

求人プラグイン

マニュアル

求人記事を公開するだけで、HTMLと構造化データが生成されますが、sitemap.xmlを設置する、Indexing APIを使用することでより効率的にGoogleにインデックスされます。

更新情報

- ・ 2020/02/15 BEMに乗っ取ったHTML構造に変更
- ・ 2020/02/28 ウィジェット機能追加
- ・ 2021/11/08 CSSの説明変更

目次

1. [メニュー](#)
2. [設定画面](#)
3. [投稿画面](#)
4. [求人一覧ページ](#)
5. [求人詳細ページ](#)
6. [求人を停止する](#)
7. [ショートコード・関数について](#)
8. [ウィジェット](#)
9. [Indexing APIについて](#)

メニュー

プラグインを有効にすると、管理画面左の帯に下図のようなメニューが追加されます。



設定画面

カスタム投稿タイプ【recruit】のアーカイブページ

表示しないにチェックを入れた場合、【http://localhost:8001/aim_/recruit】は404を返します。同じスラッグの固定ページがある場合、その固定ページも404になってしまうので、固定ページのスラッグを【recruitment】などに変更してください。

表示する 表示しない

設定を保存

カスタム投稿タイプ【recruit】の詳細ページ

表示しないにチェックを入れた場合、【http://localhost:8001/aim_/recruit/記事のスラッグ】は公開されません。記事詳細は関数を使うか、ショートコードを使い表示してください。

特定の記事詳細を表示する関数: `<?php echo insert_recruit_html(記事のID);?>`

特定の記事詳細を表示するショートコード: `[insert_myrecruit_page_my_recruit_id=記事のID]`

表示する 表示しない

設定を保存

カスタム投稿タイプ【recruit】のフロント側スタイル

適用しないにチェックを入れた場合、CSSはご自身で追加してください。

適用する 適用しない

設定を保存

求人公開・表示の設定を行います。

求人が一つのみで求人一覧ページが必要ない、求人詳細は任意のページにショートコード、またはphpの関数を使って埋め込むので、詳細ページは公開したくない、既存のCSSを利用しないなどのカスタマイズが可能です。

→[ショートコード、関数について](#)

投稿画面3-1

新規リクルートを追加

ここにタイトルを入力 ← 求人者のタイトル

求人情報入力欄

項目	内容
求人種別	<input checked="" type="radio"/> Googleに求人者の構造化データを送り、求人ページを追加 <input type="radio"/> Googleに求人者の構造化データを送らず、求人ページを追加 <input type="radio"/> Googleに求人者の構造化データを送り、求人ページはHTMLで追加 ← 構造化データ、HTMLを自動生成するか選択 求人ページはHTMLで追加を選んだ場合、HTMLは自動生成されません
*求人者の期限	日付を選択 ~ 日付を選択 ← 求人者の期限 Google for jobsに求人者を掲載する場合、開始日は必須入力 求人者の開始日は必須、終了日は記入推奨
*求人内容	リードテキスト ← 求人者のリードテキスト 必須項目
	タイトル付きテキスト タイトル ← 求人者のタイトル付きフリーテキスト 説明文を記入

入力エリアを削除 → クリア

入力エリアを追加 → 項目を追加

投稿画面3-2

項目を追加

勤務地	<p>= <input type="text"/> - <input type="text"/></p> <p>都道府県 <input type="text"/> 市区町村 <input type="text"/></p> <p>以降の住所 <input type="text"/></p> <p><small>上記の勤務地でGoogle mapが地図した表示にならない場合は、こちらにGoogle map用の住所を入力</small></p> <p>Google map用の住所 <input type="text"/></p>
労働時間	<p>時間を選択 <input type="text"/> ~ 時間を選択 <input type="text"/></p>
雇用形態	<p><input type="checkbox"/> フルタイム <input type="checkbox"/> パートタイム <input type="checkbox"/> 契約社員 <input type="checkbox"/> 一社採用 <input type="checkbox"/> インターン</p> <p><input type="checkbox"/> ボランティア <input type="checkbox"/> 日雇い <input type="checkbox"/> その他</p>
給与	<p>年俸 ▼ <input type="text"/> ~ <input type="text"/></p>
業務責任	<input type="text"/>
必要なスキル	<input type="text"/>
必要な免許・資格	<input type="text"/>
必要な学歴	<input type="text"/>
必要な経験	<input type="text"/>

