反社チェックサービス　RISK EYES

仕様書

（リクエスト URL）

第 3 版

2021 年 3 月 5 日

ソーシャルワイヤー株式会社

Confidential

－ 改訂履歴 －

版	項目	主な改訂内容（概要）
第 1 版 2019 年 06 月 03 日	―	新規作成
第 2 版 2020年 03 月 01 日	目次	「6.トークン認証」目次項目追加 目次項目追加に伴い、メニューの再番
	1.	1.1. 利用条件 RISKEYES の認証方法についての記述を追加
	6.	「6.トークン認証」新規追加
第 3 版 2021 年 3 月 5 日	3-1	パラメータの追加 ・remark ・condition_id

－ 目次 －

1. はじめに

この仕様書はリクエスト URL 検索の仕様について記載します。
リクエスト URL 検索は RISKEYES の会員ページ以外から絞込み条件を設定し、専用ページにて反社チェックの結果を表示する機能です。

1.1. 利用条件

I. RISKEYES の会員認証

専用ページは、取り扱うデータの著作権都合により RISKEYES の認証が必要です。認証はフォーム認証とトークン認証の 2 通りの方法があります。

・フォーム認証：リクエスト URL 専用ページを開くとログインフォームが表示されます。ID とパスワードを直接入力しログインしてください。ID とパスワードは RISKEYES と共通です。

※認証は会員ページと共通のため、既にログインしている場合は専用ページが表示されます。

※セッションタイムアウト（12 時間）やブラウザが異なる場合は再認証が必要です。

・トークン認証：アクセストークン取得 API にて、会員認証用のアクセストークンを取得し認証します。リクエスト URL 専用ページを開くとログインフォームは表示されず、ID とパスワードの入力が不要になります。

※トークン認証については 19 ページ以降で後述します。

II. 調査案件の設定

反社チェックの調査するためには調査案件が必要になります。認証後、専用ページにアクセスすると調査案件の有効性をチェックします。調査案件は調査媒体（新聞・Web）に対してそれぞれ必要になります。案件のタイプは API タイプのみ提供可能です。

2. 使用方法

リクエスト URL 検索をするための URL を記載します。

https://www.riskeyes.jp/mypage/requestUrl

絞込み条件の設定方法は URL にクエリ文字列（URL パラメータ）として追加します。以下に、使用例を記載します。

※クエリ文字列とは、サーバーに情報を送るために URL の末尾に付け加えられる文字列のことです。「?」を URL の末尾に付け加え、続けて「パラメータ名(key)＝値(value)」の形式で追加します。複数パラメータを指定する場合はパラメータ間の区切りに「&」を使用します。

※クエリ文字列は HTTP の使用上長さの制限はありませんが、ブラウザによって異なります。制限をオーバーした場合、本機能が正しく動かなくなる等のエラーが発生する場合があります。モダンブラウザとして知られる Chrome または Firefox であれば 32000 文字程度まで処理がされますが、Internet Explorer の場合、2000 文字を超過すると処理できなくなります。

※専用ページにアクセスした際、20 秒ほど白い画面の状態になることがありますが、正常に処理を実行しております。

例１）
調査キーワードが「ソーシャル太郎」
ネガティブワードタイプが「標準」
調査媒体が「新聞」の場合

https://www.riskeyes.jp/mypage/requestUrl?keyword=ソーシャル太郎 &negative_type=0&paper_flg=1

例２）
調査キーワードが「ソーシャル太郎」
ネガティブワードタイプが「反社」
調査媒体が「新聞と Web」の場合

https://www.riskeyes.jp/mypage/requestUrl?keyword=ソーシャル太郎 &negative_type=1&paper_flg=1&web_flg

※例で挙げているパラメータは必須パラメータのみの記載になります。パラメータの詳細については後述します。

※クエリ文字列の並び順に指定はありません。

3. パラメータについて

クエリ文字列で使用するパラメータの詳細について記載します。
パラメータにエラーがある場合、バリデーションでエラー判定となり検索処理を中断します。エラーの場合は画面にエラーメッセージを表示します。

※パラメータは必ず URL エンコードをしてください。UR パラメータとして使用できない記号が含まれる場合にエラーの原因になります。

※パラメータにカンマ(,)は含まないでください。調査キーワード・絞込ワード・除外ワードにスペース区切りで文字を追加することが可能ですが、カンマが含まれていると解析時に不具合の原因となります。

