 break
1840         }
1841     } else if (curl.match('https://xiaoyuan.zhaopin.com/') !== null) {
1842         switch (parentNodeTargetClassName) {
1843             case `normal-show-text-table`:
1844                 changeDom(
1845                     parentNodeTarget.parentNode.parentNode.parentNode,
1846                     bgColor,
1847                     isShow
1848                 )
1849                 break
1850             default:
1851                 if (
1852                     parentNodeTarget.parentNode.parentNode.parentNode.parentNode
1853                     .className === 'newset-content clearfix'
1854                 ) {
1855                     changeDom(
1856                         parentNodeTarget.parentNode.parentNode,
1857                         bgColor,
1858                         isShow
1859                     )
1860                 }
1861                 break
1862         }
1863     } else {
1864         changeDom(
1865             parentNodeTarget.parentNode.parentNode.parentNode,
1866             bgColor,
1867             isShow
1868         )
1869         parentNodeTarget.parentNode.parentNode.nextElementSibling.style.background =
1870             bgColor
1871     }
1872     break
1873 case 3:
1874     if (curl.match('https://www.liepin.com/campus/?t=1') !== null) {
1875         switch (parentNodeTargetClassName) {
1876             case 'jsx-3205826082 company-list-item-content':
1877                 changeDom(
1878                     parentNodeTarget.parentNode.parentNode,
1879                     bgColor,
1880                     isShow
1881                 )
1882                 break
1883             case 'jobcard2-company-info':
1884                 changeDom(
1885                     parentNodeTarget.parentNode.parentNode,
1886                     bgColor,
1887                     isShow
1888                 )
1889                 break
1890             case 'company-info':
1891                 changeDom(
1892                     parentNodeTarget.parentNode.parentNode,
1893                     bgColor,

```

```

1894         isShow
1895     )
1896     break
1897 default:
1898     changeDom(
1899         parentNodeTarget.parentNode.parentNode,
1900         bgColor,
1901         isShow
1902     )
1903     break
1904 }
1905 } else if (curl.indexOf('liepin.com/city') !== -1) {
1906     parentNodeTarget.parentNode.parentNode.parentNode.style.border = `10px solid
1907     ${bgColor}`
1908 } else if (curl.match('https://www.liepin.com') !== null) {
1909     switch (parentNodeTargetClassName) {
1910         case 'jobcard2-company-info':
1911             changeDom(
1912                 parentNodeTarget.parentNode.parentNode,
1913                 bgColor,
1914                 isShow
1915             )
1916             break
1917         case 'home-comp-info-content':
1918             changeDom(
1919                 target.parentNode.parentNode.parentNode,
1920                 bgColor,
1921                 isShow
1922             )
1923             break
1924         default:
1925             changeDom(
1926                 parentNodeTarget.parentNode.parentNode.parentNode.parentNode,
1927                 bgColor,
1928                 isShow
1929             )
1930             break
1931     }
1932 } else {
1933     parentNodeTarget.parentNode.parentNode.parentNode.parentNode.parentNode.style.backgroundColor =
1934     bgColor
1935 }
1936 break
1937 default:
1938     console.error('操作样式出问题啦! ')
1939     isShowTip &&
1940     myToast.multipleNotControl(
1941         'error',
1942         '出错了! ',
1943         1500,
1944         false,
1945         '操作样式出问题啦! ',
1946         () => {}
1947     )
1948     break
1949 }
1950 }
1951 function changeDom(
1952     target,
1953     bgColor,
1954     isShow,
1955     msg = '此家公司在黑名单列表中! '
1956 ) {
1957     if (!isShow) {
1958         target.remove()
1959     } else {

```

```

1959     target.style.backgroundColor = bgColor
1960     target.title = msg
1961 }
1962 }
1963 return 'end'
1964 }
1965
1966 /**
1967  * 屏蔽功能的主函数
1968  * @param {Boolean} isFirstRun 是否是第一次运行
1969  * @param {Array} blackListArr 黑名单列表
1970  * @param {Array} whiteListArr 白名单列表
1971  * @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
1972  * @param {Node} listDomSelectorArr 用于判断网站的dom元素类名
1973  * @param {Number} delay 延迟获取的毫秒数
1974  */
1975 async function autoBlock_mainFn(
1976
1977     isFirstRun,
1978     blackListArr,
1979     whiteListArr,
1980     pageType,
1981     listDomSelectorArr,
1982     curSiteType,
1983     isHandleHits,
1984     delay = 1000
1985 ) {
1986     var curNetState = await judgeNetState()
1987     if (!curNetState) return false
1988     var res = []
1989     for (let i = 0; i < listDomSelectorArr.length; i++) {
1990         if (curSiteType === listDomSelectorArr[i].siteType) {
1991             const webSelector = listDomSelectorArr[i].el
1992             const isList = listDomSelectorArr[i].isList
1993             const b = await myUtils.getElement(document, webSelector, delay)
1994             debug && console.log('第一条招聘信息的dom元素节点: ', b)
1995             if (b !== null) {
1996                 const [hitsCompanyArr, allCompanyArr] =
1997                     await autoBlock_getCompanyNameArrFromPage(
1998                         myUtils.uniqueArr(blackListArr),
1999                         myUtils.uniqueArr(whiteListArr),
2000                         document.querySelectorAll(webSelector),
2001                         pageType,
2002                         curSiteType,
2003                         isList || null
2004                     )
2005                 isHandleHits &&
2006                     (await autoBlock_handleHitsCompanyArr(
2007                         hitsCompanyArr,
2008                         allCompanyArr,
2009                         isList === 2 ? isList : pageType,
2010                         curSiteType
2011                     ))
2012                 break
2013             } else {
2014                 res.push(b)
2015             }
2016         }
2017         if (i === listDomSelectorArr.length - 1 && res[res.length - 1] === null) {
2018             debug && console.log(res)
2019             GM_getValue('allSettings')['showHint_enabled'] &&
2020                 myToast.normal('warning', '暂不支持当前页面!', 3000, 'top-end')
2021
2022             console.log('暂不支持当前页面! ')
2023         }
2024     }
2025     if (

```

