

REST Client アダプタ の使い方

マジックソフトウェア・ジャパン株式会社



OUTPERFORM THE FUTURE™



目次: REST Clientアダプタの使い方

- 第1章 概要

- 1.1 REST Client コンポーネント

- 第2章 使用例(Google Drive API 編)

- 2.1 Google API を使用するための設定

- 2.2 REST Clientリソースの設定

- 2.3 Google Driveへファイルをアップロード

- 2.4 Google Driveのファイルに説明を追加

- 2.5 Google Driveへファイルのコピー作成

- 2.6 Google Driveのファイルをダウンロード

- 2.7 Google Driveからファイルを削除

- 第3章 使用例(kintone REST API 編)

- 3.1 kintone REST API を使用するための準備

- 3.2 REST Clientリソースの設定

- 3.3 kintoneからレコードを取得する

- 3.4 kintoneへレコードを登録する

- 3.5 kintoneのレコードを更新する

- 3.6 kintoneからレコードを削除する

- 3.7 リクエスト/レスポンスデータの扱い方



OUTPERFORM THE FUTURE™

第1章 概要



OUTPERFORM THE FUTURE™

1.1 REST Client コンポーネント

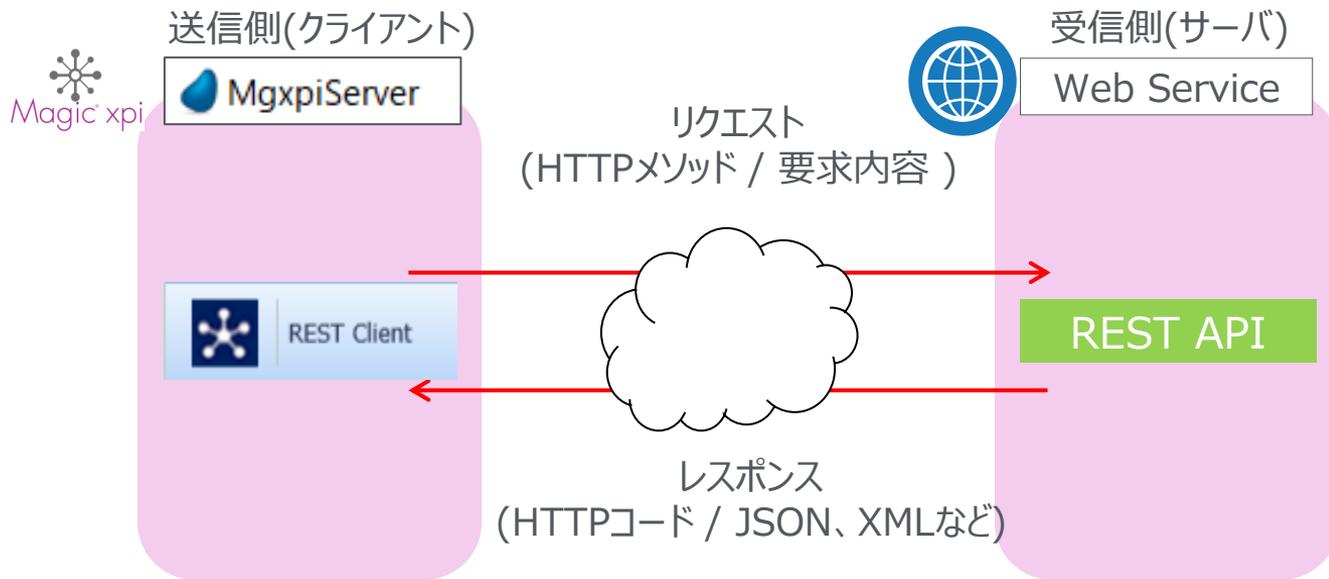


OUTPERFORM THE FUTURE™

1.1 REST Client コンポーネント

【REST Clientコンポーネント】

- ・ REST ClientコンポーネントではHTTPベースのRESTful APIを利用することができます。
- ・ HTTPメソッドでリクエストを実行し、レスポンスの結果をXML、JSONのデータ形式で取得します。



1.1 REST Client コンポーネント

【REST Client リソース】

- REST Clientコンポーネントを使用する場合、リソース設定でREST Clientエントリを作成し、APIサーバのURL、および認証プロセスに必要な情報を指定する必要があります。
- セキュリティタイプでOAuth2を選択した場合、アクセス/リフレッシュトークンを取得します。
- APIエンドポイントURLとなるパス、および各パスのパラメータを定義します。

The screenshot displays the REST Client configuration window, divided into 'リソース' (Resources) and 'パス' (Paths) sections. The 'リソース' section contains a table of resource definitions, and the 'パス' section contains a table of path definitions. A 'Parameters' dialog is also visible, showing query, header, and form parameters.

#	名	タイプ	フォーマット	値
1	サーバURL	文字	2500	https://www.googleapis.com
2	セキュリティ	文字	30	OAuth2
3	OAuth Refresh Token Status	文字	10	Available
4	認証URL	文字	1000	https://accounts.google.com/o/oauth2/auth
5	アクセストークンURL	文字	1000	https://accounts.google.com/o/oauth2/token
6	コールバックURL	文字	1000	http://localhost
7	クライアントID	文字	260	
8	クライアントシークレット	文字	260	
9	スコープ	文字	260	https://www.googleapis.com/auth/drive

#	パス名
1	/upload/drive/v3/files
2	/drive/v3/files/{field}
3	/drive/v2/files/{field}
4	/drive/v3/files
5	/drive/v2/files/{field}/copy
6	/drive/v3/files/{field}/copy
7	/drive/v3/files/{field}/export

クエリパラメータ	ヘッダパラメータ	フォームパラメータ - リクエスト	フォームパラメータ - レスポンス
#	パラメータ	デフォルト値	
1	uploadType	multipart	

APIサーバのURL、認証情報を指定

OAuth 2 認証の場合、トークン取得

APIエンドポイントURLとなるパスを指定
中括弧“{}”で囲むことで動的パスパラメータを指定可能

クエリ/ヘッダ/フォームの各パラメータを設定



OUTPERFORM THE FUTURE™

1.1 REST Client コンポーネント

【フローステップの構成】

パス	REST APIのパスを選択(リストはリソース設定から生成)
オペレーション	GET/POST/PUT/PATCH/HEAD/DELETEから選択
コンテンツタイプ	リクエスト/レスポンスの送受信されるコンテンツタイプ
エンコーディング	リクエストの一部として送信されるデータのエンコードを選択
結果保存	REST APIの結果をファイルまたは変数へ格納するかを指定
ステータスコード	リクエストによって返されるHTTPレスポンスコード
メッセージ	リクエストによって返されるHTTPレスポンスメッセージ

次頁

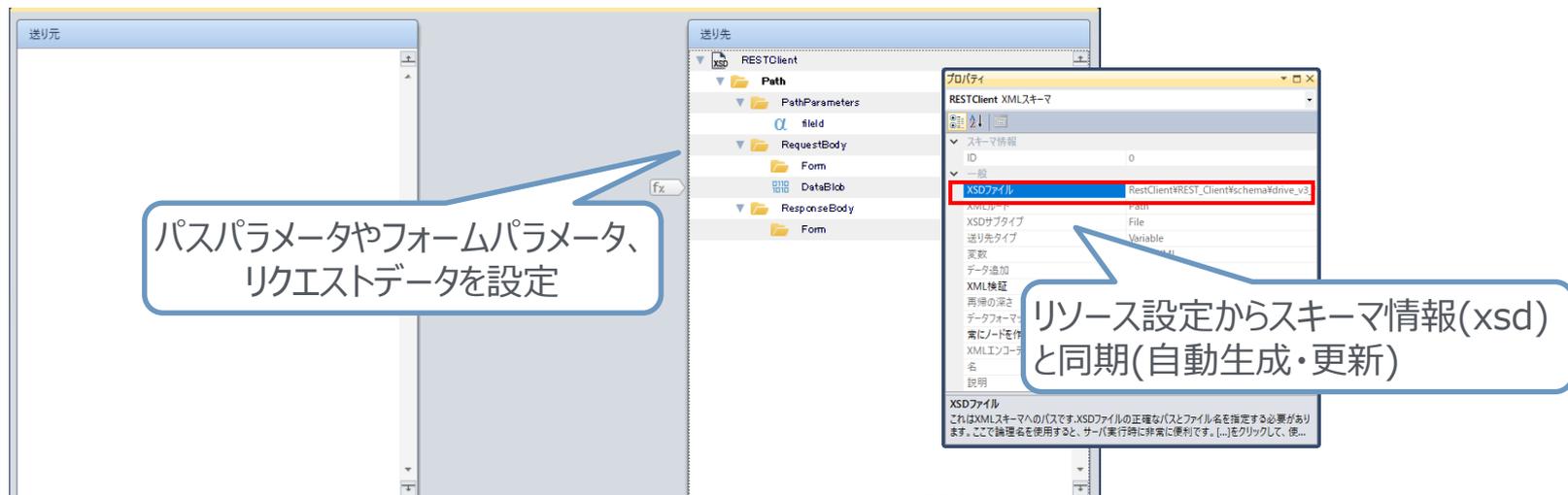
RESTクライアントリソースに設定されているクエリパラメータ、ヘッダパラメータのデフォルト値を上書きすることができます



1.1 REST Client コンポーネント

【データマッピング】

- 「RESTクライアント設定」設定ダイアログでスキーマ更新ボタンを押下するとREST ClientコンポーネントのフォームパラメータをREST Clientリソースの設定と同期します。
- 生成されたスキーマはデータマッパーのフォームパラメータを反映するために使用され、データマッパーの送り先タブで値を設定することができます。





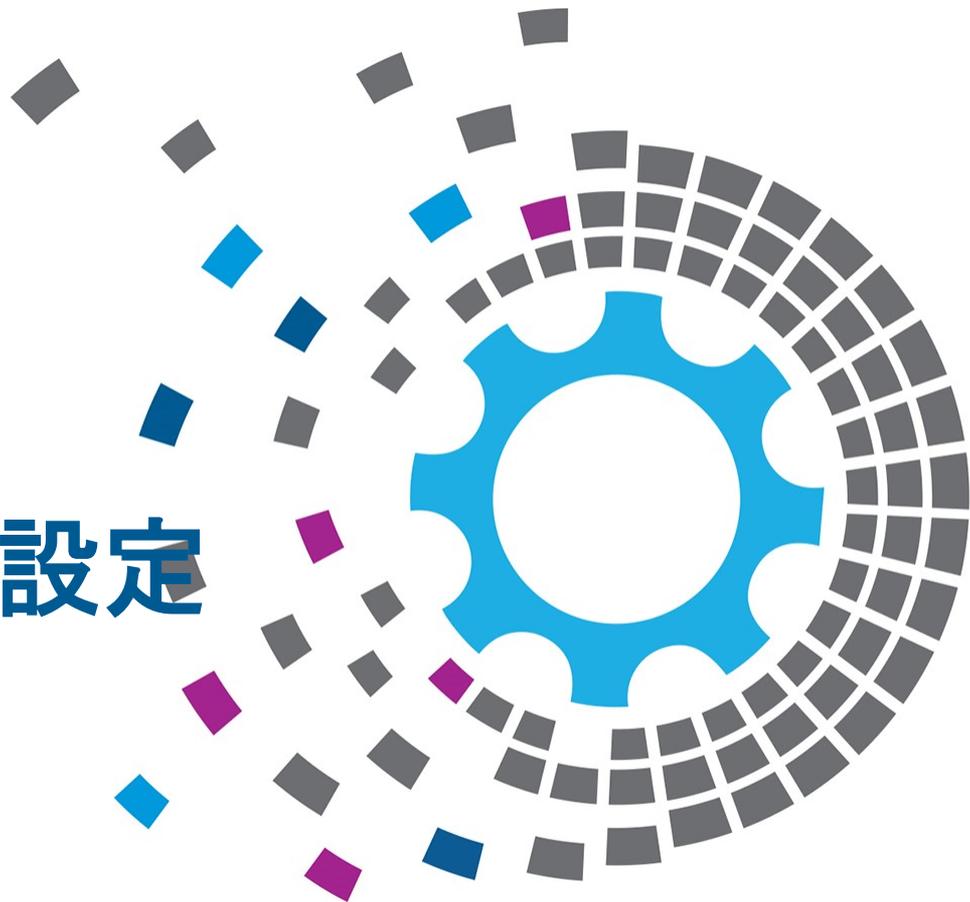
第2章 使用例 (Google Drive API 編)