3.1. リクエストパラメータの種類

パラメータ名	項目名	必須	備考
keyword	調査キーワード	○	任意文字列 ※テキスト一致による検索のため、入力ワードは見出し・本文に含まれやすい形式を推奨します。 例）人名⇒姓名を空白なしで入力 ex）松本隆司 法人名⇒法人格なし ex）株式会社 /㈱などは入力しない ※空白が含有している場合、AND 検索となります。 例）「LINK　Japan」 ⇒「LINK」と「Japan」が含まれる記事を報告
negative_type	ネガティブワードタイプ	○	デフォルト 0 ※会員ページにネガティブワードのパターンを複数設定している場合、下記リクエストに準じたネガティブワードの切り替え検索ができます。 0: 貴社標準ネガワード 1: ネガワード 1 2: ネガワード 2 3: ネガワード 3
paper_flg	新聞検索フラグ	○	固定値＝ 1 絞込み媒体を識別するためのフラグです。

			※paper_flg または Web_flg どちらか一方は必ず設定する必要があります。
web_flg	Web 検索フラグ	○	固定値＝１ 絞込み媒体を識別するためのフラグです。 ※paper_flg または Web_flg どちらか一方は必ず設定する必要があります。
address	地域名		任意文字列 複数指定する場合、空白区切りで調査キーワードに対し OR 検索します。 ※地域名が入らない記事も多いため極力使用せず、検索結果が多すぎる場合での活用を推奨します。 ※記事に表記される住所は、「東京都千代田区」「東京・千代田」「千代田区霞が関」など表記がバラバラなので、「都・道・府・県・市・区・町・村」は除外した表記で指定すると必要記事を極力取得できます。 例）東京　千代田　霞が関
limit	絞込ワード		任意文字列 複数指定する場合、空白区切りで調査キーワードに対し AND 検索します。 ※ヒットした記事数が多く、法人格つき社名で検索したい場合、調査キーワードはそのまま、絞込ワードに法人格つき社名を入力し再検索すると、法人格つき社名の検索に対し検索料が追加発生することなく結果を絞り込めます。 例）調査キーワード ⇒ソーシャルワイヤー 絞込ワード ⇒ソーシャルワイヤー株式会社

keyword_not_flg	除外ワードフラグ		1 をリクエストした場合、会員ページにある『除外ワード候補の抽出（β版）』機能を使って、調査キーワードの類似語を自動抽出します。
keyword_not	除外ワード		任意文字列 自動抽出以外に文字列をリクエスト追加できます。 複数指定する場合、空白区切りで調査キーワードに対し OR 検索します。 ※最大件数について 自動抽出も含めて最大 990 件までとします。件数オーバーの場合、先頭から区切り文字で数え、キーワード単位で切り捨てとします。優先順位はリクエストキーワード ＞ 自動抽出の順になります。
time_period_01	掲載開始期間		yyyy/mm/dd 形式
time_period_02	掲載終了期間		yyyy/mm/dd 形式
remark	備考に残す		任意文字列 文字列をリクエストすると、『備考に残す』機能にあるコメント欄に自動保存されます。
condition_id	検索条件の保存		数字（検索条件 ID） 保存済みの検索条件が反映されるようリクエスト追加できます。 ※リクエストしたい検索条件を選択し、『保存』ボタンを押すと下記の画面が表示されます。右下に検索条件 ID があります。 

3.2. 検索条件の文字数制限について

検索条件で指定できる文字列の「一つのワードの長さ」は 206 文字までの制限があります。「一つのワードの長さ」とは、パラメータで指定された文字列の文字数ではなく、検索式で指定した個々の単語の文字列の文字数になります。

たとえば、"AAA BBB"という値を指定した場合は、AAA や BBB の文字数が 206 文字までになります。※"AAA BBB"の文字数（合計の文字数）は 206 文字を超えても問題ありません。また、byte 数ではなく、あくまでも文字数での制限です。