求人者の勤務地

Google mapのHTMLも同時に生成するが、マップが意図した位置ではない場合はGoogle map用の住所を入力。
(Google map用の住所は構造化データには使われません)

求人者の勤務時間

求人者の雇用形態

複数選択可能

求人者の給与

給与のユニットは年俸、月給、週給、日給
最小のみの記入でも大丈夫です。

求人者の業務責任

求人者の必要なスキル

求人者の必要な免許・資格

求人者の必要な学歴

求人者の必要な経験

投稿画面3-3 (サイドバー)

公開

下書きとして保存 プレビュー

📍 ステータス: 下書き [編集](#)

👁️ 公開状態: 公開 [編集](#)

📅 すぐに公開する [編集](#)

公開

求人公開

アイキャッチ画像

[アイキャッチ画像を設定](#)

アイキャッチ画像

記事詳細、記事一覧で使われます

企業情報

企業ロゴ

112px x 112pxにトリミングされます

[画像を選択](#) [クリア](#)

会社名

未記入の場合サイトのタイトルを構造化データに送ります。

WEBサイト

未記入の場合サイトのURLを構造化データに送ります。

管理ID

未記入の場合投稿IDを構造化データに送ります。

お問合せ電話番号

お問合せフォームページ

企業ロゴ

会社名

WEBサイト

求人管理ID

お問合せの電話番号

問い合わせフォーム

求人一覧ページ

※有効中のテーマの中にarchive-recruit.phpがない場合、プラグインのデフォルトのアーカイブページテンプレートが使われます。デザインがテーマに合わない場合、テーマの中にarchive-recruit.phpを作成して、下記の関数でループを呼び出してください。

アーカイブテンプレートに求人一覧を表示する関数

```
<?php insert_recruit_list();?>
```

関数で表示されるHTML

```
<p>現在、{WPのタイトル}では一緒に働いてくれるスタッフの募集を行なっています。<br>募集している職種は下記の通りです。</p>
<div class= "custom-recruit__archive">
////////////////////▼ループ箇所▼////////////////////////////////////
<article class="custom-recruit__archive__item">
  <div class="custom-recruit__archive__item__img"><img src= "{アイキャッチ中サイズ}" alt="{求人のタイトル}"></div>
  <div class="custom-recruit__archive__item__body">
    <h3 class="custom-recruit__archive__item__ttl">{求人のタイトル}</h3>
    <div class= "custom-recruit__archive__item__description">{求人のリードテキスト (50文字) }</div>
    <div class="custom-recruit__archive__item__detail-btn"><a href= "{求人詳細のURL}" title="{求人のタイトル}">求人詳細</a></div>
  </div>
</article>
////////////////////▲ループ箇所▲////////////////////////////////////
</div>
////////////////////記事なし////////////////////////////////////
<p>申し訳ありません。<br>現在、{WPのタイトル}では募集中の求人はありません。</p>
```

求人詳細ページ3-1

※有効中のテーマの中にsingle-recruit.phpがない場合、プラグインのデフォルトのシングルページテンプレートが使われます。デザインがテーマに合わない場合、テーマの中にsingle-recruit.phpを作成して、下記の関数で記事詳細を呼び出してください。