```

2026     isFirstRun &&
2027     res.length === listDomSelectorArr.length &&
2028     res[res.length - 1] === null
2029 ) {
2030     GM_getValue('allSettings')['showHint_enabled'] &&
2031     myToast.normal('warning', '暂不支持当前页面!', 3000, 'top-end')
2032
2033     console.log('暂不支持当前页面! ')
2034 }
2035 }
2036 class Boss {
2037     constructor(token, jobJson) {
2038         this.zp_token = token
2039         this._jobListJson_boss = jobJson
2040     }
2041     /**
2042      * 发送简历-boss-主函数
2043      * @param {String} token boss身份验证zp_token
2044      * @param {Array} jobListJson 岗位列表数组
2045      * @returns
2046      */
2047     async main(token = this.zp_token, jobListJson = this._jobListJson_boss) {
2048         debugger
2049         try {
2050             if (isTimesExhausted_bs) {
2051                 debug && console.log(errMsg_bs)
2052                 GM_getValue('allSettings')['showHint_enabled'] &&
2053                 myToast.normal('error', errMsg_bs, 5000, 'center')
2054                 return false
2055             }
2056             var newArr = jobListJson.map((item, index) => {
2057                 return Object.assign(
2058                     {},
2059                     {
2060                         jobId: item.encryptJobId,
2061                         securityId: item.securityId,
2062                         lid: item.lid,
2063                         brandName: item.brandName, // 公司名
2064                         brandIndustry: item.brandIndustry, // 公司所属行业
2065                         brandScaleName: item.brandScaleName, // 公司规模
2066                         brandStageName: item.brandStageName // 公司阶段
2067                     }
2068                 )
2069             })
2070             var curNetState = await judgeNetState()
2071             if (!curNetState) return false
2072             debug && console.log(`总共要和${newArr.length}家公司发起沟通`)
2073             var err, rs, templateText
2074             var errorTimes = 0
2075             var succesTimes = 0
2076             for (let i = 0; i < newArr.length; i++) {
2077                 var {
2078                     securityId,
2079                     jobId,
2080                     lid,
2081                     brandName,
2082                     brandIndustry,
2083                     brandScaleName,
2084                     brandStageName
2085                 } = newArr[i]
2086                 if (brandIndustry && brandScaleName && brandStageName) {
2087                     templateText = `(${brandIndustry}/${brandScaleName}/${brandStageName})`
2088                 } else if (!brandScaleName && brandStageName) {
2089                     templateText = `(${brandIndustry}/${brandStageName})`
2090                 } else if (brandScaleName && !brandStageName) {
2091                     templateText = `(${brandIndustry}/${brandScaleName})`
2092                 } else {

```



```

2093         templateText = ''
2094     }
2095     ;[err, rs] = await myUtils.awaitWrap(
2096         this.try2Contact(securityId, jobId, lid, token)
2097     )
2098     if (rs) {
2099         debug &&
2100         console.log(
2101             '[success]: ',
2102             `第${i + 1}家\n${brandName}${templateText}\n问候语: ${rs}`
2103         )
2104         // document
2105         // .querySelectorAll('.res')[0]
2106         // .insertAdjacentHTML(
2107             // 'beforeend',
2108             // `</br><span class="res" style="color:green">[success]: 第${i + 1}
2109             //     i + 1
2110             // )家</br>${brandName}${templateText}</br>问候语: ${rs}</span>`
2111             // )
2112         successTimes++
2113     }
2114     if (err) {
2115         debug &&
2116         console.log(
2117             '[error]: ',
2118             `第${i + 1}家\n${brandName}${templateText}\n${err[0]}: ${
2119                 err[1]
2120             }`
2121         )
2122         // document
2123         // .querySelectorAll('.res')[0]
2124         // .insertAdjacentHTML(
2125             // 'beforeend',
2126             // `</br><span class="res" style="color:red">[error]: 第${i + 1}
2127             //     i + 1
2128             // )家</br>${brandName}${templateText}</br>${err[0]}: ${
2129             //     err[1]
2130             // }</span>`
2131             // )
2132         errorTimes++
2133     }
2134     if (err[1] === '今日沟通人数已达上限, 请明天再试') {
2135         errMsg_bs = '今日沟通人数已达上限, 请明天再试'
2136         debug && console.log('已达上限啦, 为你退出本次脚本! 等明天再来吧')
2137         isTimesExhausted_bs = true
2138         break
2139     }
2140 }
2141 await myUtils.randomSleep(3, 6)
2142 }
2143 // document
2144 // .querySelectorAll('.res')[0]
2145 // .insertAdjacentHTML(
2146     // 'beforeend',
2147     // `</br><span class="res">--- 结束 ---</span>`
2148     // )
2149 debug &&
2150 console.log(
2151     `与当前页面的全部公司结束沟通! \n成功: ${successTimes}家; 失败: ${errorTimes}家`
2152 )
2153 } catch (error) {
2154     debug && console.error(error.message)
2155 }
2156 }
2157
2158 removeBlackList(arr, drr) {
2159     return arr.filter((item) => {

```