OUTPERFORM THE FUTURE™

2.1

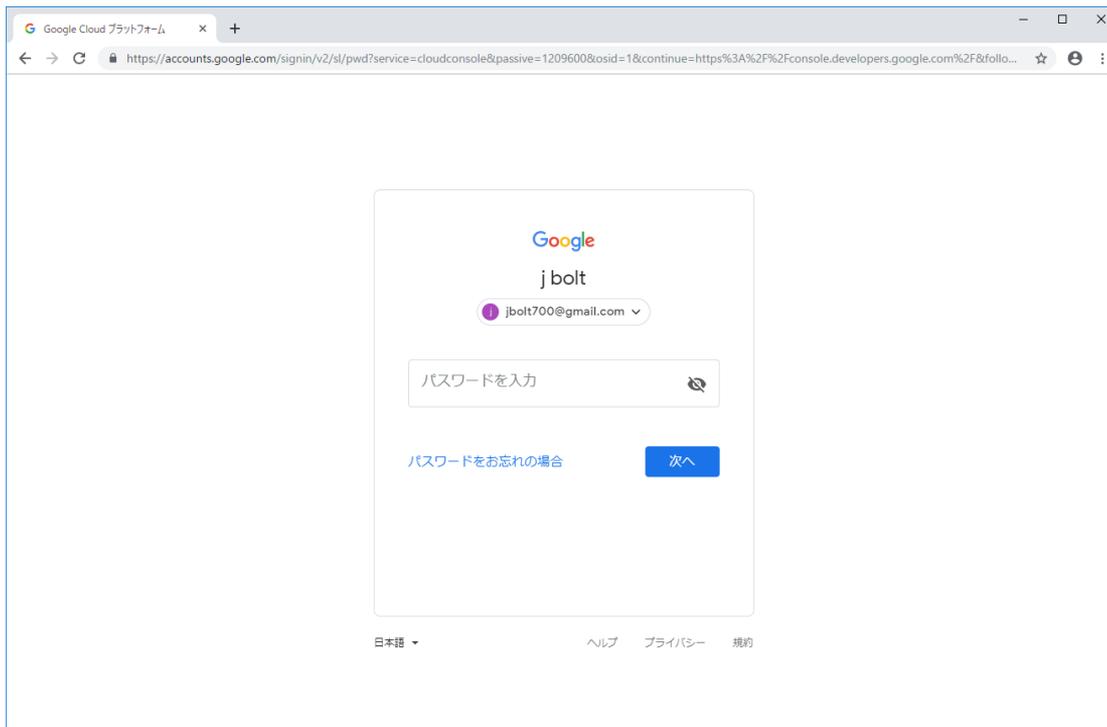
Google API を 使用するための設定



OUTPERFORM THE FUTURE™

2.1 Google API を使用するための設定

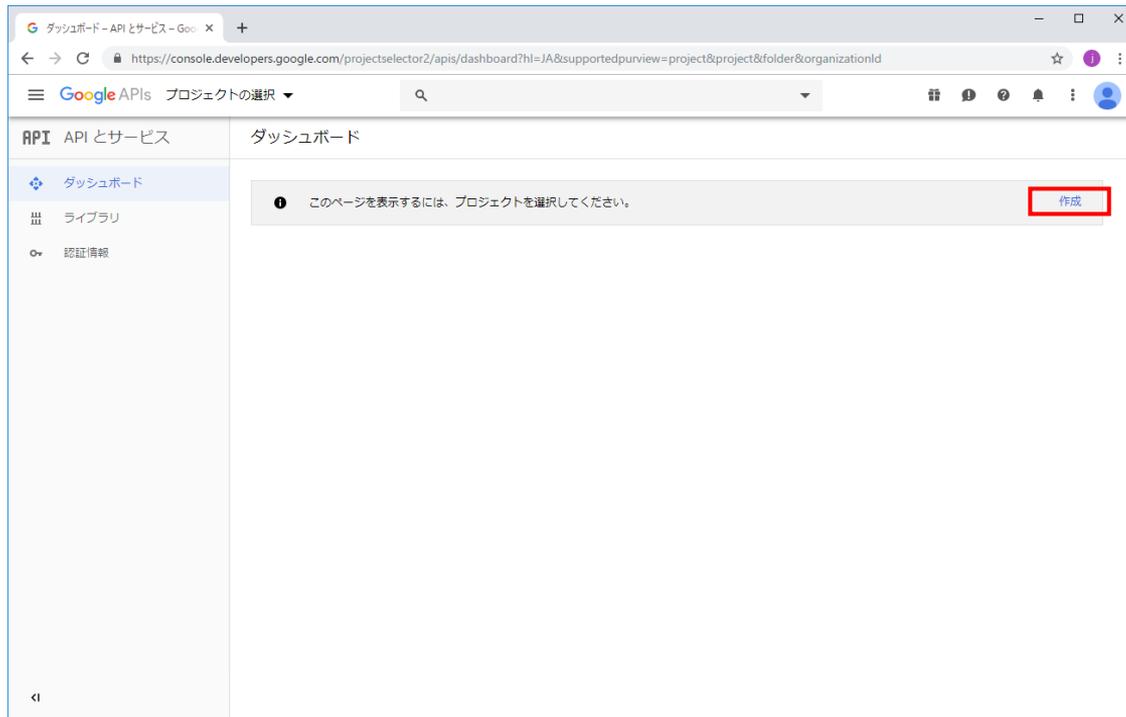
- ブラウザからGoogle Developers Console (<https://console.developers.google.com>) にアクセスし、ご利用のGoogle アカウントでログインします。



OUTPERFORM THE FUTURE™

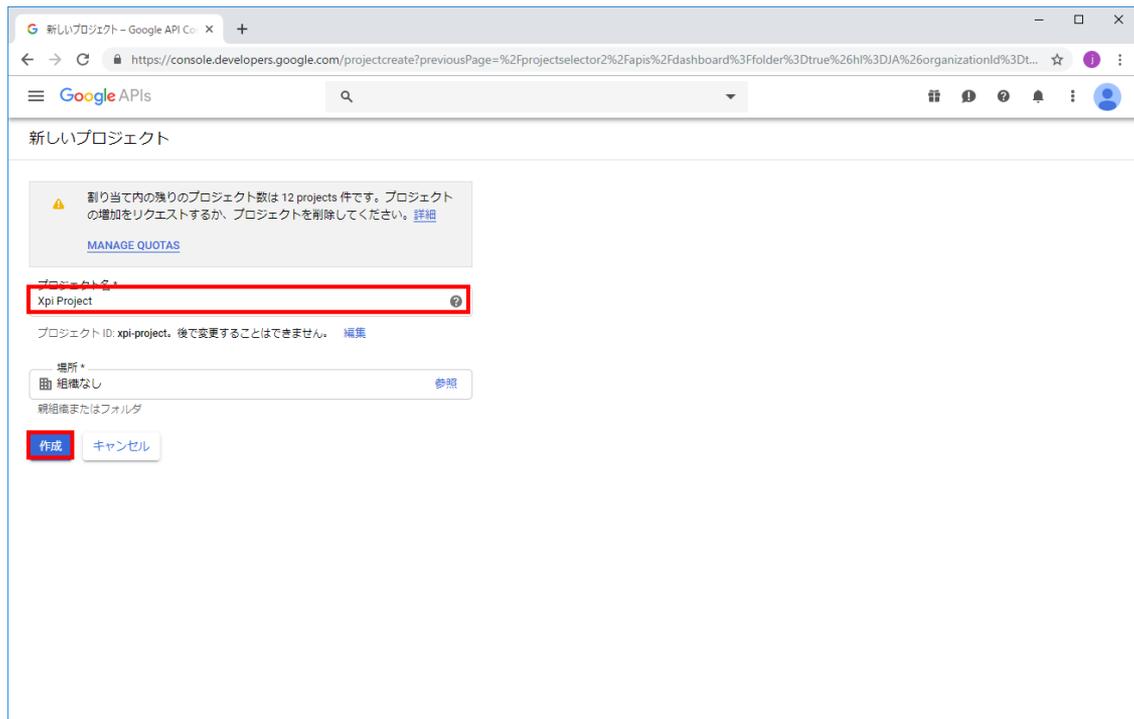
2.1 Google API を使用するための設定

- アクセスして表示された画面上に「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」が表示されます。その横の「作成」をクリックします。(または「プロジェクトの選択▼」から)