3.3. 検索条件で指定できる文字列の制限について

- 注意が必要なワードについて
  - 漢字以外の文字（ひらがな・カタカナ・英数字・記号）は 1 文字では検索できません。
    ＜検索不可ワードの例＞

    ```
    あ
    カ
    1
    &
    あ AND 富士通
    ```

  - 全角英字は、大文字と小文字が区別されます。

- 同一とみなされる文字について
  - 半角カタカナと全角カタカナ
  - 半角数字と全角数字
  - 読点（、）とカンマ（，）
  - 句点（。）とピリオド（．）
  - 単一引用符の全角（’）と半角（ ' ）
  - 二重引用符の全角（”）と半角（ " ）
  - 長音（ー）、全角ハイフン（ - ）、半角ハイフン（ - ）、全角マイナス（−）、半角マイナス（ - ）

- キーワードとして使えない文字について
  - 丸カッコ（）
  - 「AND」「OR」「NOT」
  - 機種依存文字、環境依存文字（※4）
  - 半角カンマ ,

(※4)記事が収録される段階で、機種依存文字、環境依存文字は常用漢字に変換されます。常用漢字が存在しない場合、下駄記号（〓）を代替文字とし

ています。

4. バリデーション

リクエスト URL にアクセス後、利用条件を満たしていない場合やパラメータにエラーがある場合、バリデーションの判定が False になります。

以下では、エラーメッセージと対処方法を記載します。

4.1. エラーメッセージ一覧

メッセージ	備考
検索用の DB タイプが設定されていません。	調査案件設定時に設定する項目です。運営事務局までご連絡ください。
有効な案件が設定されていません。［エラー案件］ 例）有効な案件が設定されていません。［新聞］	調査案件設定時に設定する項目です。 ※お申込みされていない案件に対して調査媒体を指定した場合もエラーになります。両方エラーになる場合は運営事務局までご連絡ください。
ネガティブワードが設定されていません。	調査案件設定時に設定する項目です。運営事務局までご連絡ください。
キーワードが指定されていません。	必須パラメータです。
検索媒体のフラグが指定されていません。	必須パラメータです。
検索不可ワードが含まれています［エラーパラメータ名］ 例）検索不可ワードが含まれています［keyword］	(※1)
１文字の語句を含めることはできません。（漢字は除く）［エラーパラメータ名］ 例）１文字の語句を含めることはできません。（漢字は除く）［keyword］	(※1)
日付項目は yyyy/mm/dd の形式で指定してください。	(※1)
除外キーワード抽出中にエラーが発生しました。	API の通信エラーが発生した可能性があります。運営事務局までご連絡ください。

(※1)「3.パラメータについて」をご参照ください。

5. 実装例

実装方法は様々ありますが、以下では入力した内容をクエリ文字列として生成するパターンを記載します。言語は HTML と JavaScript を使用しており、JavaScript で動的に URL を生成し、モーダル（モーダル内に I フレームを使用）で開きます。

<HTML>

```
<ul class="inputs-list ">
    <li>
        <span>調査キーワード</span>
        <input type="text" name="keyword">
    </li>
    <li class="negative_type">
        <span>ネガティブワードタイプ</span>
        <input type="radio" name="negative_type" value="0" checked>
        <label for="negative_word_list_0">標準</label>
        <input type="radio" name="negative_type" value="1">
        <label for="negative_word_list_1">反社</label>
        <input type="radio" name="negative_type" value="2">
        <label for="negative_word_list_2">反社以外</label>
        <input type="radio" name="negative_type" value="3">
        <label for="negative_word_list_3">カスタム</label>
    </li>
    <li class="search_media">
        <span>調査媒体</span>
        <input type="checkbox" id="paper" name="paper_flg" checked="">
        <label for="paper">新聞</label>
        <input type="checkbox" id="web" name="web_flg" checked="">
        <label for="web">WEB ニュース</label>
    </li>
    <li>
        <span>地域</span>
        <input type="text" name="address">
    </li>
    <li>
```

```
        <span>絞り込み</span>
        <textarea name="limit"></textarea>
    </li>
    <li>
        <span>除外キーワード</span>
        <textarea name="keyword_not"></textarea>
    </li>
    <li class="keyword_not_flg">
        <span>除外キーワード抽出</span>
        <input type="checkbox" id="keyword_not_flg" name="keyword_not_flg" checked="">
        <label for="web">抽出する</label>
    </li>
    <li>
        <span>期間</span>
        <input type="text" name="time_period_01"> ～
        <input type="text" name="time_period_02">
    </li>
    <li class="action">
        <input type="button" value="モーダル" onclick="makeUrl.showDialog()" class="btn btn-info">
    </li>
</ul>
```