```

2160     var target = item['brandName']
2161     return !drf.find((element) => {
2162         var name = element['bName'] ?? element['oName']
2163         return (
2164             name.indexOf(target) !== -1 ||
2165             name.match(new RegExp(target, 'ig')) ||
2166             name === target
2167         )
2168     })
2169 })
2170 }
2171
2172 /**
2173  * 获取boss的zp_token
2174  * @returns zp_token
2175  */
2176 get_zp_token() {
2177     return new Promise((resolve, reject) => {
2178         try {
2179             axios
2180                 .get('https://www.zhipin.com/wapi/zppassport/get/zpToken')
2181                 .then((result) => {
2182                     var res = result.data
2183                     if (res.zpData) {
2184                         resolve(res.zpData.token)
2185                     } else {
2186                         reject('error?:', res)
2187                     }
2188                 })
2189             } catch (error) {
2190                 debug && console.log('error: ', error.message)
2191             }
2192         })
2193     }
2194
2195 /**
2196  * 发起沟通-boss, 因为boss不能直接发简历
2197  * @param {String} securityId boss身份验证securityId
2198  * @param {String} jobId jobid
2199  * @param {String} lid boss身份验证lid
2200  * @param {String} zp_token boss身份验证zp_token
2201  * @returns
2202  */
2203 try2Contact(securityId, jobId, lid, zp_token = this.zp_token) {
2204     return new Promise((resolve, reject) => {
2205         try {
2206             var config = {
2207                 method: 'post',
2208                 url: `https://www.zhipin.com/wapi/zpgeek/friend/add.json?
securityId=${securityId}&jobId=${jobId}&lid=${lid}`,
2209                 headers: {
2210                     zp_token: zp_token,
2211                     'User-Agent':
2212                         userAgent[parseInt(Math.random() * userAgent.length)]
2213                 }
2214             }
2215             axios(config)
2216                 .then(function (result) {
2217                     try {
2218                         var res = result.data
2219                         if (res && res.zpData && res.zpData !== {}) {
2220                             if (res.zpData.greeting) {
2221                                 resolve(res.zpData.greeting)
2222                             } else if (res.zpData.showGreeting) {
2223                                 resolve('已发起沟通! ')
2224                             } else if (res.zpData.bizData) {
2225                                 reject([

```

```

2226         res.zpData.bizData.chatRemindDialog.title,
2227         res.zpData.bizData.chatRemindDialog.content
2228     ])
2229     }
2230     } else {
2231         reject('api失效了')
2232     }
2233     } catch (error) {
2234         debug && console.log('error: ', error)
2235     }
2236 })
2237 .catch(function (error) {
2238     debug && console.log(error)
2239     reject('error: ', error)
2240 })
2241 } catch (error) {
2242     debug && console.log('error: ', error)
2243 }
2244 })
2245 }
2246 }
2247
2248 class ZhiLian {
2249     constructor(cityId) {
2250         this.cityId = cityId
2251     }
2252     /**
2253      * 发送简历-zl-提取页面中所有岗位的jobNumbers
2254      * @param {String} str 岗位招聘信息的链接
2255      * @returns jobNumbers
2256      */
2257     extract_jobNumbers(str) {
2258         var nurl = new URL(str)
2259         var reg = /\\/([^&#]+)\.htm/
2260         var query = nurl.pathname.match(reg)
2261         return query[1] || null
2262     }
2263     /**
2264      * 发送简历-zl-主函数
2265      * @param {Number} cityId 岗位所在城市的代码，其实也是所在地区的代码
2266      */
2267     async main(cityId = this.cityId) {
2268         var _self = this
2269         try {
2270             const allHrefNodeArr = document.querySelectorAll(
2271                 'a.joblist-box__iteminfo'
2272             )
2273             var allHrefArr = [].slice.call(allHrefNodeArr, 0)
2274             const jobNumbersArr = allHrefArr
2275                 ? allHrefArr.map(function (el, index) {
2276                     var txt = _self.extract_jobNumbers(el.href)
2277                     return { jobNumber: txt, index: index }
2278                 })
2279                 : []
2280             var webSelector = _config_jobListDomSelectorArr.filter((item) => {
2281                 if (item.siteType === 2) {
2282                     return true
2283                 }
2284             })
2285             for (let i = 0; i < webSelector.length; i++) {
2286                 var target = webSelector[i].el
2287                 var a = await myUtils.getElement(document, target, 1500)
2288                 if (a !== null) {
2289                     var firstRes = document.querySelectorAll(target)
2290                     if (!firstRes || firstRes === null) {
2291                         var _targetarr = target.split(' ')
2292                         debug && console.log('resume-getElementsByClassName')

```