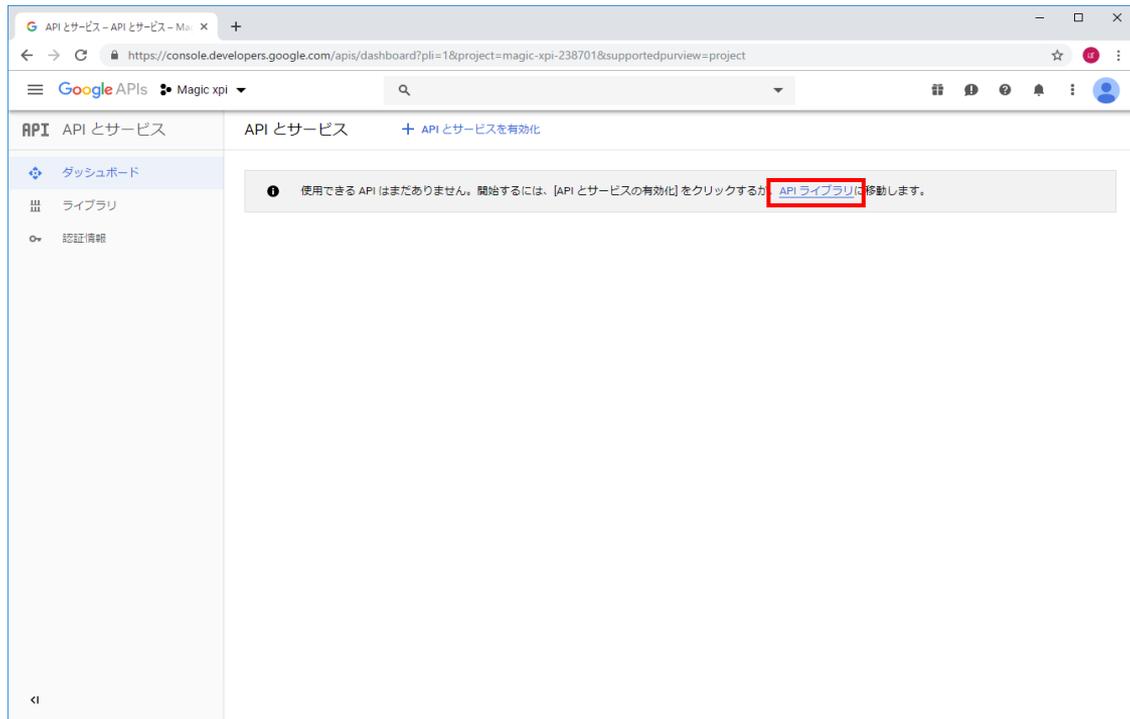
2.1 Google API を使用するための設定

- 新しいプロジェクトの作成画面で「プロジェクト名」に任意の名前を入力し、「作成」ボタンを押します。



2.1 Google API を使用するための設定

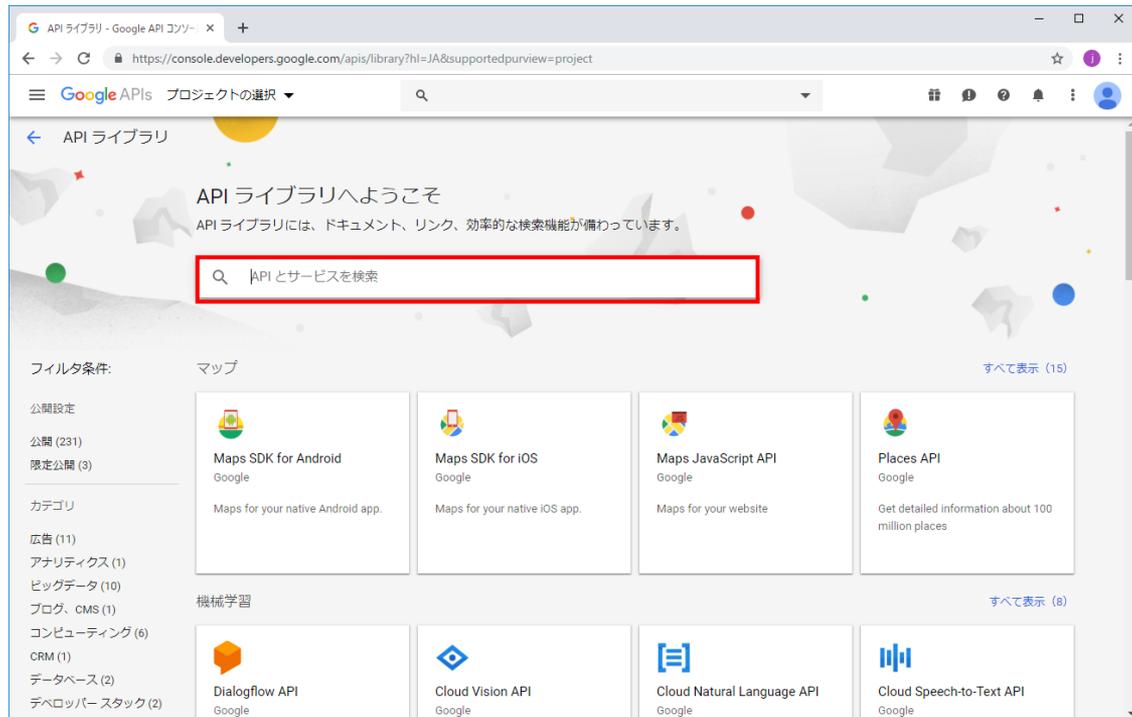
- 画面に表示されたリンク「APIライブラリ」をクリックします。
(左メニューから「ライブラリ」、または上段の「APIとサービスの有効化」でも可)



OUTPERFORM THE FUTURE™

2.1 Google API を使用するための設定

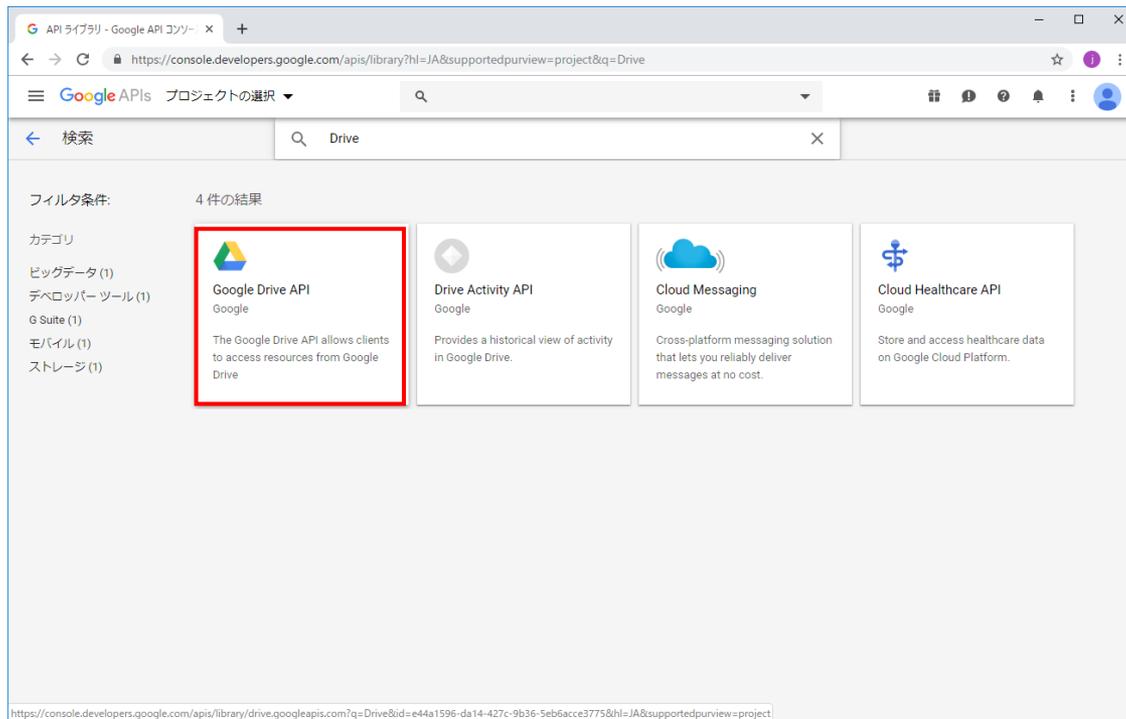
- APIライブラリの検索ボックスへ「Drive」と入力します。



OUTPERFORM THE FUTURE™

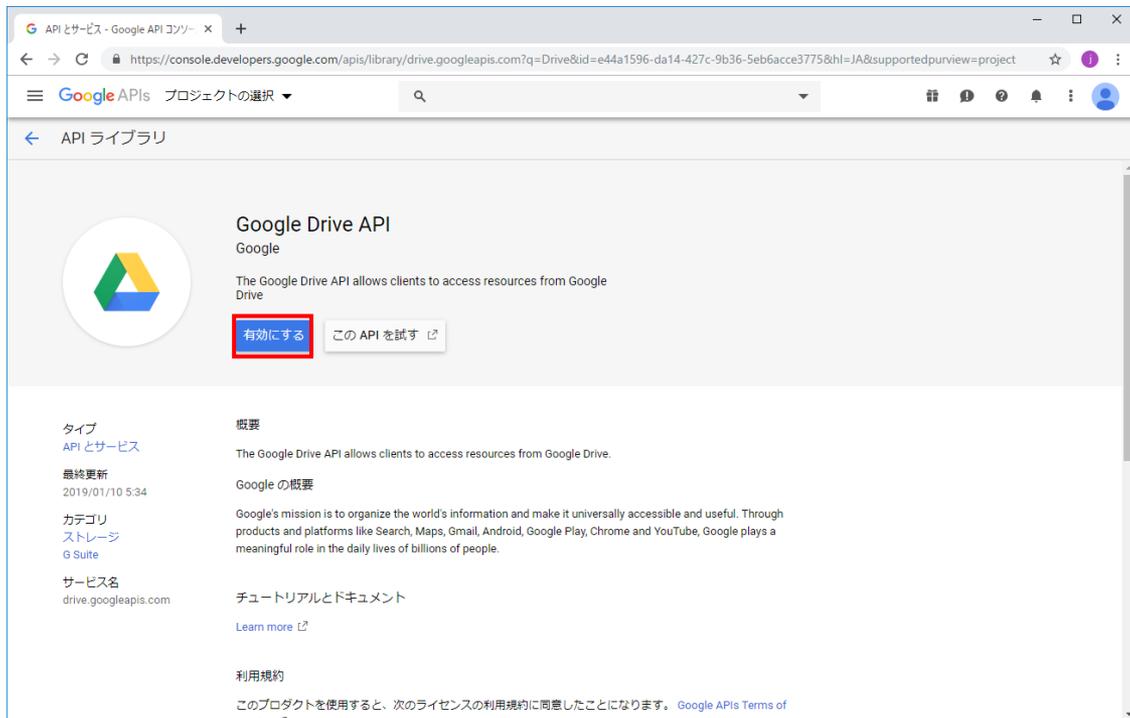
2.1 Google API を使用するための設定

- 検索結果から「Google Drive API」をクリックします。



2.1 Google API を使用するための設定

- 画面中央の「有効にする」ボタンを押します。



OUTPERFORM THE FUTURE™

2.1 Google API を使用するための設定

- 次に認証情報を作成します。画面の左メニューから「認証情報」をクリックします。
(右上の「認証情報を作成」でも可)

The screenshot shows the Google Cloud Console interface for the Google Drive API. The left-hand navigation menu has the '認証情報' (Credentials) option highlighted with a red rectangular box. At the top of the main content area, a message states: 'このAPIを使用するには、認証情報が必要になる可能性があります。開始するには、[認証情報を作成]をクリックしてください。' (To use this API, you may need authentication information. To get started, click [Create Credentials]). A button labeled '認証情報を作成' is visible next to this message. Below the message, there are sections for '詳細' (Details) and 'チュートリアルとドキュメント' (Tutorials and Documentation). The '詳細' section lists the API name (Google Drive API), provider (Google), service name (drive.googleapis.com), and status (有効). The 'チュートリアルとドキュメント' section includes links for 'Learn more' and 'API Explorer で試す'. On the right side, there is a chart titled 'レスポンスコード別のトラフィック' (Traffic by Response Code) showing '要求数/秒 (2 時間の平均)' (Requests per second (2-hour average)). The chart area is empty, with a warning icon and the text '選択した時間帯で使用できるデータがありません。' (No data available for the selected time period). The x-axis of the chart shows dates: 3月 24, 3月 31, 4月 07, and 4月 14. A button labeled '指標を表示' (Show metrics) is located at the bottom of the chart area.



2.1 Google API を使用するための設定

- 「認証情報を作成」ボタンを押し、リストから「OAuthクライアントID」をクリックします。

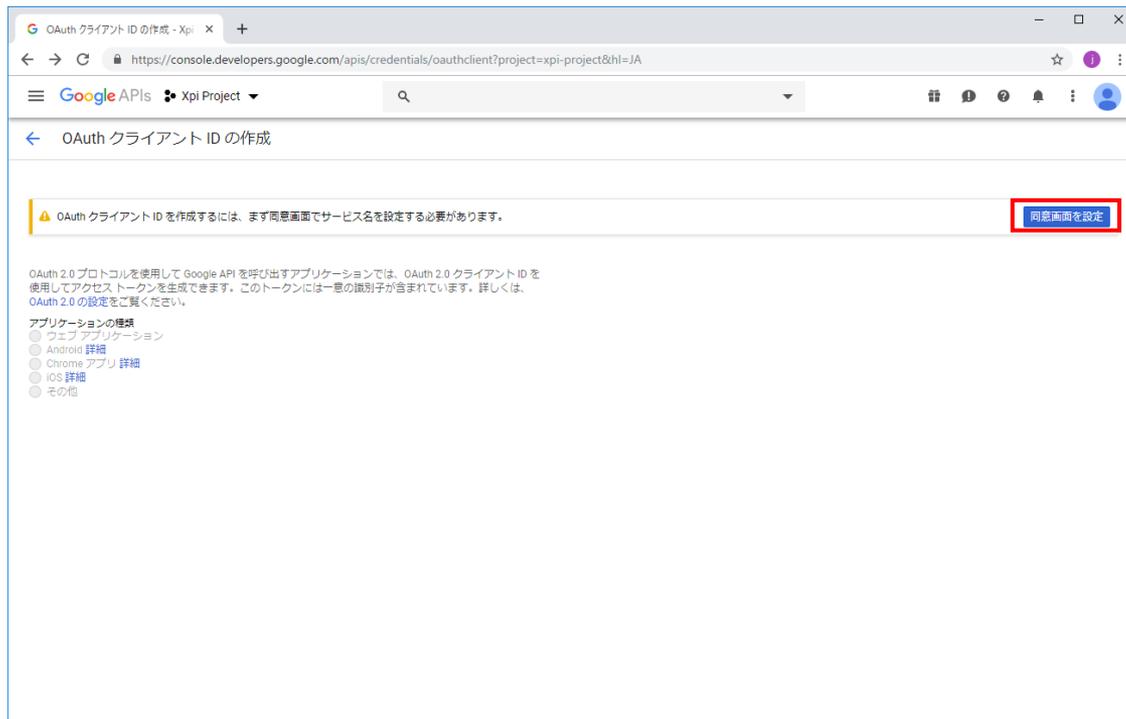


The screenshot shows the Google API console interface. The left sidebar contains navigation links: 'ダッシュボード', 'ライブラリ', and '認証情報'. The main content area is titled 'API 認証情報' and includes sub-links for 'OAuth 同意画面' and 'ドメインの確認'. A modal dialog is displayed in the center, titled 'API 認証情報', with the following text: 'API にアクセスするには認証情報が必要です。使用する API を有効にしてから、必要な認証情報を作成します。API に応じて、API キー、サービスアカウント、または OAuth 2.0 クラウド ID が必要です。詳細については、認証のドキュメントをご覧ください。' Below this text is a dropdown menu labeled '認証情報を作成' with the following options: 'API キー' (Simple API key), 'OAuth クラウド ID' (highlighted with a red box), 'サービスアカウントキー', and 'ウィザードで選択'.