※ネガティブワードタイプは実際に設定している値を実装してください。

※日付項目は yyyy/mm/dd の形式にして送信する必要があります。

<Javascript>

```
<script>
    let makeUrl = {
        showDialog: function() {
            // 入力パラメータの取得
            const inputParams = makeUrl.getInputParams();

            // バリデーション
            const error = makeUrl.checkValidation(inputParams);
            if (error) {
                return false;
            }

            // URL パラメータの生成
            const urlParams = makeUrl.getUrlParams(inputParams);
            const host = (("https:" == document.location.protocol) ? "https://" : "http://");
            const url = host + "/www/riskeyes/mypage/requestUrl?" + urlParams;
            const iframeStyle =
                'min-width        : 100%; ' +
                'height           : 100%; ' +
                'padding-top      : 0; ' +
                'padding-right : 0; ' +
                'padding-bottom: 0; ' +
                'padding-left     : 0;';
            const iframe = $('<iframe frameborder="0" src="' + encodeURI(url) + '" style="' +
iframeStyle + '"></iframe>');
            const dialogOptions = {
                width : 640,
                height: 570,
                dialogClass: 'open-search-login',
                title : '',
                modal : true,
                open: function() {
                    $('.open-search-login').find('.ui-dialog-titlebar-close').blur();
                }
```

```
        };

        // モーダル表示位置の指定
        makeUrl.centeringModalSyncer();
        iframe.dialog(dialogOptions);

        $(".ui-dialog-titlebar").removeClass('ui-widget-header');
    },
    getInputParams: function() {

        const keyword           = $("input[name=keyword]").val();
        const negative_type     = $("input:radio[name=negative_word_list]:checked").val();
        const address           = $("input[name=address]").val();
        const limit             = $("textarea[name=limit]").val();
        const keyword_not       = $("textarea[name=keyword_not]").val();
        const time_period_01    = $("input[name=time_period_01]").val();
        const time_period_02    = $("input[name=time_period_02]").val();
        let keyword_not_flg     = '';
        let paper_flg           = ''
        let web_flg             = '';

        if($("input:checkbox[name=keyword_not_flg]").prop("checked")){
            keyword_not_flg = 1;
        }
        if($("input:checkbox[name=paper_flg]").prop("checked")){
            paper_flg = 1;
        }
        if($("input:checkbox[name=web_flg]").prop("checked")){
            web_flg = 1;
        }

        let parameters = [];
        parameters['keyword'] = keyword;
        parameters['negative_type'] = negative_type;
        parameters['address'] = address;
        parameters['limit'] = limit;
```

```
                parameters['keyword_not'] = keyword_not;
                parameters['time_period_01'] = time_period_01;
                parameters['time_period_02'] = time_period_02;
                parameters['keyword_not_flg'] = keyword_not_flg;
                parameters['paper_flg'] = paper_flg;
                parameters['web_flg'] = web_flg;

                return parameters;
            },
            checkValidation: function(inputParams) {
                // エラー用のセレクタ定義
                const keyword_label_selector     = $("input#keyword");
                const address_label_selector     = $("input#address");
                const limit_label_selector       = $("textarea#limit");
                const keyword_not_label_selector = $("textarea#keyword_not");

                if($('.inputs-list').find('input_err')) {
                    $('.inputs-list').find('.input_err').removeClass('input_err');
                }

                let error = false;
                // チェック処理
                const mes_keyword = checkCharacter(inputParams['keyword']);
                if( null != mes_keyword ){
                    keyword_label_selector.addClass('input_err');
                    $('#error_message_area').html("<p>" + mes_keyword + "<span
id='delte_mes'></span></p>");
                    error = true;
                }

                var mes_address = checkCharacter(inputParams['address']);
                if( null != mes_address ){
                    address_label_selector.addClass('input_err');
                    $('#error_message_area').html("<p>" + mes_address + "<span
id='delte_mes'></span></p>");
                    error = true;
```

```
                }


                var mes_limit = checkCharacter(inputParams['limit']);
                if( null != mes_limit ){
                    limit_label_selector.addClass('input_err');
                    $('#error_message_area').html("<p>" + mes_limit + "<span
id='delte_mes'></span></p>");
                    error = true;
                }


                var mes_not = checkCharacter(inputParams['keyword_not']);
                if( null != mes_not ){
                    keyword_not_label_selector.addClass('input_err');
                    $('#error_message_area').html("<p>" + mes_not + "<span
id='delte_mes'></span></p>");
                    error = true;
                }


                return error;
            },
            getUrlParams: function(inputParams) {


                let urlParams = "keyword=" + encodeURIComponent(inputParams['keyword']);
                if (inputParams['negative_type']) {
                    urlParams += "&negative_type=" + inputParams['negative_type'];
                }
                if (inputParams['address']) {
                    urlParams += "&address=" + encodeURIComponent(inputParams['address']);
                }
                if (inputParams['limit']) {
                    urlParams += "&limit=" + encodeURIComponent(inputParams['limit']);
                }
                if (inputParams['keyword_not']) {
                    urlParams += "&keyword_not=" + encodeURIComponent(inputParams['keyword_not']);
                }
                if (inputParams['keyword_not_flg']) {
```

```
                urlParams += "&keyword_not_flg=" + inputParams['keyword_not_flg'];
            }
            if (inputParams['time_period_01']) {
                urlParams += "&time_period_01=" + inputParams['time_period_01'];
            }
            if (inputParams['time_period_02']) {
                urlParams += "&time_period_02=" + inputParams['time_period_02'];
            }
            if (inputParams['paper_flg']) {
                urlParams += "&paper_flg=" + inputParams['paper_flg'];
            }
            if (inputParams['web_flg']) {
                urlParams += "&web_flg=" + inputParams['web_flg'];
            }

            return urlParams;
        },
        centeringModalSyncer: function() {
            // 画面サイズを取得
            const w = $(window).width();
            const h = $(window).height();
            // コンテンツ配置位置の計算
            const pxleft = ((w - 640)/2);
            const pxtop = ((h - 570)/2);
            // CSS を設定
            $(".open-search-login").css({"left": pxleft + "px"});
            $(".open-search-login").css({"top": pxtop + "px"});
        }
};

/* バリデーション */
function checkCharacter(str) {
    var value = new RegExp("[,]");
    if(null != str.match(value)){
        return "カンマ(,)を含めることはできません。";
    }
```

```
        return null;
    }
</script>
```