```

2293         firstRes = document.getElementsByClassName(
2294             _targetarr[_targetarr.length - 1].replace('.', '')
2295         )
2296     }
2297     const rr = await autoBlock_getCompanyNameArrFromPage(
2298         myUtils.uniqueArr(_config_blackListArr),
2299         myUtils.uniqueArr(_config_whiteListArr),
2300         firstRes,
2301         2,
2302         3
2303     )
2304     var _jobNumbersArr = jobNumbersArr.filter((item) => {
2305         return !rr[0].find((comp) => {
2306             var target = this.extract_jobNumbers(
2307                 comp['elementNode'].parentNode.parentNode.parentNode.href
2308             )
2309             return (
2310                 target === item.jobNumber &&
2311                 comp['documentIndex'] === item.index
2312             )
2313         })
2314     })
2315     debug && console.log('_jobNumbersArr: ', _jobNumbersArr)
2316     var jobNumbers = _jobNumbersArr
2317         ? _jobNumbersArr.map(function (el) {
2318             return el.jobNumber
2319         })
2320         : []
2321 }
2322 var cityIds = Array(jobNumbers.length).fill(cityId)
2323 var curNetState = await judgeNetState()
2324 if (!curNetState) return false
2325 await myUtils.randomSleep(3, 6)
2326 var obj = {}
2327 try {
2328     obj = GM_getValue('_z1_params_details')[0]
2329     if (JSON.stringify(obj) === '{}' || obj === null) {
2330         isShowTip &&
2331         myToast.normal('error', '你还未登录!', 1500, 'center')
2332         return console.error('你还未登录! ')
2333     }
2334     const [err, rs] = await myUtils.awaitWrap(
2335         this.send(
2336             jobNumbers,
2337             cityIds,
2338             obj['_resumeNumber'],
2339             obj['_at'],
2340             obj['_rt']
2341         )
2342     )
2343     debug && console.log('rs: ', rs)
2344     debug && console.log('err: ', err)
2345     if (rs) {
2346         debug &&
2347         console.log(
2348             `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! `
2349         )
2350     }
2351 } catch (error) {
2352     debug && console.log('error: ', error, '你还未登录! ')
2353 }
2354 }
2355 } catch (error) {
2356     debug && console.log('error: ', error)
2357 }
2358 }
2359

```

```

2360  /**
2361   * 发送简历-zl
2362   * @param {Array} jobNumbers 所有岗位的jobNumbers数组
2363   * @param {Array} cityIds 岗位所在城市的数组
2364   * @param {String} resumeNumber 简历id
2365   * @param {String} at zl的身份验证at
2366   * @param {String} rt zl的身份验证rt
2367   * @returns
2368   */
2369  send(jobNumbers, cityIds = this.cityId, resumeNumber, at, rt) {
2370      return new Promise((resolve, reject) => {
2371          try {
2372              if (resumeNumber === null || !resumeNumber) {
2373                  isShowTip && myToast.normal('error', '你还未登录! ', 1500, 'center')
2374                  return reject('你还未登录! ')
2375              }
2376              debug && console.log('1.5秒后将开始自动投! ')
2377              setTimeout(() => {
2378                  var config = {
2379                      url: `https://fe-api.zhaopin.com/c/pc/alan/jobs/application`,
2380                      method: 'POST',
2381                      headers: {
2382                          'User-Agent':
2383                              userAgentList[parseInt(Math.random() * userAgentList.length)]
2384                      },
2385                      data: JSON.stringify({
2386                          jobNumbers,
2387                          cityIds,
2388                          resumeNumber: String(resumeNumber),
2389                          at: String(at),
2390                          rt: String(rt),
2391                          language: 3,
2392                          batched: true,
2393                          inviteCode: '',
2394                          ignoreIntention: 1,
2395                          ignoreBlackType: '',
2396                          deliveryChannelType: 1,
2397                          pageCode: 0,
2398                          jobSource: 'RECOMMENDATION'
2399                      })
2400                  }
2401                  axios(config)
2402                      .then((result) => {
2403                          try {
2404                              var res = result.data
2405                              !res.error ? resolve(res.message) : reject(res.message)
2406                          } catch (e) {
2407                              debug && console.error('[autoSendResume_zl_send]', e.message)
2408                          }
2409                      })
2410                      .catch(function (error) {
2411                          debug && console.log(error)
2412                      })
2413                  }, 1500)
2414              } catch (error) {
2415                  debug && console.log('error: ', error)
2416              }
2417          })
2418      }
2419  }
2420
2421  class LiePin {
2422      constructor(originJobList, resId) {
2423          this.originJobList = originJobList
2424          this.resId = resId
2425      }
2426      /**

```