2.1 Google API を使用するための設定

- はじめてクライアントIDを作成する場合、同意画面の設定が求められます。右上の「同意画面を設定」ボタンを押します。



2.1 Google API を使用するための設定

- OAuth同意画面で必須となる「アプリケーション名」と「サポートメール」を入力します。

The screenshot shows the Google API console's OAuth consent screen configuration page. The browser address bar shows the URL: `https://console.developers.google.com/apis/credentials/consent?createClient&project=xpi-project&hl=JA`. The page title is "同意画面 - Xpi Project - Google".

The left sidebar shows the navigation menu with "API とサービス" selected, and "認証情報" highlighted. The main content area is titled "認証情報" and has sub-tabs for "認証情報", "OAuth 同意画面", and "ドメインの確認".

The "OAuth 同意画面" tab is active, showing the following fields:

- 確認ステータス:** 非公開
- アプリケーション名:** Magic xpi (highlighted with a red box)
- アプリケーションのロゴ:** (empty)
- サポートメール:** jboh700@gmail.com (highlighted with a red box)

The right side of the page contains explanatory text:

同意画面について
同意画面は、誰がどの種類のデータへのアクセスをリクエストしているのかを表明する画面です。

OAuth の確認
デベロッパーとユーザーを保護するために、同意画面とアプリケーションを Google で確認する必要があります。アプリが [公開] になっていて、次のうち少なくとも 1 つが該当する場合、確認が必要です。

- プライベートにかかわるスコープや制限されたスコープがアプリで使用されています
- アプリの OAuth 同意画面にアイコンが表示されます
- アプリに多数の承認済みドメインがあります
- 以前に検証された OAuth 同意画面を変更しました

この確認には数週間かかる場合があります。進捗に関する最新情報はメールで通知されます。詳しくは、[確認プロセスの詳細](#)をご覧ください。

同意画面とアプリケーションを Google が確認する前でも、制限付きでアプリケーションをテストできます。確認前のアプリの動作の詳細は、[こちら](#)をご覧ください。

Google の OAuth に関するご意見やご要望があればお寄せください。



2.1 Google API を使用するための設定

- アクセスレベルを付与するため、「スコープを追加」ボタンを押します。



The screenshot shows the Google API console interface for an Xpi Project. The left sidebar contains a menu with 'API とサービス' and '認証情報'. The main content area is titled '認証情報' and displays the 'Scopes' section. A red box highlights the 'スコープを追加' button. Below this button, there are sections for '承認済みドメイン', 'アプリケーション ホームページ', 'アプリケーション プライバシーポリシー', and 'アプリケーション 利用規約', each with a text input field and a 'ドメインを入力し、Enter キーを押して追加します' instruction. At the bottom, there are '保存', '確認のため送信', and 'キャンセル' buttons.



2.1 Google API を使用するための設定

- 表示されたスコープ一覧から Google Drive API で必要なスコープを確認し、チェックします。
(参照 : Drive API のOAuthスコープ (<https://developers.google.com/drive/api/v3/about-auth>))

スコープの追加

スコープは、エンドユーザーに代わって、さまざまなアクセスレベルをアプリケーションに付与する場合に使用します。 OAuth 2.0 の詳細

一覧表示されるのは、有効な API のスコープのみです。

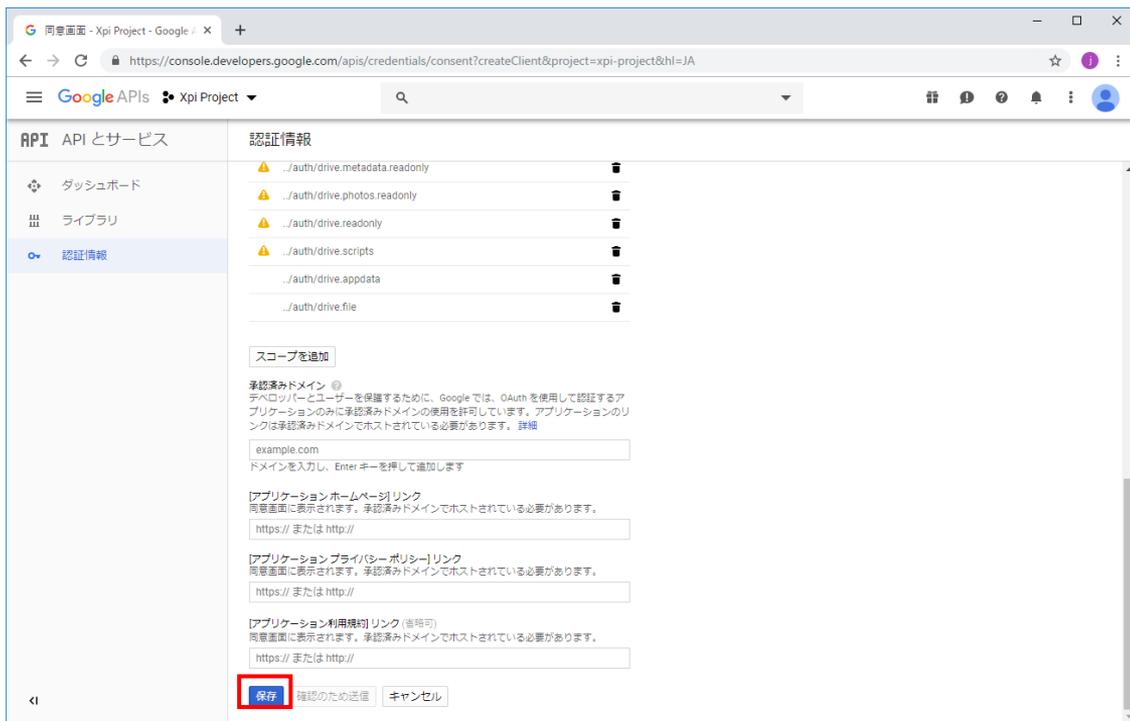
API	Scope	Description
<input type="checkbox"/>	email	メールアドレスの表示
<input type="checkbox"/>	profile	ユーザーの個人情報の表示 (ユーザーが一般公開しているすべての個人情報を含む)
<input type="checkbox"/>	openid	Google に公開されているお客様の個人情報とお客様を関連付ける
<input checked="" type="checkbox"/>	Google Drive API ./auth/drive	Google ドライブのすべてのファイルの表示、編集、作成、削除
<input checked="" type="checkbox"/>	Google Drive API ./auth/drive.appdata	Google ドライブでの独自の設定データの表示と管理
<input checked="" type="checkbox"/>	Google Drive API ./auth/drive.file	このアプリで開いたり作成したりした、Google ドライブのファイルやフォルダの表示と管理
<input checked="" type="checkbox"/>	Google Drive API ./auth/drive.metadata	Google ドライブ内のファイルのメタデータの表示と管理
<input checked="" type="checkbox"/>	Google Drive API ./auth/drive.metadata.readonly	Google ドライブ ファイルのメタデータの表示
<input checked="" type="checkbox"/>	Google Drive API ./auth/drive.photos.readonly	Google フォトの写真、動画、アルバムを表示

① スコープが見つからない場合: 一覧表示されるのは有効な API のスコープだけです。表示されていないスコープを追加するには、Google API ライブラリにアクセスし、使用する API を有効にしてからここに戻り、そのスコープを追加するか、スコープを手動で貼り付けてください。



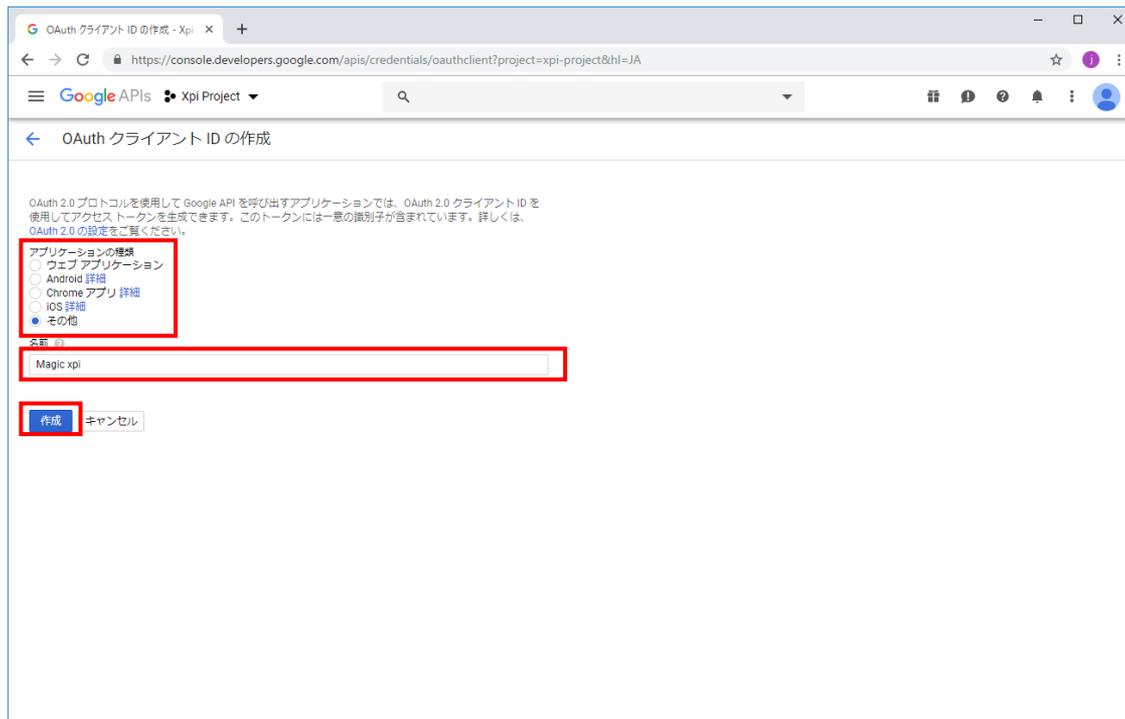
2.1 Google API を使用するための設定

- 「保存」ボタンを押して、OAuth同意画面を終了します。
(保存後にクライアントID作成画面に戻ります)