※上記コードはエラー用のセレクタ等、実装環境によっては不要箇所もあります。

6. トークン認証

6.1. 利用条件

トークン認証をするためにはアクセスキーが必要になります。アクセスキーとは会員を識別するためのユニークな文字列で、トークンを取得する際に必要なパラメータです。

運営事務局にて管理しているため、トークン認証を利用する場合はご連絡をお願いします。

6.2. 利用方法

リクエスト URL 機能の基本的な使用方法は変わりません。

リクエスト URL 専用ページの URL に対して、取得したアクセストークンをクエリ文字列（URL パラメータ）として追加することでトークン認証モードになります。

```
https://www.riskeyes.jp/mypage/requestUrl?keyword=”ソーシャル太郎”&
negative_type=0&paper_flg=1&access_token=”uMsAVS2rtKxMyDEjZU9BtYFP9+7TDfTyqo7lbi09fc4=”
```

※クエリ文字列は URL エンコードしてください。

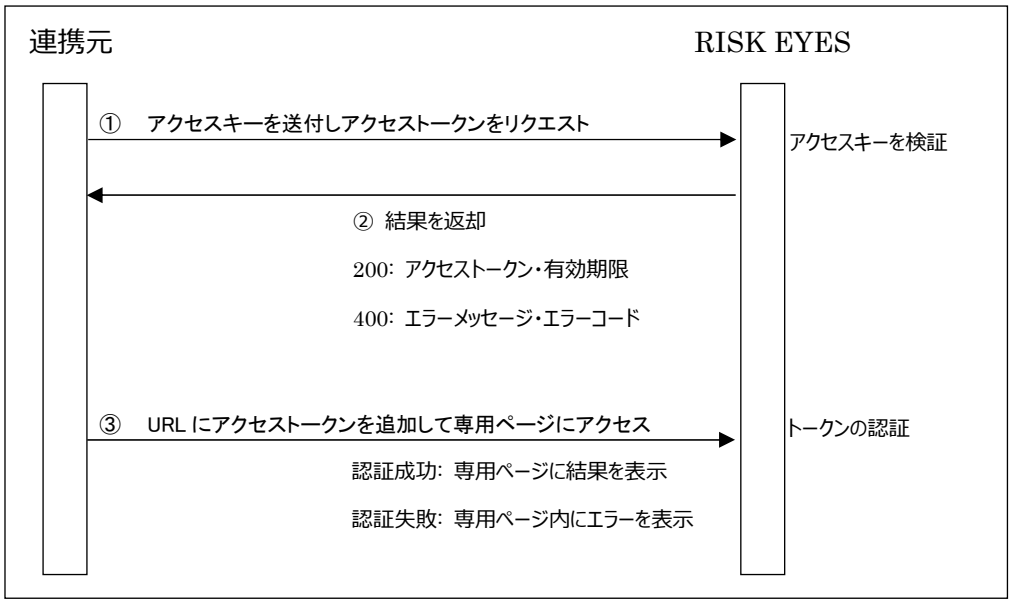
※クエリ文字列（URL パラメータ）の順番は順不同ですが、Internet Explorer の場合、2000 文字を超過すると処理できなくなるため注意してください。