シングルページテンプレートに求人HTMLを表示する関数

```
<?php echo insert_recruit_html();?>
```

関数で表示されるHTML

```
<div class="custom-recruit__detail">
  <div class="custom-recruit__detail__description">
    <div class="custom-recruit__detail__description-lead">
      <p>{求人のリードテキスト}</p>
    </div>
    <div class="custom-recruit__detail__description-list">
      <div class="custom-recruit__detail__description-list__item">
        <div class="custom-recruit__detail__description-list__title">
          <h2>{求人のタイトル付きフリーテキスト[タイトル]}</h2>
        </div>
        <div class="custom-recruit__detail__description-list__txt">
          <p>{求人のタイトル付きフリーテキスト[テキスト]}</p>
        </div>
      </div>
    </div>
  </div>
  <div class="custom-recruit__detail__datalist">
    <table class="custom-recruit__detail__datalist__table">
```

次ページに続きます

求人詳細ページ3-2

前ページの続き

```
<tbody>
  <tr>
    <th>勤務地</th><td>{求人の勤務地}<div class= "recruit_map">{求人の勤務地から生成したGoogle mapのiframe}</div></td>
  </tr>
  <tr>
    <th>勤務時間</th><td>{求人の勤務時間}</td>
  </tr>
  <tr>
    <th>雇用形態</th><td>{求人の雇用形態}</td>
  </tr>
  <tr>
    <th>給与</th><td>{求人の給与}</td>
  </tr>
  <tr>
    <th>業務責任</th><td>{求人の業務責任}</td>
  </tr>
  <tr>
    <th>必要なスキル</th><td>{求人の必要なスキル}</td>
  </tr>
  <tr>
    <th>必要な免許・資格</th><td>{求人の必要な免許・資格}</td>
  </tr>
  <tr>
    <th>必要な学歴</th><td>{求人の必要な学歴}</td>
  </tr>
  <tr>
    <th>必要な経験</th><td>{求人の必要な経験}</td>
  </tr>
</tbody>
</table>
</div>
</div>
<p>採用に関するお問合せは、<a href="{お問合せのフォーム}">フォーム</a>または<a href="tel:{お問合せの電話番号}">お電話</a>でお気軽にお問合せください。</p>
```

求人詳細ページ3-3

関数で生成される構造化データ

```
<script type="application/ld+json">
{
  "@context": "http://schema.org/", "@type": "JobPosting", "title": "{求人のタイトル} 【求人の雇用形態(区切り)】", "description":
  "<h1>■■■■{求人のタイトル}の求人内容■■■■</h1><br /><p>{求人のリードテキスト}</p><br /><h2>【{求人のタイトル付きフリーテキスト[タイトル]}</h2><p>{求人のタイトル付きフリーテキスト[テキスト]}</p><br /><h2>【業務責任}</h2><p>{求人の業務責任}</p><br /><h2>【応募に必要なスキル}</h2><p>{求人の必要なスキル}</p><br /><h2>【応募に必要な免許・資格}</h2><p>{求人の必要な免許・資格}</p><br /><h2>【応募に必要な学歴}</h2><p>{求人の必要な学歴}</p><br /><h2>【応募に必要な経験}</h2><p>{求人の必要な経験}</p><br />",
  "identifier": {"@type": "PropertyValue", "name": "{会社名}", "value": "{求人の管理ID}"},
  "hiringOrganization": {"@type": "Organization", "name": "{会社名}", "sameAs": "{WEBサイト}", "logo": "{企業ロゴ}"},
  "industry": "",
  "employmentType": "[{求人の雇用形態(区切り)}]",
  "workHours": "{求人の勤務時間}",
  "datePosted": "{求人の期限(開始日)}",
  "validThrough": "{求人の期限(終了日)}",
  "jobLocation": {"@type": "Place", "address": {"@type": "PostalAddress", "streetAddress": "{求人の勤務地(それ以降の住所)}", "addressLocality": "{求人の勤務地(市区町村)}", "addressRegion": "{求人の勤務地(都道府県)}", "postalCode": "{求人の勤務地(郵便番号)}", "addressCountry": "JA"}},
  "baseSalary": {"@type": "MonetaryAmount", "currency": "JPY", "value": {"@type": "QuantitativeValue", "minValue": "{求人の給与(最小)}", "maxValue": "{求人の給与(最大)}", "unitText": "{求人の給与(ユニット)}", "value": "{求人の給与(最小)}"}},
  "responsibilities": "{求人の業務責任}",
  "skills": "{求人の必要なスキル}",
  "qualifications": "{求人の必要な免許・資格}",
  "educationRequirements": "{求人の必要な学歴}",
  "experienceRequirements": "{求人の必要な経験}"
}
</script>
</div>
```