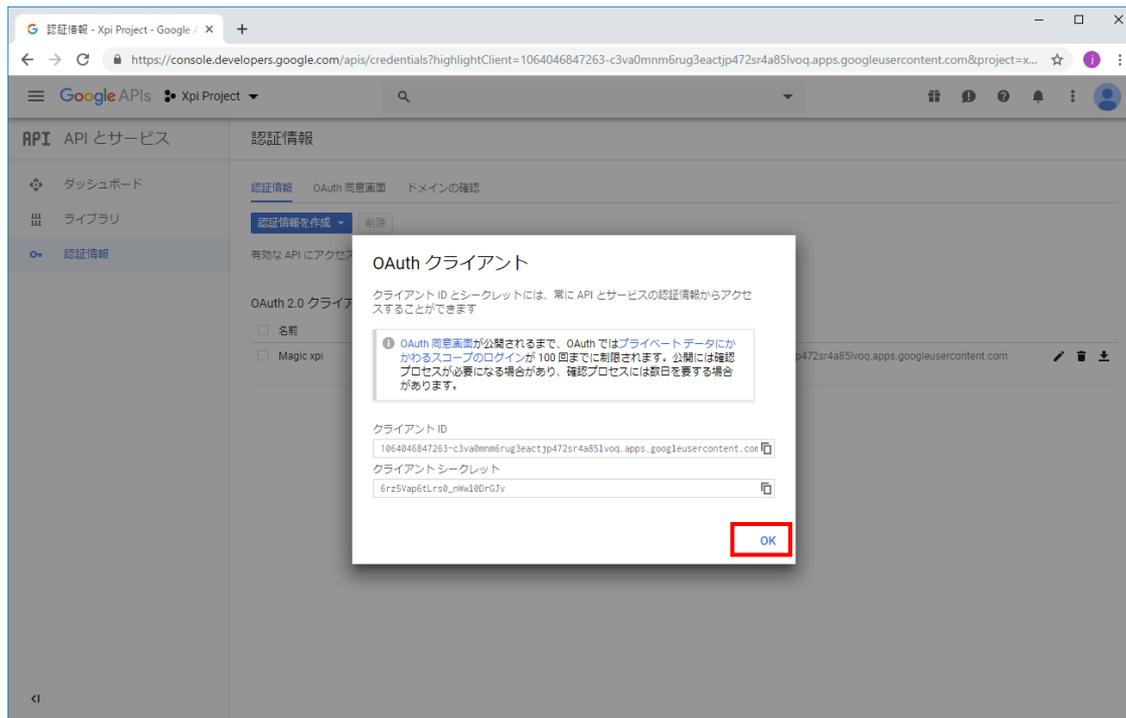
2.1 Google API を使用するための設定

- OAuthクライアントID作成画面で「アプリケーションの種類」で”その他”を選択、「名前」に任意のアプリケーション名を入力し、「作成」ボタンを押します。



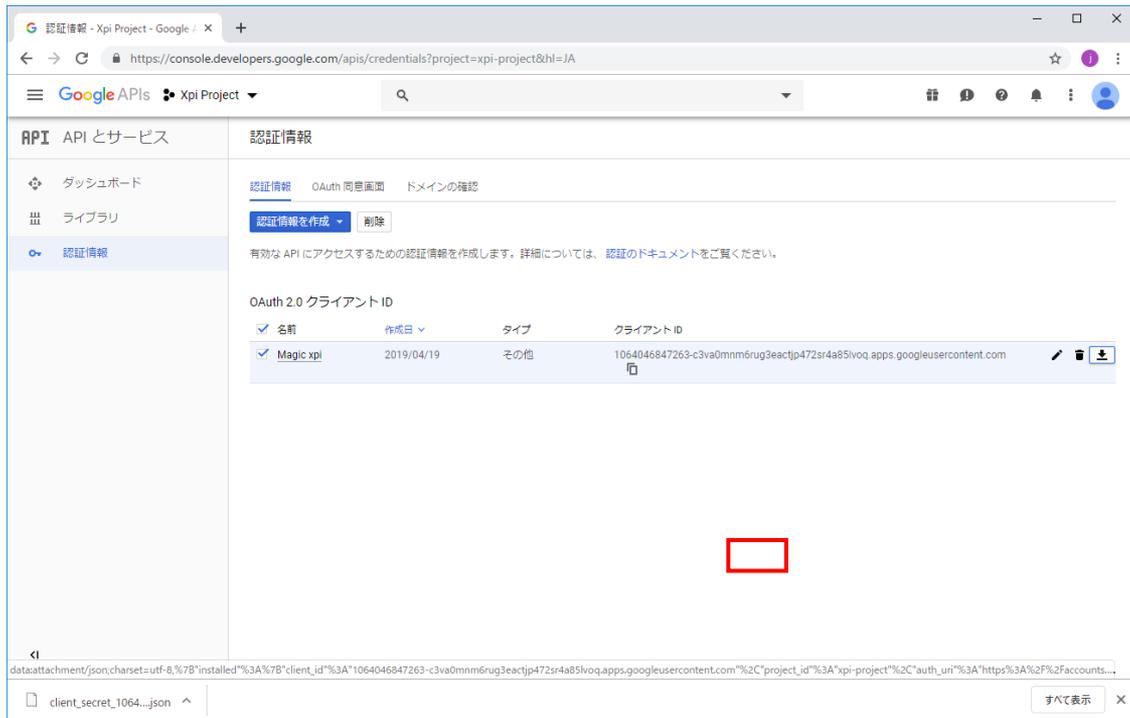
2.1 Google API を使用するための設定

- クライアントIDとクライアントシークレットが発行されたことを確認し、「OK」ボタンを押します。



2.1 Google API を使用するための設定

- 作成されたクライアントIDが一覧表示されます。
右端のダウンロードアイコンをクリックし、JSONファイルダウンロードしておきます。



The screenshot shows the Google API console interface. The left sidebar has '認証情報' (Credentials) selected. The main content area shows 'OAuth 2.0 クライアント ID' (OAuth 2.0 Client IDs) with a table listing one client ID: 'Magic xpi', created on 2019/04/19, of type 'その他' (Other), with client ID '1064046847263-c3va0mm6rug3eactjp472sr4a85ivoq.apps.googleusercontent.com'. A red box highlights the download icon (a downward arrow) in the rightmost column of the table row.

名前	作成日	タイプ	クライアント ID
✓ Magic xpi	2019/04/19	その他	1064046847263-c3va0mm6rug3eactjp472sr4a85ivoq.apps.googleusercontent.com



OUTPERFORM THE FUTURE™

2.2

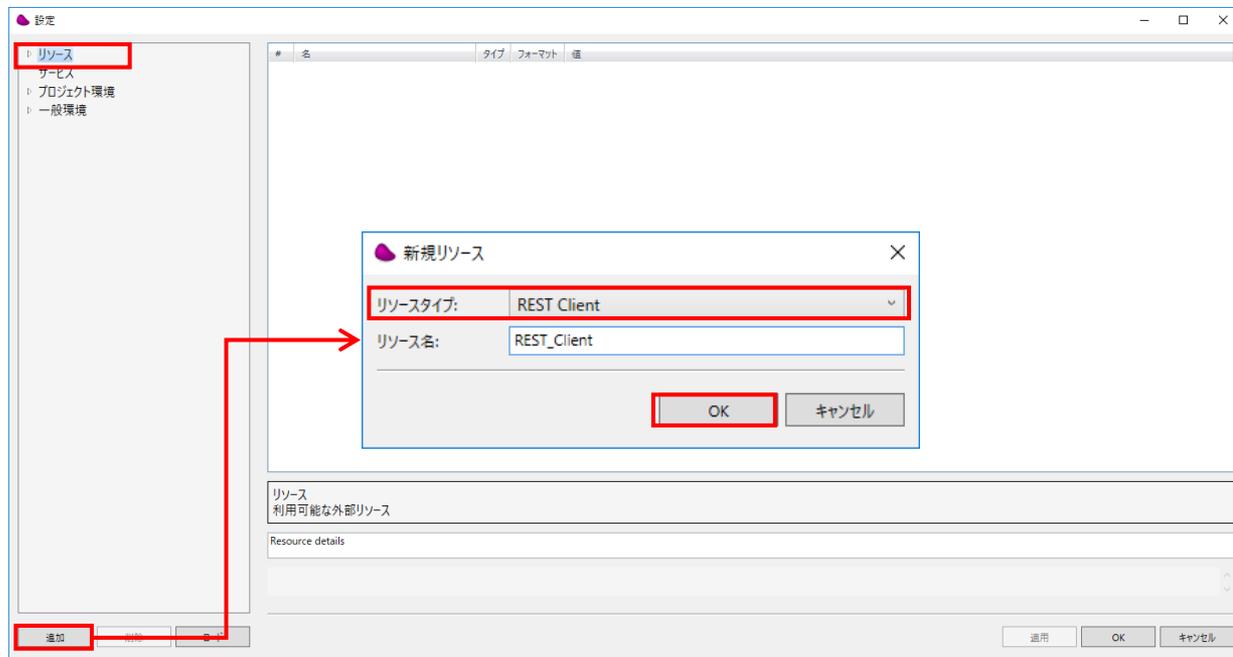
REST Client リソースの設定



OUTPERFORM THE FUTURE™

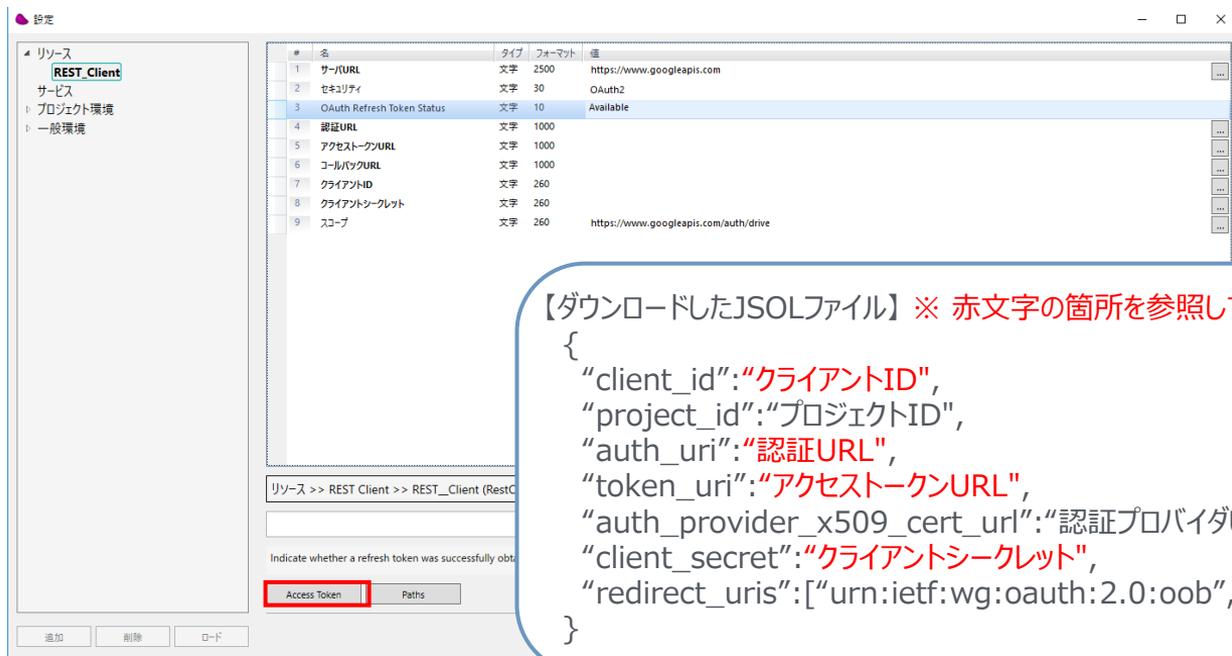
2.2 REST Clientリソースの設定

- ・ xpiスタジオのメニュー > プロジェクト > 設定より設定画面を開きます。
- ・ 左ペインで「リソース」を選択した状態で「追加」ボタンを押し、リソースタイプ「REST Client」のリソースを作成します。