6.3. アクセストークン取得方法

アクセストークン取得 API を利用して取得してください。

API の詳細については、アクセストークン取得 API 仕様書をご覧ください。

アクセストークン取得 API を使用して、専用ページにアクセスした際のフローを下図に示します。



6.4. 認証結果

リクエスト URL にアクセス後、アクセストークンの認証に失敗した場合、専用ページにエラーを表示します。
下記エラーが表示されましたら、再度トークンを取得してお試しください。解決しない場合は、事務局までお問い合わせください。

エラーの一覧

No	エラーメッセージ
1	アクセストークンが存在しないため、認証エラーが発生しました。
2	アクセストークンの有効期限が過ぎているため、認証エラーが発生しました。

6.5.　トークン認証実装例

実装方法は様々ですが、以下に一例を記載します。

<HTML>

JavaScript で URL を生成するためのボタンを用意します。

```
<input type="button" value="モーダル" onclick="makeUrl.showDialog()"
```

<Javascript>

URL 生成後、モーダル（I フレーム）でリクエスト URL 検索の画面を開きます。

```
<script>
    let makeUrl = {
        showDialog: function() {
            // アクセストークンを非同期処理にて取得
            const promise = makeUrl.getAccessToken();

            promise.done(function (token, message) {
                // エラー
                if (message !=='') {
                    console.log(message);
                    return false;
                }
                if (token !='') {
                    let inputParams = [];
                    inputParams ['keyword'] = 'ソーシャル太郎';
                    inputParams ['negative_type'] = 0;
                    inputParams ['address'] = '';
                    inputParams ['limit'] = '';
                    inputParams ['keyword_not'] ='';
                    inputParams ['time_period_01'] = '';
                    inputParams ['time_period_02'] = '';
                    inputParams ['keyword_not_flg'] = 1;
                    inputParams ['paper_flg'] = 1;
                    inputParams ['web_flg'] = 0;
                    inputParams ['access_token'] = token;
```

```
                    // URL パラメータの生成
                    const urlParams = makeUrl.getUrlParams(inputParams);
                    const url = "https://www/riskeyes/mypage/requestUrl?" + urlParams;
                    const iframeStyle =
                        'min-width      : 100%; ' +
                        'height         : 100%; ';
                    const iframe = $('<iframe frameborder="0" src="' + encodeURI(url) + '" style="' +
iframeStyle + '"></iframe>');
                    const dialogOptions = {
                        width : 640,
                        height: 570,
                        dialogClass: '',
                        title : '',
                        modal : true,
                        open: function() {
                            $('.open-search-login').find('.ui-dialog-titlebar-close').blur();
                        }
                    };

                    // モーダル表示位置の指定
                    makeUrl.centeringModalSyncer();
                    iframe.dialog(dialogOptions);
                    $(".ui-dialog-titlebar").removeClass('ui-widget-header');
                }
            });
        },
        getAccessToken: function () {
            // アクセストークン取得 API
            const accessKey = "{{ access_key }}";
            const URL = "https://www/riskeyes/api/token/access";
            const aryJSON = {"access_key": accessKey};
            const jsonData = JSON.stringify(aryJSON);
            let dfd = $.Deferred();
            $.ajax({
                url: URL,
```

```
            type: 'POST',
            contentType: 'application/json',
            data: jsonData,
            dataType: 'json',
            processData: false,
        })
        .done(function (data) {
            if (data.status == 200) {
                dfd.resolve(data.token, "");
            } else {
                dfd.resolve("", data.errors.message);
            }
        })
        .fail(function (jqXHR, textStatus, errorThrown) {
            dfd.resolve("", "System error");
        });

        return dfd.promise();
    },
    getUrlParams: function(inputParams) {
        let urlParams = "keyword=" + encodeURIComponent(inputParams['keyword']);
        if (inputParams['negative_type']) {
            urlParams += "&negative_type=" + inputParams['negative_type'];
        }
        if (inputParams['address']) {
            urlParams += "&address=" + encodeURIComponent(inputParams['address']);
        }
        if (inputParams['limit']) {
            urlParams += "&limit=" + encodeURIComponent(inputParams['limit']);
        }
        if (inputParams['keyword_not']) {
            urlParams += "&keyword_not=" + encodeURIComponent(inputParams['keyword_not']);
        }
        if (inputParams['keyword_not_flg']) {
            urlParams += "&keyword_not_flg=" + inputParams['keyword_not_flg'];
        }
```

```
                if (inputParams['time_period_01']) {
                    urlParams += "&time_period_01=" + inputParams['time_period_01'];
                }
                if (inputParams['time_period_02']) {
                    urlParams += "&time_period_02=" + inputParams['time_period_02'];
                }
                if (inputParams['paper_flg']) {
                    urlParams += "&paper_flg=" + inputParams['paper_flg'];
                }
                if (inputParams['web_flg']) {
                    urlParams += "&web_flg=" + inputParams['web_flg'];
                }
                if (inputParams['access_token']) {
                    urlParams += "&access_token=" +
encodeURIComponent(inputParams['access_token']);
                }

                return urlParams;
            },
            centeringModalSyncer: function() {
                // 画面サイズを取得
                const w = $(window).width();
                const h = $(window).height();
                // コンテンツ配置位置の計算
                const pxleft = ((w - 640)/2);
                const pxtop = ((h - 570)/2);
                // CSS を設定
                $(".open-search-login").css({"left": pxleft + "px"});
                $(".open-search-login").css({"top": pxtop + "px"});
            }
        };
    </script>
```