```

2427 * 发送简历-lp-提取工作列表 (关于公司信息)
2428 * @param {Array} originJobList 工作列表 jobkind、jobId、jobIndex
2429 * @param {Boolean} issaveCompanyInfo 是否要保存公司信息
2430 * @returns
2431 */
2432 extract_JobList(originJobList, issaveCompanyInfo = false) {
2433   return originJobList
2434     ? originJobList.map(function (el, index) {
2435       var job = el.job
2436       var comp = el.comp
2437       var obj = {
2438         jobIndex: String(+(index % 10)),
2439         jobId: String(job.jobId),
2440         jobkind: job.jobkind !== 2 ? String(2) : String(job.jobkind),
2441         info: `第${+(index + 1)}家公司\n${job.title} [${job.dq}] ${
2442           job.salary
2443         }\n${comp.compName} | ${comp.compIndustry} ${
2444           comp.compStage ? comp.compStage : ''
2445         } ${comp.compScale}`
2446       }
2447       if (!issaveCompanyInfo) {
2448         delete obj.info
2449       }
2450       return obj
2451     })
2452   : []
2453 }
2454
2455 /**
2456 * 发送简历-lp-主函数
2457 * @param {Array} originJobList 工作列表 jobkind、jobId、jobIndex
2458 * @param {String} resId 简历id
2459 * @returns
2460 */
2461 async main(originJobList = this.originJobList, resId = this.resId) {
2462   if (isTimesExhausted_lp) {
2463     debug && console.log(errMsg_lp)
2464     GM_getValue('allSettings')['showHint_enabled'] &&
2465       myToast.normal('error', errMsg_lp, 5000, 'center')
2466     return false
2467   }
2468   if (!resId || resId === undefined || resId === null) {
2469     debug && console.log('!!!')
2470     resId = await this.get_resId()
2471     if (resId === '您的帐号信息已过期, 请退出帐号重新登录') {
2472       isShowTip && myToast.normal('error', '请重新登录!', 1500, 'center')
2473       return debug && console.error('请重新登录! ')
2474     }
2475   }
2476   var webSelector = _config_jobListDomSelectorArr.filter((item) => {
2477     if (item.siteType === 3) {
2478       return true
2479     }
2480   })
2481   for (let i = 0; i < webSelector.length; i++) {
2482     var target = webSelector[i].el
2483     var a = await myUtils.getElement(document, target, 1500)
2484     if (a !== null) {
2485       var firstRes = document.querySelectorAll(target)
2486       if (!firstRes || firstRes === null) {
2487         var _targetarr = target.split(' ')
2488         debug && console.log('resume-getElementsByClassName')
2489         firstRes = document.getElementsByClassName(
2490           _targetarr[_targetarr.length - 1].replace('.', '')
2491         )
2492       }
2493       const rr = await autoBlock_getCompanyNameArrFromPage(

```

```

2494         myUtils.uniqueArr(_config_blackListArr),
2495         myUtils.uniqueArr(_config_whiteListArr),
2496         firstRes,
2497         2,
2498         3
2499     )
2500     var _originJobList = originJobList.filter((item) => {
2501         return !rr[0].find((comp) => {
2502             return (
2503                 comp['oName'] === item.comp.compName ||
2504                 comp['bName'] === item.comp.compName ||
2505                 comp['bName'].indexOf(item.comp.compName) !== -1 ||
2506                 comp['bName'].match(new RegExp(item.comp.compName, 'ig')) ||
2507                 comp['bName'] === item.comp.compName
2508             )
2509         })
2510     })
2511     var newJobList = myUtils.sliceArrLikeBeEQLengthGroup(
2512         this.extract_JobList(_originJobList, false),
2513         10
2514     )
2515
2516     var _newJobList = myUtils.sliceArrLikeBeEQLengthGroup(
2517         this.extract_JobList(_originJobList, true),
2518         10
2519     )
2520
2521     var curNetState = await judgeNetState()
2522     if (!curNetState) return false
2523     var err, rs
2524     var errorTimes = 0
2525     var successTimes = 0
2526
2527     for (let i = 0; i < newJobList.length; i++) {
2528         ;[err, rs] = await myUtils.awaitwrap(
2529             this.send(resId, newJobList[i])
2530         )
2531         if (rs) {
2532             successTimes++
2533             debug && console.log(rs)
2534             var templateText = ''
2535             for (var j = 0; j < _newJobList[i].length; j++) {
2536                 templateText += '\n' + _newJobList[i][j].info
2537             }
2538             debug && console.log('templateText: ', templateText)
2539         }
2540         if (err) {
2541             errorTimes++
2542             if (err.code === '20010') {
2543                 errMsg_lp = err.msg
2544                 debug && console.info(errMsg_lp)
2545                 isTimesExhausted_lp = true
2546                 break
2547             } else if (err.code === '-1401') {
2548                 GM_getValue('allSettings')['showHint_enabled'] &&
2549                     myToast.normal('warning', err.msg, 3000, 'center')
2550                 debug && console.log(err.msg)
2551                 break
2552             }
2553         }
2554         await myUtils.randomSleep(3, 6)
2555     }
2556     debug &&
2557     console.log(
2558         `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`
2559     )
2560 }

```