求人を停止する

求人停止する場合は求人記事を削除してください。
停止したことを早くGoogleにインデックスして欲しい場合はIndexing APIを使用する。

関数・ショートコードについて

このプラグインで使える関数・ショートコードの一覧です。

アーカイブテンプレートに求人一覧を表示する関数

```
<?php insert_recruit_list();?>
```

シングルページテンプレートに求人HTMLを表示する関数

```
<?php echo insert_recruit_html();?>
```

任意のテンプレートに特定の求人詳細を表示する関数

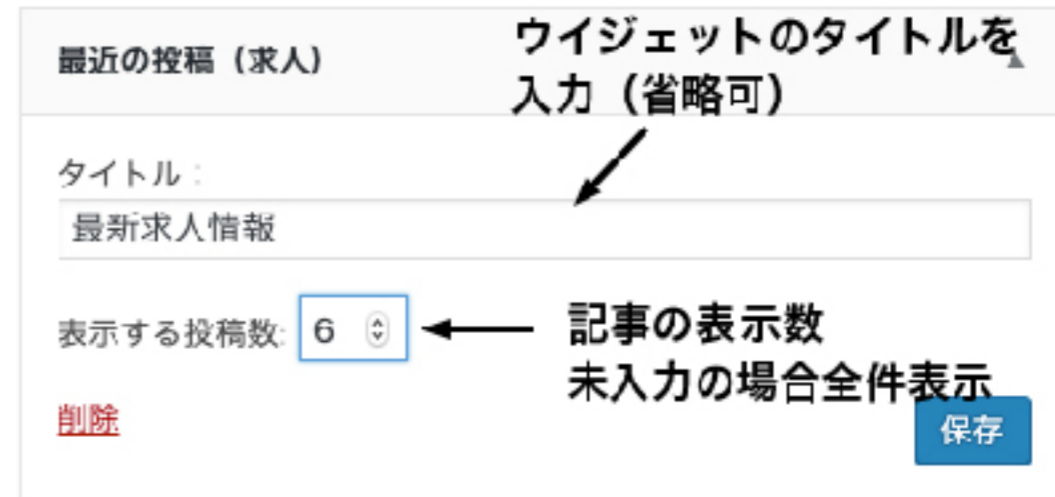
```
<?php echo insert_recruit_html(表示したい求人の投稿ID);?>
```

固定ページ、投稿ページに特定の求人詳細を表示するショートコード

```
[insert_myrecruit_page my_recruit_id=表示したい求人の投稿ID]
```

ウィジェットについて

ウィジェットの使えるテーマの場合プラグインを有効化すると「最近の投稿（求人）」が利用できます。



```
<h2>ウィジェットタイトル</h2>
<div class="widget_recent_entries">
  <ul class="custom-recruit__widget-ul">
    <li class="custom-recruit__widget-li">
      <a href="記事リンク" title="記事タイトル">記事タイトル</a>
    </li>
  </ul>
</div>
```