2.2 REST Clientリソースの設定

- サーバURL「https://www.googleapis.com」を入力し、セキュリティに「OAuth」を選択します。
- Google Developers Console でダウンロードしたJSONファイルを元に認証情報を入力します。
- スコープ「https://www.googleapis.com/auth/drive」を入力し「Access Token」ボタンを押します。



#	名	タイプ	フォーマット	値
1	サーバURL	文字	2500	https://www.googleapis.com
2	セキュリティ	文字	30	OAuth2
3	OAuth Refresh Token Status	文字	10	Available
4	認証URL	文字	1000	
5	アクセストークンURL	文字	1000	
6	コールバックURL	文字	1000	
7	クライアントID	文字	260	
8	クライアントシークレット	文字	260	
9	スコープ	文字	260	https://www.googleapis.com/auth/drive

リソース >> REST Client >> REST_Client (RestC

Indicate whether a refresh token was successfully obt

Access Token Paths

追加 削除 ロード

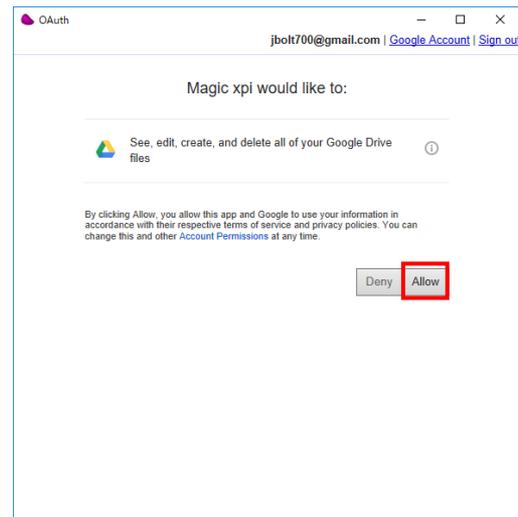
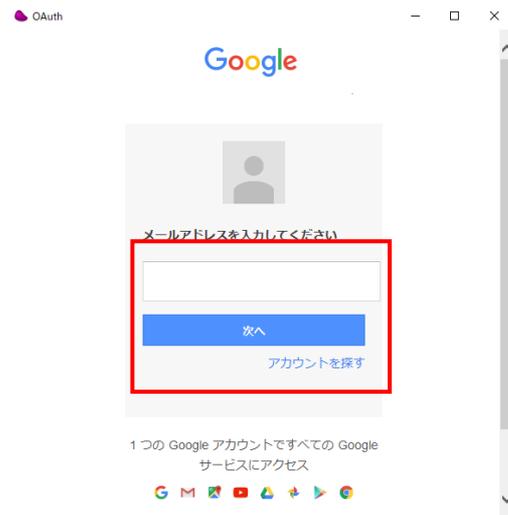
【ダウンロードしたJSONファイル】 ※ 赤字の箇所を参照して入力します

```
{
  "client_id": "クライアントID",
  "project_id": "プロジェクトID",
  "auth_uri": "認証URL",
  "token_uri": "アクセストークンURL",
  "auth_provider_x509_cert_url": "認証プロバイダURL",
  "client_secret": "クライアントシークレット",
  "redirect_uris": ["urn:ietf:wg:oauth:2.0:oob", "コールバックURL"]
}
```



2.2 REST Clientリソースの設定

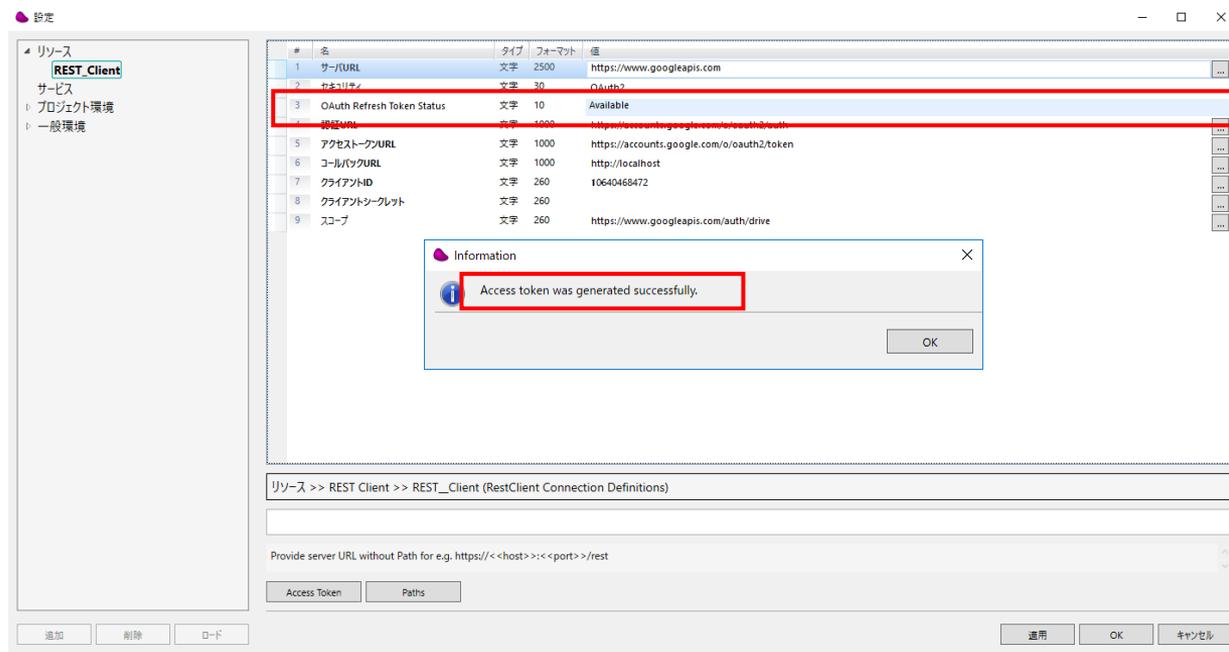
- 承認画面でユーザIDとパスワードを入力しログインします。
- 許可の確認画面(スコープによる)が表示されるので「Allow」ボタンを押します。



OUTPERFORM THE FUTURE™

2.2 REST Clientリソースの設定

- ・ アクセストークンが取得できたことを確認します。
- ・ 「OAuth Refresh Token Status」が「Available」になったことを確認します。



2.3

Google Driveへ ファイルをアップロード



OUTPERFORM THE FUTURE™

2.3 Google Driveへファイルをアップロード

【API仕様】 ※Google Drive API V3 を元に記載しています。

公式リファレンス(<https://developers.google.com/drive/api/v3/reference>)「Files:create」より

- ・ アップロード URI

POST <https://www.googleapis.com/upload/drive/v3/files>

- ・ クエリパラメータ(必須)

uploadType (media/multipart/resumable)

- ・ リクエスト Body

- a) メタデータ

ファイルのメタデータをJSON形式で指定 (application/json; charset=UTF-8)

- b) メディア

ファイルのデータを指定(任意のMIMEタイプ)

- ・ レスポンス Body



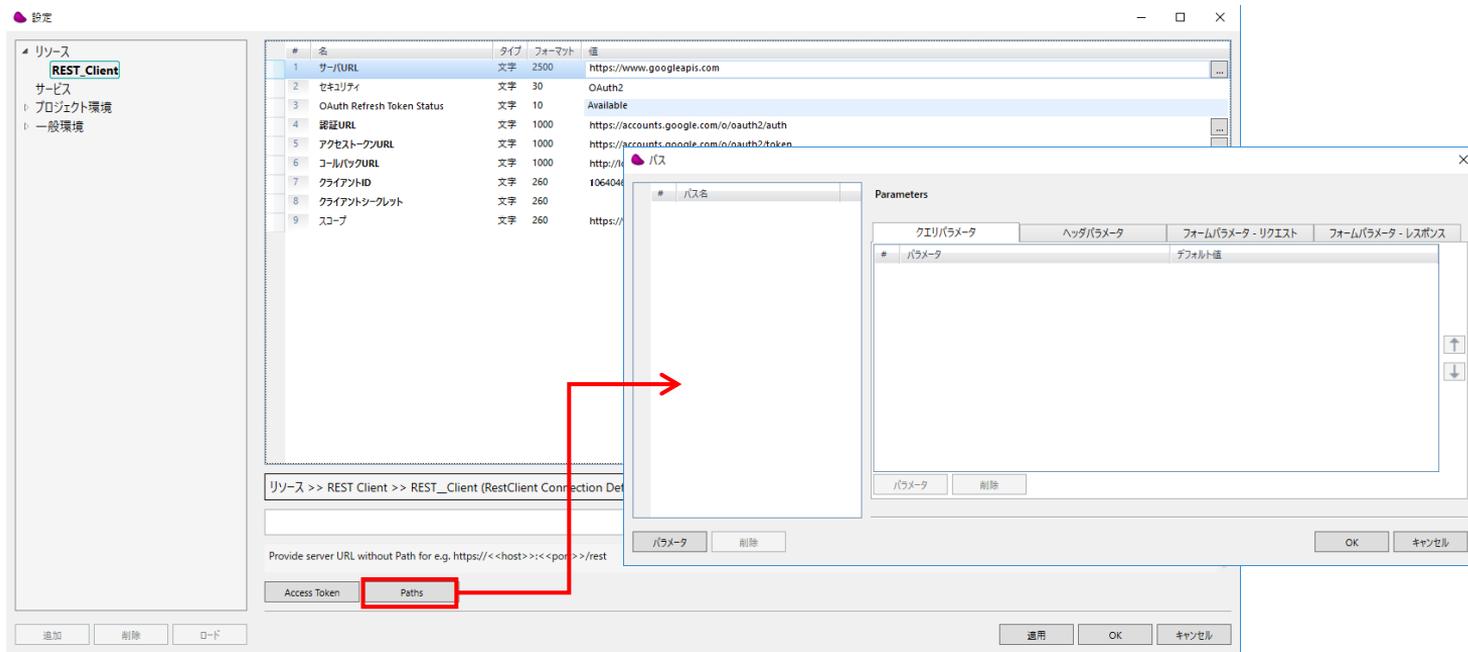
OUTPERFORM THE FUTURE™

ファイルのメタデータ(V3では最低限の情報のみ。必要なリソースはリクエスト時に指定)

2.3 Google Driveへファイルをアップロード

【URIパスとパラメータの設定】

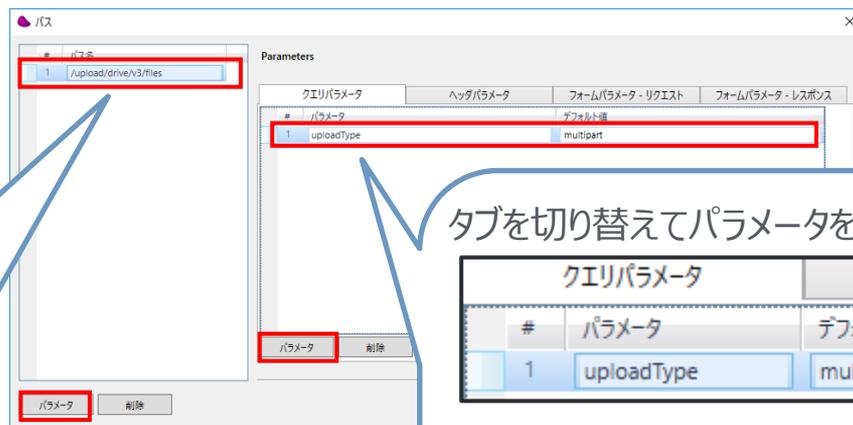
- ・ xpiスタジオのメニュー > プロジェクト > 設定より設定画面を開きます。
- ・ 左ペインで「リソース」> 「REST Client」を選択した状態で「Paths」ボタンを押します。