```

2561     }
2562 }
2563
2564 /**
2565  * 发送简历-lp
2566  * @param {String} resId 简历id
2567  * @param {Array} jobInfo 工作的jobkind、jobId、jobIndex
2568  * @returns code || msg
2569  */
2570 send(resId = this.resId, jobInfo) {
2571     return new Promise((resolve, reject) => {
2572         if (jobInfo) return resolve('not send:', jobInfo)
2573         var myHeaders = new Headers()
2574         myHeaders.append('Accept', 'application/json, text/plain, */*')
2575         myHeaders.append('Accept-Language', 'zh-CN,zh;q=0.9')
2576         myHeaders.append('Connection', 'keep-alive')
2577         myHeaders.append('Content-Type', 'application/x-www-form-urlencoded')
2578         myHeaders.append('Origin', 'https://c.liepin.com')
2579         myHeaders.append('Referer', 'https://c.liepin.com/')
2580         myHeaders.append('Sec-Fetch-Dest', 'empty')
2581         myHeaders.append('Sec-Fetch-Mode', 'cors')
2582         myHeaders.append('Sec-Fetch-Site', 'same-site')
2583         myHeaders.append(
2584             'User-Agent',
2585             userAgents[parseInt(Math.random() * userAgents.length)]
2586         )
2587         myHeaders.append('X-Client-Type', 'web')
2588         myHeaders.append('X-Fscp-Fe-Version', '6f6ae4a')
2589         myHeaders.append('X-Fscp-Std-Info', '{"client_id": "40106"}')
2590         myHeaders.append(
2591             'X-Fscp-Trace-Id',
2592             '9751cb68-507c-4580-ad2c-b3a6c295aca3'
2593         )
2594         myHeaders.append('X-Fscp-Version', '1.1')
2595         myHeaders.append('X-Requested-With', 'XMLHttpRequest')
2596         var urlencoded = new URLSearchParams()
2597         urlencoded.append('resId', resId)
2598         urlencoded.append('jobInfo', JSON.stringify(jobInfo))
2599
2600         var requestOptions = {
2601             method: 'POST',
2602             headers: myHeaders,
2603             body: urlencoded,
2604             redirect: 'follow',
2605             credentials: 'include'
2606         }
2607
2608         fetch(
2609             'https://apic.liepin.com/api/com.liepin.capply.platform.apply.batch-apply',
2610             requestOptions
2611         )
2612             .then((response) => response.json())
2613             .then((result) => {
2614                 if (
2615                     result.flag === 0 &&
2616                     (result.code === '20010' || result.code === '-1401')
2617                 ) {
2618                     reject({
2619                         code: result.code,
2620                         msg: result.msg
2621                     })
2622                 } else resolve(result)
2623             })
2624             .catch((error) => reject('error', error))
2625     })
2626 }
2627

```



```

2628     /**
2629     * 获取Ip的resId
2630     * @returns resId || message
2631     */
2632     get_resId() {
2633         return new Promise((resolve, reject) => {
2634             var myHeaders = new Headers()
2635             myHeaders.append('Accept', 'application/json, text/plain, */*')
2636             myHeaders.append(
2637                 'Accept-Language',
2638                 'zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6'
2639             )
2640             myHeaders.append('Connection', 'keep-alive')
2641             myHeaders.append('Content-Length', '0')
2642             myHeaders.append('Content-Type', 'application/x-www-form-urlencoded')
2643             myHeaders.append('DNT', '1')
2644             myHeaders.append('Origin', 'https://www.liepin.com')
2645             myHeaders.append('Referer', 'https://www.liepin.com/')
2646             myHeaders.append('Sec-Fetch-Dest', 'empty')
2647             myHeaders.append('Sec-Fetch-Mode', 'cors')
2648             myHeaders.append('Sec-Fetch-Site', 'same-site')
2649             myHeaders.append(
2650                 'User-Agent',
2651                 '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'
2652             )
2653             myHeaders.append('X-Client-Type', 'web')
2654             myHeaders.append(
2655                 'X-Fscp-Bi-Stat',
2656                 '{"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="}'
2657             )
2658             myHeaders.append('X-Fscp-Fe-Version', '2e3a0e1')
2659             myHeaders.append('X-Fscp-Std-Info', '{"client_id": "40108"}')
2660             myHeaders.append(
2661                 'X-Fscp-Trace-Id',
2662                 'fb6d6c4e-8618-4ae1-a799-b81eae2e09c6'
2663             )
2664             myHeaders.append('X-Fscp-Version', '1.1')
2665             myHeaders.append('X-Requested-With', 'XMLHttpRequest')
2666
2667             var requestOptions = {
2668                 method: 'POST',
2669                 headers: myHeaders,
2670                 redirect: 'follow',
2671                 credentials: 'include'
2672             }
2673
2674             fetch(
2675                 'https://apic.liepin.com/api/com.liepin.usercx.pc.user.base-property',
2676                 requestOptions
2677             )
2678                 .then((response) => response.json())
2679                 .then((result) => {
2680                     return resolve(result.data ? result.data.resId : result.msg)
2681                 })
2682                 .catch((error) => reject('error', error))
2683             })
2684         }
2685     }
2686
2687     await firstRun(_config_blackListArr, _config_whiteListArr)
2688     // #endregion
2689
2690     function getUserAgentObj() {

```