Indexing API

前提条件

1. Indexing API を有効にし、新しいサービス アカウントを作成
2. Search Console で所有権を確認し、アクセス トークンを取得して API 呼び出しを認証する

1-1. Indexing API を有効にし、新しいサービス アカウントを作成

[セットアップツール](#)を使用して、Google APIコンソールでプロジェクトの作成、APIの有効化、認証情報の作成を行います。

☰ Google APIs プロジェクトの選択 ▼

Google API コンソール に Indexing API のアプリケーションを登録

Google API コンソール では、アプリケーションの管理と API 使用のモニタリング
ができます。

アプリケーションを登録するプロジェクトの選択

1つのプロジェクトですべてのアプリケーションを管理できます。また、アプリケーショ
ンごとに異なるプロジェクトを作成することもできます。

プロジェクトを作成 ▼

続行

プロジェクト作成を選び続行

Indexing API

Google APIs プロジェクトの選択

API は有効になっています

プロジェクトが作成され Undefined parameter - API_NAMES が有効化されました。

次に、API を使用するための適切な認証情報が必要です。

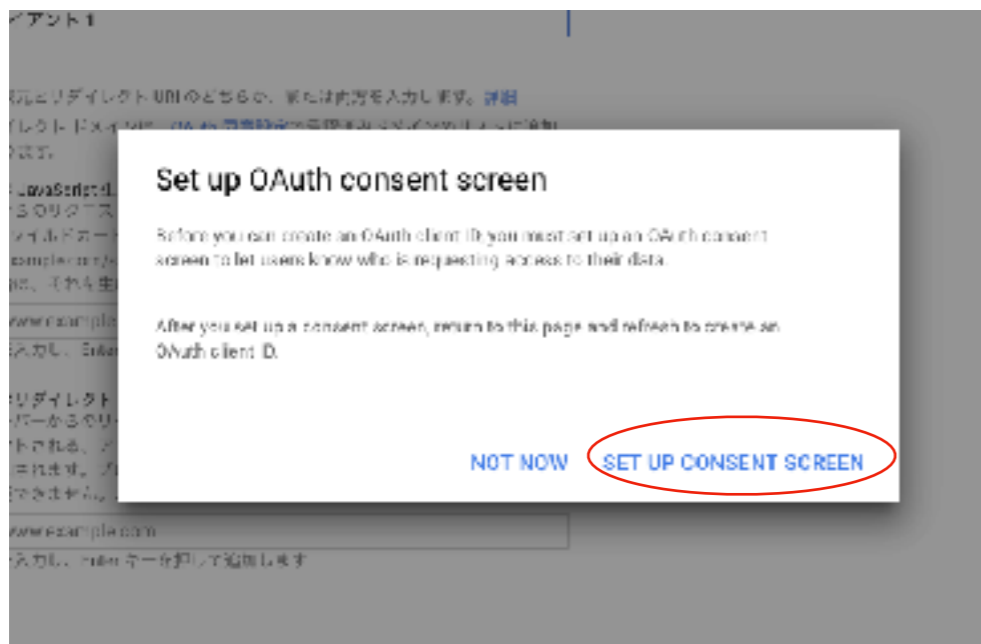
認証情報に進む

認証情報へ進む

API	API とサービス	認証情報
🏠	ダッシュボード	<h2>プロジェクトへの認証情報の追加</h2> <ol style="list-style-type: none">必要な認証情報の種類を調べる 正しい認証情報の設定をおこなってください。 この手順をスキップして、API キー、クライアント ID、サービスアカウントを作成することもできます。 <p>使用する API</p> <p>API によって使用する認証プラットフォームが異なります。また、一部の認証情報を削除して、特定の API のみを呼び出すことができます。</p> <p>Indexing API</p> <p>API を呼び出す場所</p> <p>認証情報は、その呼び出し元であるコンテキストの詳細情報を使用して判別することができます。認証情報によっては、特定のコンテキストで安全に使用できません。</p> <p>ウェブブラウザ (JavaScript)</p> <p>アクセスするデータの種別</p> <p>リクエストするデータのタイプに応じてアクセスを承認するには、各種の認証情報が必要です。</p> <p><input checked="" type="radio"/> ユーザーデータ Google ユーザーに関連するデータにアクセスします (ユーザー ID あり)</p> <p><input type="radio"/> アプリケーションデータ 自分のアプリケーションに関連するデータにアクセスします</p> <p>必要な認証情報</p> <hr/> <p>認証情報を取得する</p> <p>キャンセル</p>
📁	ライブラリ	
🔍	認証情報	