※上記例ではアクセスキーを PHP 側で定義し、twig 変数{{ access_key }}を使用して値を表示します。

※上記例では検索ワードを固定にしていますが、導入するシステムにあわせて実装してください。

※上記例ではバリデーションを省略しているため、導入するシステムにあわせて実装してください。

7. 留意点

7.1. 会員ページでのリアルタイム検索について

(1) 検索料について

記事がヒットした取引先に関する記事は、会員ページでリアルタイム検索を実行し閲覧が可能です。その場合、同一の調査キーワード（法人名・個人名）において、記事数表示 API もしくはリアルタイム検索の初回検索から 31 日以内であれば、記事数表示 API もしくはリアルタイム検索で再検索した場合、検索料は追加発生いたしません。

ただし、記事数表示 API もしくはリアルタイム検索の初回検索から 31 日以内で再検索した場合、初回検索時点で収録されている記事を検索します。

たとえば、同一の調査キーワードにおいて、初回検索が 12 月 1 日の場合、12 月 31 日までは 12 月 1 日時点で収録されている記事を再検索します。そのため、12 月 20 日に収録された記事は閲覧できません。
また 1 月 1 日に再検索した場合、初回検索から 32 日以上経過しているため、再度初回検索扱いとなります。検索料が課金され、1 月 1 日時点で収録されている記事を検索します。

(2) 記事数の差異について

記事数表示 API・リアルタイム検索にて初回検索・再検索した場合、同一の調査キーワードであっても記事数の異なる結果が表示される場合があります。
これは定常的に不定期で収録記事・除外ワードのメンテナンスを行っているためです。あらかじめご了承ください。

7.2. バッチ処理について

記事数表示 API を用いて、バッチ処理プログラムを組んでいただくことは可能です。その際、システム負荷の増大で検索エラーが多発し全顧客に悪影響を及ぼす可能性があるため、同会員内において、新聞・WEB 毎に記事数表示 API を複数同時に起動、並行処理することはおやめください。
（記事数表示 API-新聞を 1 本、記事数表示 API-WEB を 1 本、同時起動・並行処理することは可能です）

なお、夜間はシステム更新を頻繁に行なっているため、バッチ処理の起動可能時間は AM6：00～PM22：00 までとします。

その他、検索の留意点などはリアルタイム検索のマニュアルをご参照ください。

以上