```

2691 let browserReg = {
2692     Chrome: /Chrome/,
2693     IE: /MSIE/,
2694     Firefox: /Firefox/,
2695     Opera: /Presto/,
2696     Safari: /Version\/([\d.]+).*Safari/,
2697     360: /360SE/,
2698     QQBrowswe: /QQ/
2699 }
2700
2701 let deviceReg = {
2702     iPhone: /iPhone/,
2703     iPad: /iPad/,
2704     Android: /Android/,
2705     windows: /windows/,
2706     Mac: /Macintosh/
2707 }
2708 let userAgentStr = navigator.userAgent
2709 const userAgentObj = {
2710     browserName: '', // 浏览器名称
2711     browserVersion: '', // 浏览器版本
2712     osName: '', // 操作系统名称
2713     osVersion: '', // 操作系统版本
2714     deviceName: '' // 设备名称
2715 }
2716 console.log(userAgentStr)
2717
2718 for (let key in browserReg) {
2719     if (browserReg[key].test(userAgentStr)) {
2720         userAgentObj.browserName = key
2721         if (key === 'Chrome') {
2722             userAgentObj.browserVersion = userAgentStr
2723                 .split('Chrome/')[1]
2724                 .split(' ')[0]
2725         } else if (key === 'IE') {
2726             userAgentObj.browserVersion = userAgentStr
2727                 .split('MSIE ')[1]
2728                 .split(' ')[1]
2729         } else if (key === 'Firefox') {
2730             userAgentObj.browserVersion = userAgentStr.split('Firefox/')[1]
2731         } else if (key === 'Opera') {
2732             userAgentObj.browserVersion = userAgentStr.split('Version/')[1]
2733         } else if (key === 'Safari') {
2734             userAgentObj.browserVersion = userAgentStr
2735                 .split('Version/')[1]
2736                 .split(' ')[0]
2737         } else if (key === '360') {
2738             userAgentObj.browserVersion = ''
2739         } else if (key === 'QQBrowswe') {
2740             userAgentObj.browserVersion = userAgentStr
2741                 .split('Version/')[1]
2742                 .split(' ')[0]
2743         }
2744     }
2745 }
2746
2747 for (let key in deviceReg) {
2748     if (deviceReg[key].test(userAgentStr)) {
2749         userAgentObj.osName = key
2750         if (key === 'windows') {
2751             userAgentObj.osVersion = userAgentStr
2752                 .split('windows NT ')[1]
2753                 .split(';')[0]
2754         } else if (key === 'Mac') {
2755             userAgentObj.osVersion = userAgentStr
2756                 .split('Mac OS X ')[1]
2757                 .split(' ')[0]

```

```

2758     } else if (key === 'iPhone') {
2759         userAgentObj.osVersion = userAgentStr
2760             .split('iPhone OS ')[1]
2761             .split(' ')[0]
2762     } else if (key === 'iPad') {
2763         userAgentObj.osVersion = userAgentStr
2764             .split('iPad; CPU OS ')[1]
2765             .split(' ')[0]
2766     } else if (key === 'Android') {
2767         userAgentObj.osVersion = userAgentStr
2768             .split('Android ')[1]
2769             .split(';')[0]
2770         userAgentObj.deviceName = userAgentStr
2771             .split('(Linux; Android ')[1]
2772             .split('; ')[1]
2773             .split(' Build')[0]
2774     }
2775 }
2776 }
2777
2778 return userAgentObj
2779 }
2780
2781 function handleNewPageElements(pageType, siteType, delayMS) {
2782     const uselessDiv = document.createElement('div')
2783     uselessDiv.id = 'uselessDiv'
2784     uselessDiv.style.display = 'none'
2785     document.body.append(uselessDiv)
2786     setTimeout(() => {
2787         autoBlock_mainFn(
2788             false,
2789             myUtils.uniqueArr(_config_blackListArr),
2790             myUtils.uniqueArr(_config_whiteListArr),
2791             pageType,
2792             _config_jobListDomSelectorArr,
2793             siteType,
2794             true,
2795             delayMS
2796         )
2797         setTimeout(() => {
2798             var sss = document.getElementById('uselessDiv')
2799             if (sss !== null) {
2800                 document.body.removeChild(uselessDiv)
2801             }
2802         }, 5)
2803     }, 3000)
2804 }
2805
2806 // 定义一个函数来处理新元素
2807 function handleNewPage() {
2808     // newElements.forEach((element) => {
2809     //     // 在这里处理每个新元素
2810     //     // 例如，打印新元素的内容
2811     //     console.log('New element:', element.textContent)
2812     // })
2813     var firstRunFlag = document.getElementById('uselessDiv')
2814     if (firstRunFlag === null) {
2815         debug && console.info('检测到翻页!')
2816         isShowTip && myToast.normal('success', '检测到翻页!', 1000, 'top-end')
2817         if (GM_getValue('allSettings')['autoBlocking_enabled']) {
2818             var curPageUrlArr_jobList = myUtils.findSthFromObjArr(
2819                 location.href,
2820                 _config_jobListPageUrlArr,
2821                 'url'
2822             )
2823             var curPageUrlArr_jobDetail = myUtils.findSthFromObjArr(
2824                 location.href,

```