ウェブブラウザかChromeアプリを選ぶ

Indexing API



同意画面に行き情報を入力する

認証情報

プロジェクトへの認証情報の追加

- 必要な認証情報の種類を調べる
Indexing API を Chrome アプリケーション から呼び出します

2 OAuth 2.0 クライアント ID を作成する

名前

Chrome クライアント 1

アプリケーション ID

このアプリの Chrome ウェブストア URL の末尾部分

https://chrome.google.com/webstore/detail/**ewwrsrhhckhwkoyoumftpqavmzft**

OAuth クライアント ID を作成

APIを呼び出す場所でChromeアプリを選んだ場合

名前とアプリケーションIDの入力をする。

アプリケーションIDは英小文字32文字で入力。

([パスワード自動生成](#)してくれるサービスなどで作成すると楽)

Indexing API

認証情報

プロジェクトへの認証情報の追加

- ✓ 必要な認証情報の種類を調べる
Indexing API を Chrome アプリケーション から呼び出します
- ✓ OAuth 2.0 クライアント ID を作成する
OAuth クライアント「Chrome クライアント 1」を作成しました

3 認証情報をダウンロードする

Client ID

この認証情報を JSON 形式でダウンロードします。認証情報ページでいつでも行えます。

ダウンロード 後で

認証情報をローカルにダウンロードしておく

完了 キャンセル

Indexing API

1-2. 新しいサービス アカウントを作成

[サービスアカウント](#) ページへ移動する

サービス アカウント

i このページを表示するには、プロジェクトを選択してください。

選択 作成

1-1で作成したプロジェクトを選ぶ

サービス アカウント

+ サービス アカウントを作成

削除

遷移したページ上部にあるサービスアカウントを作成をクリック

Indexing API

サービス アカウントの詳細

サービス アカウント名

このサービス アカウントの表示名

サービス アカウント ID

サービス アカウントの説明

このサービス アカウントで行うことを説明します

任意のアカウント名を入力し、作成をクリック

キャンセル

サービス アカウントの権限 (オプション)

サービス アカウントに My Project へのアクセス権を付与して、プロジェクトのリソースに対して特定のアクションを完了するための権限を与えます。 [詳細](#)

役割

全リソースへの完全アクセス権

役割をオーナーにして続行をクリック

[+ 別の役割を追加](#)

キャンセル

Indexing API

サービスの作成

- サービスアカウントの詳細
- このサービスアカウントにプロジェクトへのアクセスを許可する (省略可)
- ユーザーにこのサービスアカウントへのアクセスを許可 (省略可)

ユーザーにこのサービスアカウントへのアクセス権を付与 (オプション)

Grant access to users or groups that need to perform actions on this service account.
[Learn more](#)

サービスアカウント コーポレーの役割

このサービスアカウントを使用してジョブとVMをデプロイする権限をユーザーに付与します。

サービスアカウント 管理者の役割

このサービスアカウントを管理する権限をユーザーに付与します。

キーの作成 (オプション)

秘密鍵を含むファイルをダウンロードします。この鍵を紛失すると復元できなくなるため、大切に保管してください。ただし、鍵が必要な理由が不明な場合は、とりあえずこの手順をスキップしてください。

+ キーを作成

完了 キャンセル

キーの作成 (オプション)

秘密鍵を含むファイルをダウンロードします。この鍵を紛失すると復元できなくなるため、大切に保管してください。ただし、鍵が必要な理由が不明な場合は、とりあえずこの手順をスキップしてください。

