

# PTRESTfulAPIClient

---

## クイックスタート

### 事前準備

ブラウザの場合は`dist/pt-restful-api-client.js`をダウンロードし、script要素で読みこんでください。

```
<script src="/path/to/pt-restful-api-client.js"></script>
```

Node環境の場合は`dist/pt-restful-api-client.js`をダウンロードし、import等で読みこんでください。

```
import PTRESTfulAPIClient from './pt-restful-api-client';
```

### 初期化

APIディレクトリ（例：`/powercmsx/api`）までのパスとAPIバージョン（現在は`1`のみ）を指定します。

```
const client = new PTRESTfulAPIClient('/powercmsx/api', 1);
```

### データの取得例

#### モデルのオブジェクト一覧を取得

例えばワークスペースID`1`の`entry`モデルの場合、以下のコードで記事一覧が取得できます。

```
const params = {
  sort_by: 'published_on',
  sort_order: 'desc',
};
const response = await client.listObjects('entry', 1, params); // 引数: モデル名, ワークスペースID, パラメーター
const json = await response.json();
// json.itemsにモデルのオブジェクト一覧が格納されている
```

※`await`を使用しているので非同期関数の中で使用します。

```
async function getList() {
  const response = await client.listObjects('entry', 1);
  const json = await response.json();
}
```

## モデルのオブジェクトを取得

例えばワークスペースID1・entryモデルにあるIDが2の記事の場合、以下のコードで取得できます。

```
const response = await client.getObject('entry', 2, 1); // 引数: モデル名, オブジェクトのID, ワークスペースID
const entry = await response.json();
// entryに記事オブジェクトが格納されている
```

## データの作成例

認証処理を行うと、オブジェクトの作成・更新・削除を行うことができます。

※ブラウザ環境の場合は認証情報が見えてしまいますので扱いには十分ご注意ください。

## オブジェクトの新規作成

例えばワークスペースID1のentryモデルの場合、以下のコードで記事が作成できます。

```
const client = new PTRESTfulAPIClient('/powercmsx/api', 1);

// 認証
const name = 'username';
const password = 'password';
const authResponse = await client.authentication(name, password);
const authData = await authResponse.json();
const token = authData.access_token;

// 投稿
const data = {
  title: 'Hello World!',
  text: '本文を入力します。',
  basename: 'hello_world',
  status: 4,
};
const response = await client.createObject('entry', 1, token, data); // 引数: モデル名, ワークスペースID, トークン, リクエストボディ
```

## メソッド一覧

[powercmsx/docs/README-RESTfulAPI.md](#)もあわせてご確認ください。

### authentication

ユーザー認証を実行します。

### 引数

- name: string ユーザー名（必須）
- password: string パスワード（必須）
- remember: boolean アクセストークンの有効期限延長

## getScheme

スキーマの取得を行います。

### 引数

- model: string モデル（必須）
- token: string アクセストークン（必須）
- workspaceId: number ワークスペースID
- keys: null | string 取得するキー

## createObject

オブジェクトを作成します。

### 引数

- model: string モデル（必須）
- workspaceId: number ワークスペースID（必須）
- token: string アクセストークン（必須）
- data: object リクエストボディ（必須）

## listObjects

オブジェクト一覧を取得します。

### 引数

- model: string モデル（必須）
- workspaceId: number ワークスペースID
- params: object パラメーター
- token: string アクセストークン

## getObject

オブジェクトを取得します。

### 引数

- model: string モデル（必須）
- id: null | string | number オブジェクトID（またはプライマリカラムの値）（必須）
- workspaceId: number ワークスペースID
- params: object パラメーター
- token: string アクセストークン

## updateObject

オブジェクトを更新します。

## 引数

- model: string モデル（必須）
- id: number オブジェクトID（必須）
- workspaceId: number ワークスペースID（必須）
- token: string アクセストークン（必須）
- data: object リクエストボディ（必須）

## deleteObject

オブジェクトを削除します。

## 引数

- model: string モデル（必須）
- id: number オブジェクトID（必須）
- workspaceId: number ワークスペースID（必須）
- token: string アクセストークン（必須）

## getContactToken

コンタクトのトークンを取得します。

## 引数

- formId: number フォームID（必須）
- workspaceId: number ワークスペースID

## confirmContact

コンタクトデータのバリデーションを実行します。

## 引数

- formId: number フォームID（必須）
- workspaceId: number ワークスペースID
- data: object リクエストボディ

## submitContact

コンタクトデータを送信します。

## 引数

- formId: number フォームID（必須）
- workspaceId: number ワークスペースID（必須）
- data: object リクエストボディ（必須）

## searchObjects

全文検索（SearchEstraiierプラグイン）を実行します。

### 引数

- model: string モデル（必須）
- workspaceId: number ワークスペースID（必須）
- params: object パラメーター（必須）

## runFetch

Fetch APIの呼び出しを行います。任意のエンドポイント（プラグインで拡張したエンドポイントなど）に対してリソース取得が可能なメソッドです。引数に応じて適切なリクエストを発行します。

### 引数

- endpoint: string エンドポイントのパス（必須）（例：/entry/list, /authorization）
- method: string リクエストメソッド（必須）
- workspaceId: number ワークスペースID（必須）
- params: object パラメーター
- data: object リクエストボディ
- token: string アクセストークン