```

2825     _config_jobDetailPageUrlArr,
2826     'url'
2827 )
2828 var curPageUrlArr_companyDetail = myUtils.findSthFromObjArr(
2829     location.href,
2830     _config_companyDetailPageUrlArr,
2831     'url'
2832 )
2833 var pageType = null
2834 if (
2835     curPageUrlArr_jobDetail.length &&
2836     !curPageUrlArr_jobList.length &&
2837     !curPageUrlArr_companyDetail.length
2838 ) {
2839     isShowTip &&
2840     myToast.normal(
2841         'success',
2842         '检测到当前可能为岗位详情页!',
2843         1000,
2844         'top-end'
2845     )
2846     return debug && console.info('检测到当前可能为岗位详情页')
2847 } else if (
2848     curPageUrlArr_companyDetail.length &&
2849     !curPageUrlArr_jobDetail.length &&
2850     !curPageUrlArr_jobList.length
2851 ) {
2852     isShowTip &&
2853     myToast.normal(
2854         'success',
2855         '检测到当前可能为公司详情页!',
2856         1000,
2857         'top-end'
2858     )
2859     return debug && console.info('检测到当前可能为公司详情页')
2860 } else if (
2861     curPageUrlArr_jobList.length &&
2862     !curPageUrlArr_jobDetail.length &&
2863     !curPageUrlArr_companyDetail.length
2864 ) {
2865     isShowTip &&
2866     myToast.normal(
2867         'success',
2868         '检测到当前页面可能为岗位列表页!',
2869         1000,
2870         'top-end'
2871     )
2872     debug && console.info('检测到当前页面可能为岗位列表页')
2873     pageType = 2
2874     handleNewPageElements(
2875         pageType,
2876         curPageUrlArr_jobList[0].siteType,
2877         1500
2878     )
2879 } else {
2880     setTimeout(() => {
2881         var curPageUrlArr_jobList1 = myUtils.findSthFromObjArr(
2882             location.href,
2883             _config_jobListPageUrlArr,
2884             'url'
2885         )
2886         var curPageUrlArr_jobDetail1 = myUtils.findSthFromObjArr(
2887             location.href,
2888             _config_jobDetailPageUrlArr,
2889             'url'
2890         )
2891         var curPageUrlArr_companyDetail1 = myUtils.findSthFromObjArr(

```

```
2892         location.href,
2893         _config_companyDetailPageUrlArr,
2894         'url'
2895     )
2896     console.log(
2897         curPageUrlArr_jobList1,
2898         curPageUrlArr_jobDetail1,
2899         curPageUrlArr_companyDetail1
2900     )
2901     if (
2902         curPageUrlArr_jobList1.length &&
2903         curPageUrlArr_jobDetail1.length &&
2904         curPageUrlArr_companyDetail1.length
2905     ) {
2906         debug && console.info('奇了个怪! 快来给我反馈一下! ')
2907         isShowTip &&
2908             myToast.multipleNotControl(
2909                 'error',
2910                 '出错了! ',
2911                 1500,
2912                 false,
2913                 '奇了个怪! 快来给我反馈一下! ',
2914                 () => {}
2915             )
2916     } else {
2917         debug &&
2918             console.log(
2919                 '当前页面暂不支持! ',
2920                 location.origin + location.pathname
2921             )
2922         isShowTip &&
2923             myToast.normal(
2924                 'info',
2925                 `当前页面暂不支持! \n当前网页链接: ${location.origin}${location.pathname}`,
2926                 1000,
2927                 'top-end'
2928             )
2929     }
2930     }, 1500)
2931 }
2932 }
2933 if (GM_getValue('allSettings')['autoSendResume_enabled']) {
2934     console.log('success', '你已开启自动投简历! ', 3000, 'center')
2935     autoSend()
2936 }
2937 } else {
2938     firstRunFlag.remove()
2939 }
2940 }
2941 })()
```



