2.3 Google Driveへファイルをアップロード

【URIパスとパラメータの設定例】

- ・ 左ペイン下の「パラメータ」ボタンを押して、パスを入力します。
- ・ パスを選択した状態で、右ペイン下の「パラメータ」ボタンを押し、必要なパラメータを入力します。



サーバURLを除いた(APIバージョンやリソースを識別する部分)パスを指定します。ここでは、「/upload/drive/v3/files」

タブを切り替えてパラメータを設定します

クエリパラメータ		ヘッダパラメータ	
#	パラメータ	デフォルト値	
1	uploadType	multipart	

クエリパラメータ		ヘッダパラメータ		フォームパラメータ - リクエスト	
#	Name	Part Type	Content-Type	Encoding	
1	name	Text	application/json	UTF-8	
2	UploadFile	Attachment	image/jpeg	Binary	



OUTPERFORM THE FUTURE™

2.3 Google Driveへファイルをアップロード

【ステップの設定例】

項目	値
リソース名	プロパティペインでREST Clientリソースを選択。
パス	リソース設定のリストからパスを選択。
オペレーション	「POST」を選択。
リクエスト Content-Type	ボタンを押して「multipart/form-data」を選択
レスポンス Content-Type	ボタンを押して「application/json」を選択
結果保存	「変数」を選択し、結果を格納する変数を指定。 APIから戻されるJSONデータが格納されます
ステータスコード	HTTPステータスを格納する変数を指定
メッセージ	HTTPレスポンスメッセージを格納する変数を指定
成功	「Variable」を選択し、成否を格納する変数を指定。 成功時は「True」、失敗時は「False」が格納されます。



2.3 Google Driveへファイルをアップロード

【マッパー画面の設定例】

ルートノードは自動的に「変数」に設定され、
リクエスト用のデータがXML変数「C.UserXML」に格納されます。

ここではフォームパラメータのプロパティ「計算値」に直接、値を指定していますが、送り元にFlat Fileなどを指定し、送り元から値をマッピングすることも可能です。

プロパティ
RESTClient XMLスキーマ
変数 C.UserXML
変数
送り元タイプ特性または送り先タイプ特性が変数に設定されている場合は、使用するデータ...

プロパティ
name Simple Field
計算値 { "name": "Upload Magic Xpi" }
計算値

プロパティ
UploadFile Simple Field
計算値 'C:%temp%Magicxpi.jpg'
計算値

画像ファイル「C:¥temp¥Magicxpi.jpg」をファイル名
「Upload Magic Xpi」でアップロード



2.3 Google Driveへファイルをアップロード

【実行結果】

アップロードファイル



The screenshot shows the Google Drive interface. A file named "Upload Magic Xpi" is listed in the "マイドライブ" (My Drive) section. A callout box points to the file name with the text "ファイル名「Upload Magic Xpi」でアップロード". The file's owner is listed as "自分" (Me) and it was updated at "16:30 自分".

レスポンスデータ

```
ContextView_BLOB.TXT - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
{
  "kind": "drive#file",
  "id": "1li1HG9StZwhbu1G-Hyj6fyUTlhm2pQC8",
  "name": "Upload Magic Xpi",
  "mimeType": "image/jpeg"
}
```

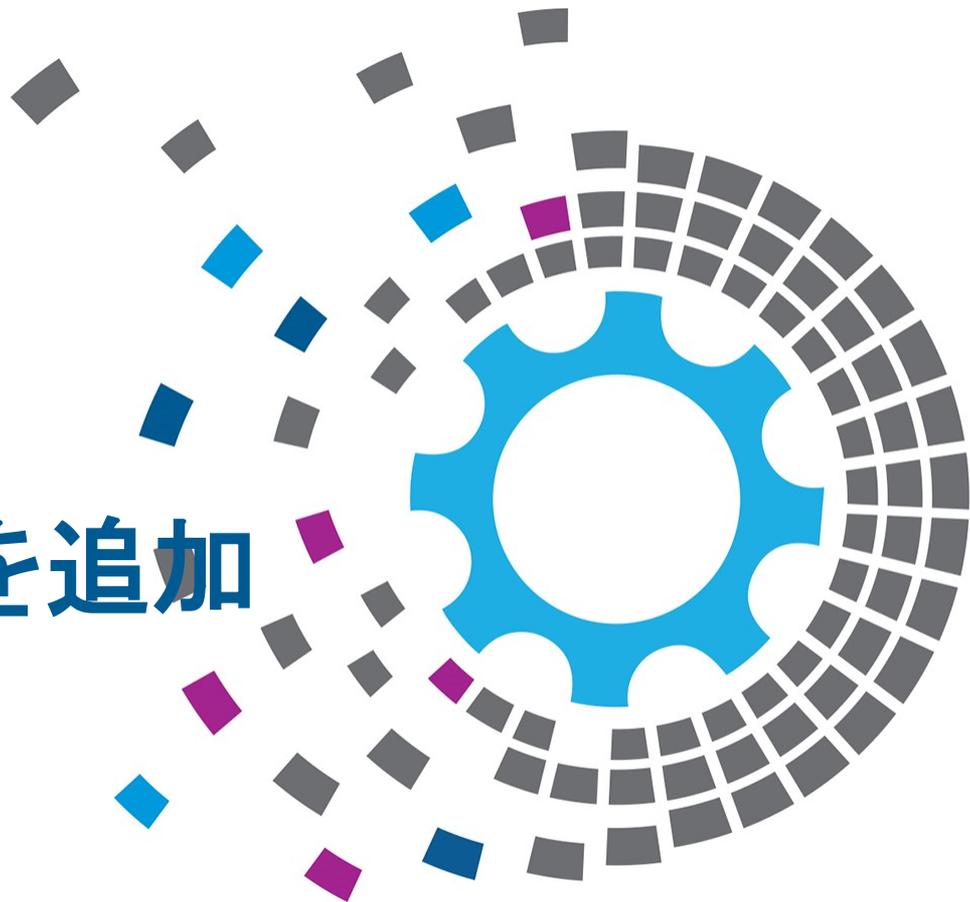
※次章以降の更新・削除処理では、こちらのレスポンスデータから「FileId」を取得したものを利用しています。



OUTPERFORM THE FUTURE™

2.4

Google Driveの ファイルに説明を追加



OUTPERFORM THE FUTURE™

2.4 Google Driveのファイルに説明を追加

【API仕様】 ※Google Drive API V3 を元に記載しています。

公式リファレンス(<https://developers.google.com/drive/api/v3/reference>)「Files:update」より

- ・ メタデータ URI

PATCH <https://www.googleapis.com/drive/v3/files/{fileId}>

※注 V3ではPATCHメソッドを使用。PUTメソッドではエラーとなります。

- ・ パスパラメータ(必須)

fileId (対象のファイルID)

- ・ リクエスト Body

ファイルのメタデータをJSON形式で指定 (application/json; charset=UTF-8)

- ・ レスポンス Body

ファイルのメタデータ(V3では最低限の情報のみ。必要なリソースはリクエスト時に指定)

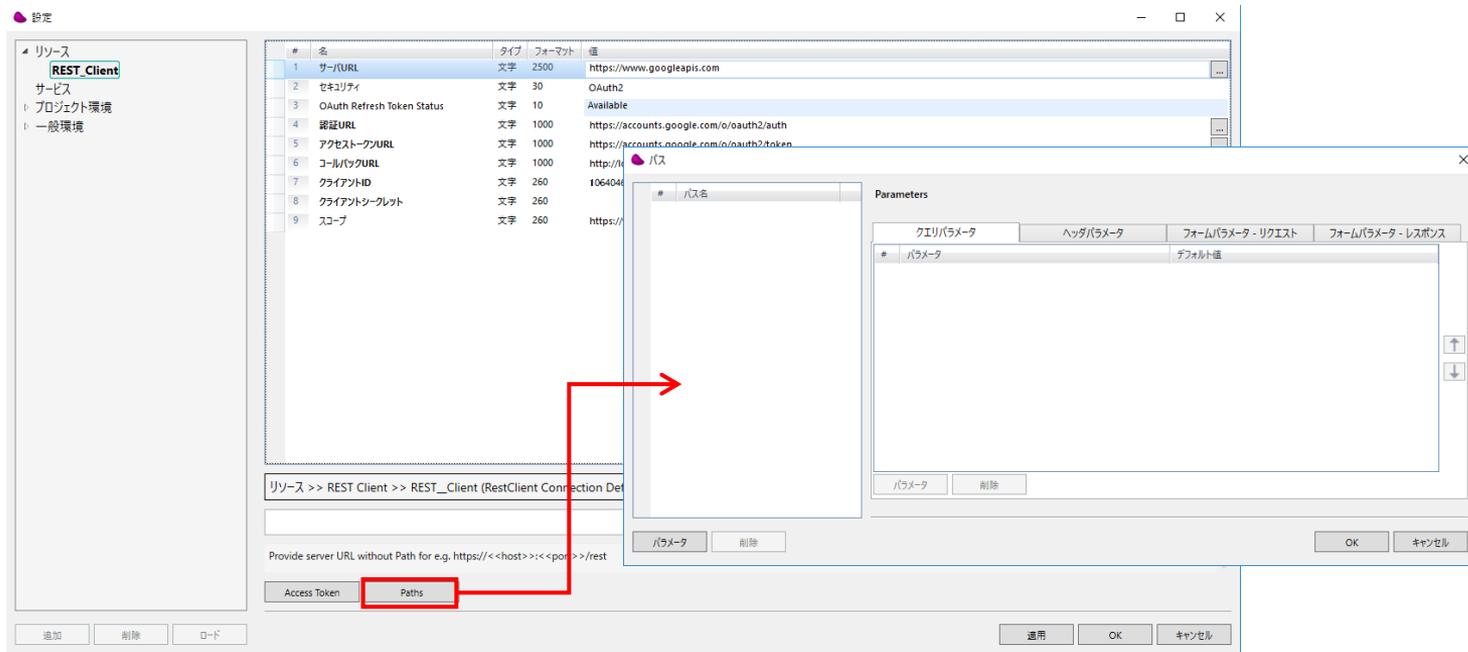


OUTPERFORM THE FUTURE™

2.4 Google Driveのファイルに説明を追加

【URIパスとパラメータの設定】

- ・ xpiスタジオのメニュー > プロジェクト > 設定より設定画面を開きます。
- ・ 左ペインで「リソース」> 「REST Client」を選択した状態で「Paths」ボタンを押します。

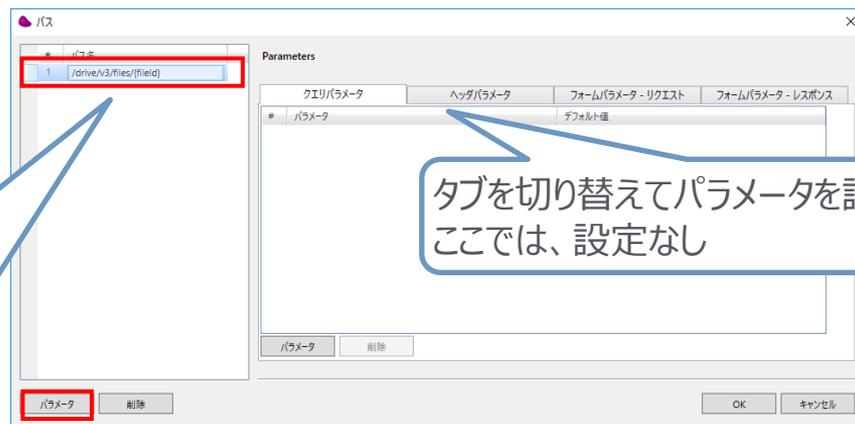


OUTPERFORM THE FUTURE™

2.4 Google Driveのファイルに説明を追加

【URIパスとパラメータの設定例】

- ・ 左ペイン下の「パラメータ」ボタンを押して、パスを入力します。
- ・ パスを選択した状態で、右ペイン下の「パラメータ」ボタンを押し、必要なパラメータを入力します。