キーのタイプ

JSON
推奨

P12
P12形式を使用したコードとの互換性を目的としています

作成 キャンセル

- ・ キーを作成をクリック
- ・ キーのタイプをJSONにチェック
- ・ 作成をクリック

ダウンロードされた秘密鍵は重要な鍵ファイルなので扱いに注意

Indexing API

プロジェクト「My Project」のサービス アカウント

サービス アカウントは Google Cloud サービス ID (Compute Engine VM、App Engine アプリ、Google 以外で実行中のシステムで実行されているコードなど) を表します。 [詳細](#)

☰ 表をフィルタリング ? |||

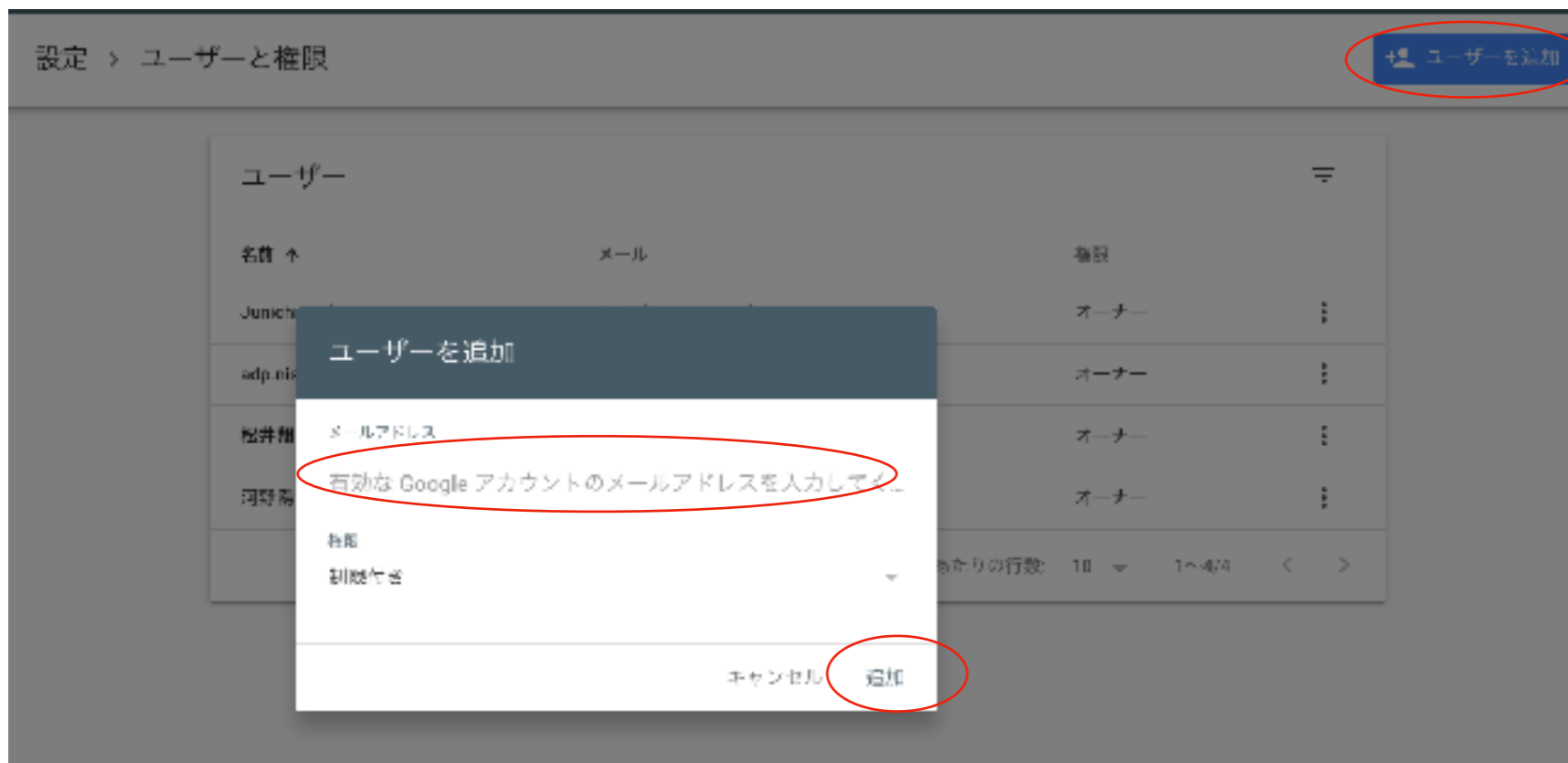
<input checked="" type="checkbox"/>	メール	名前 ↑	説明	キー ID	キーの作成日	操作
<input checked="" type="checkbox"/>	[redacted]	example		[redacted]	2019/04/05	⋮

サービスアカウントのページに戻りメールを控えておく

Indexing API

2-1. Search Console で所有権を確認

Search Consoleで求人ページの置いてあるサイトをあらかじめ登録しておく



登録が済んだらユーザーを追加から先ほどサービスアカウントページで控えたメールアドレスを入力しユーザーを追加する。

Indexing API

2-2. OAuth アクセストークンを取得、求人更新・削除を通知する

google-api-php-clientを[ダウンロード](#)して解凍し、フォルダ名をgoogle-api-php-client-2.2.2からgoogle-api-php-clientにリネームする。
google-api-php-clientをPHPが実行可能な環境（mamp、xamppなど、PHPモジュールの入ったサーバー）に設置する。
同じディレクトリに1-2で取得した鍵ファイルを置く。
アクセストークンを取得、求人を更新するプログラムと、求人を削除するプログラムを同じディレクトリに置く。

example : mampにindexingというディレクトリを作り設置した場合



<http://mampのurl/indexing/update.php> → ブラウザでアクセスすることでプログラム実行し、求人更新をGoogleに知らせる

<http://mampのurl/indexing/delete.php> → ブラウザでアクセスすることでプログラム実行し、求人削除をGoogleに知らせる

Indexing API

update.php

```
<?php

require_once './google-api-php-client/vendor/autoload.php';
$client = new Google_Client();
// service_account_file.json is the private key that you created for your service account.
$client->setAuthConfig('鍵ファイル名.json');
$client->addScope('https://www.googleapis.com/auth/indexing');
// Get a Guzzle HTTP Client
$httpclient = $client->authorize();
$endpoint = 'https://indexing.googleapis.com/v3/urlNotifications:publish';
// Define contents here. The structure of the content is described in the next step.
$content = "{
  \"url\": \"https://リクルート詳細記事のURL/\",
  \"type\": \"URL_UPDATED\"
}";
$response = $httpClient->post($endpoint, [ 'body' => $content ]);
$status_code = $response->getStatusCode();

print_r($response);
echo "\n";
echo "$status_code";

?>
```

実行結果のstatus_codeの箇所が200になっていたら成功

Indexing API

delete.php

```
<?php

require_once './google-api-php-client/vendor/autoload.php';
$client = new Google_Client();
// service_account_file.json is the private key that you created for your service account.
$client->setAuthConfig('鍵ファイル名.json');
$client->addScope('https://www.googleapis.com/auth/indexing');
// Get a Guzzle HTTP Client
$httpclient = $client->authorize();
$endpoint = 'https://indexing.googleapis.com/v3/urlNotifications:publish';
// Define contents here. The structure of the content is described in the next step.
$content = "{
  \"url\": \"https://削除されたリクルート詳細記事のURL/\",
  \"type\": \"URL_DELETED\"
}";
$response = $httpclient->post($endpoint, [ 'body' => $content ]);
$status_code = $response->getStatusCode();

print_r($response);
echo "\n";
echo "$status_code";

?>
```