パスパラメータは中括弧{ }で括弧することで動的に値を設定可能です。ここでは、「 /drive/v3/files/{fileId} 」

タブを切り替えてパラメータを設定します。ここでは、設定なし



2.4 Google Driveのファイルに説明を追加

【ステップの設定例】

REST Client Configuration

接続

リソース名: REST_Client

オペレーション

パス: /drive/v3/files/{fileId}

オペレーション: Patch

セキュリティ: Yes

リクエスト Content Type: application/json

レスポンス Content Type: application/json

要求エンコーディング: UTF-8

出力

結果保存: Variable (C.UserBlob)

ヘッダフィルタ:

ヘッダ:

ステータスコード: C.UserCode

メッセージ: C.UserString

成功: Variable (F.Success)

パラメータ スキーマ更新 OK キャンセル

項目	値
リソース名	プロパティペインでREST Clientリソースを選択。
パス	リソース設定のリストからパスを選択。
オペレーション	「PATCH」を選択。
リクエスト Content-Type	ボタンを押して「application/json」を選択
レスポンス Content-Type	ボタンを押して「application/json」を選択
要求エンコーディング	リクエストの一部として送信されるデータのエンコードを指定。リストから「UTF-8」を選択
結果保存	「変数」を選択し、結果を格納する変数を指定。APIから戻されるJSONデータが格納されます
ステータスコード	HTTPステータスを格納する変数を指定
メッセージ	HTTPレスポンスメッセージを格納する変数を指定
成功	「Variable」を選択し、成否を格納する変数を指定。成功時は「True」、失敗時は「False」が格納されます。



2.4 Google Driveのファイルに説明を追加

【マップ画面の設定例 -送り元-】

- ・ ファイルのメタ情報を更新するには、ファイルIDを知る必要があります。
下記の例では、アップロード時のレスポンスデータから取得したIDを送り元に設定しています。

アップロード時に取得したファイルIDをパスパラメータの「fileId」へマッピング。
不明な場合は事前にファイル一覧の取得APIでメタ情報を取得する必要があります。



OUTPERFORM THE FUTURE™

2.4 Google Driveのファイルに説明を追加

【マップ画面の設定例 -送り先-】

ルートノードは自動的に「変数」に設定され、
リクエスト用のデータがXML変数「C.UserXML」に格納されます。

プロパティ
RESTClient XMLスキマ
変数 C.UserXML
変数
送り元タイプ特性または送り先タイプ特性が変数に設定されている場合は、使用するデータ...

プロパティ
DataBlob Simple Field
計算値 '{ "description": "Description is Added by Magic Xpi" }'
計算値

ファイルの説明に
「 Description is Added by Magic Xpi 」
を追加

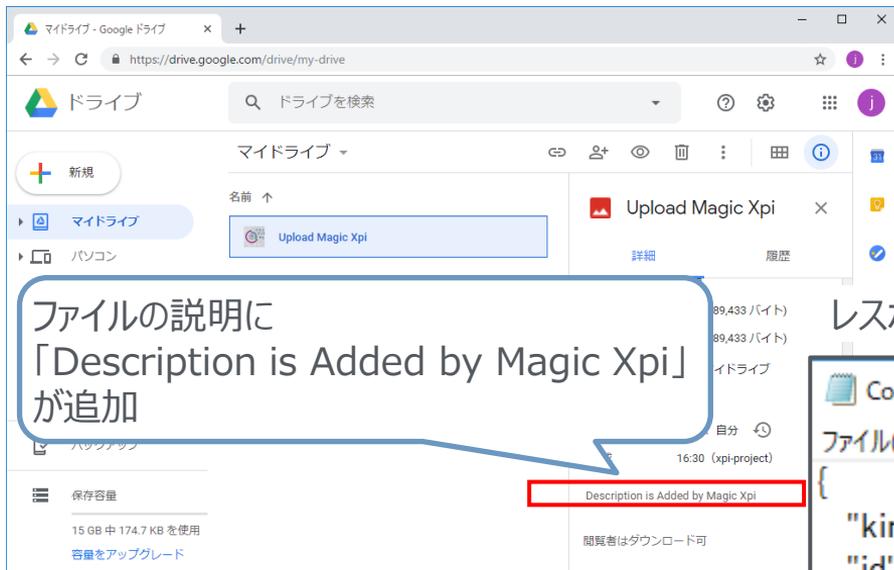


OUTPERFORM THE FUTURE™

2.4 Google Driveのファイルに説明を追加

【実行結果】

アップロードファイル



ファイルの説明に
「Description is Added by Magic Xpi」
が追加

レスポンスデータ

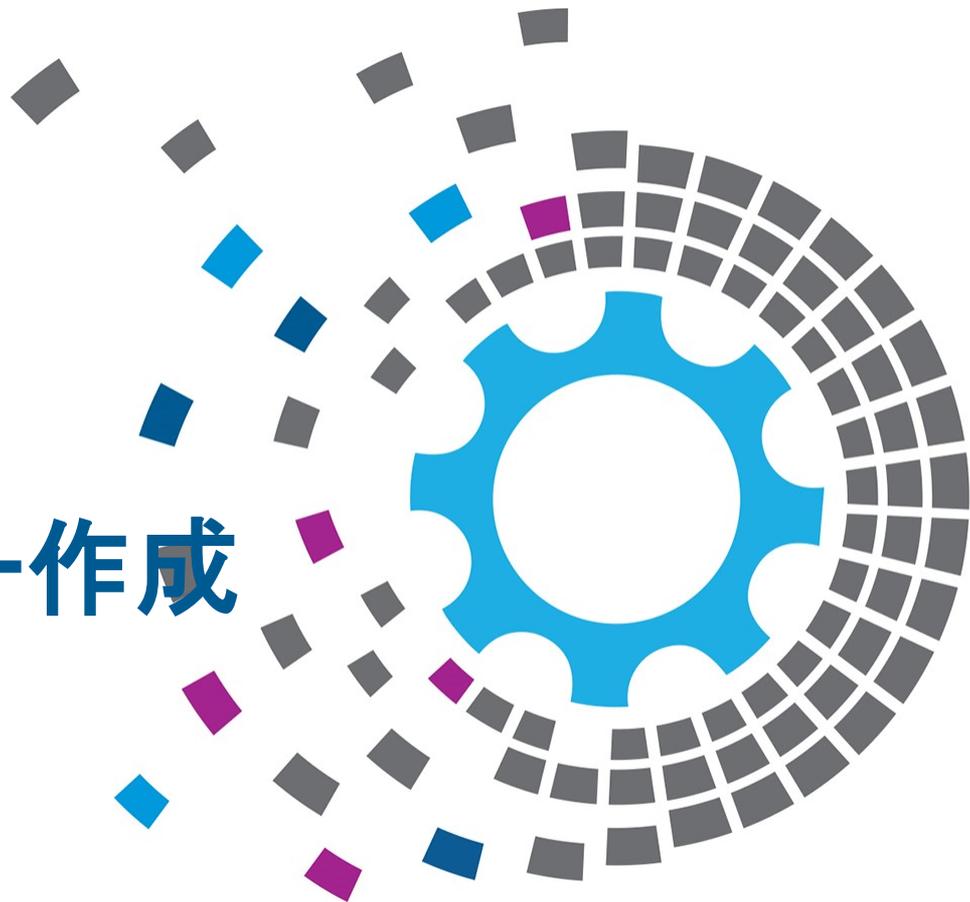
```
ContextView_BLOB.TXT - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
{
  "kind": "drive#file",
  "id": "1li1HG9StZwhbu1G-Hyj6fyUTlhM2pQC8",
  "name": "Upload Magic Xpi",
  "mimeType": "image/jpeg"
}
```



OUTPERFORM THE FUTURE™

2.5

Google Driveへ ファイルのコピー作成



OUTPERFORM THE FUTURE™

2.5 Google Driveへファイルのコピー作成

【API仕様】 ※Google Drive API V3 を元に記載しています。

公式リファレンス(<https://developers.google.com/drive/api/v3/reference>)「Files:copy」より

- ・ メタデータ URI

POST `https://www.googleapis.com/drive/v3/files/ {fileId} /copy`

- ・ パスパラメータ(必須)

fileId (対象のファイルID)

- ・ リクエスト Body

ファイルのメタデータをJSON形式で指定 (application/json; charset=UTF-8)

- ・ レスポンス Body

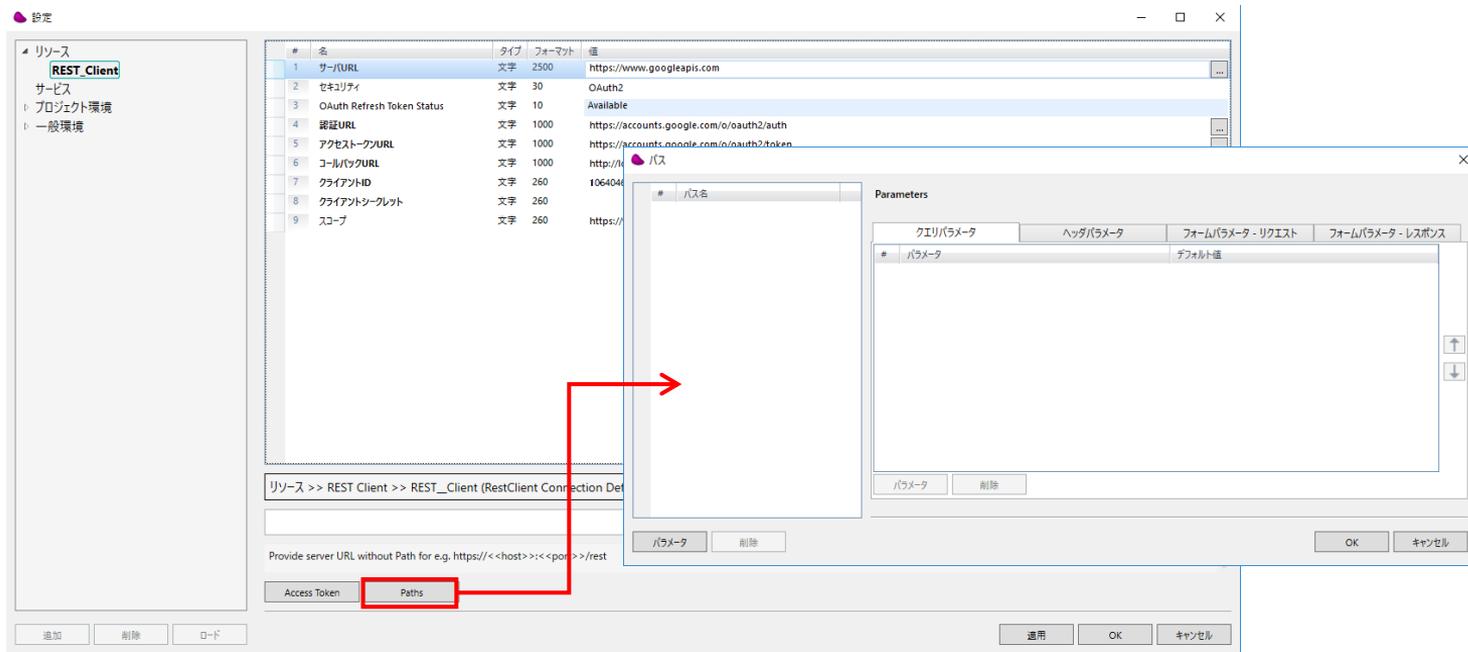
ファイルのメタデータ(V3では最低限の情報のみ。必要なリソースはリクエスト時に指定)



2.5 Google Driveへファイルのコピー作成

【URIパスとパラメータの設定】

- ・ xpiスタジオのメニュー > プロジェクト > 設定より設定画面を開きます。
- ・ 左ペインで「リソース」> 「REST Client」を選択した状態で「Paths」ボタンを押します。

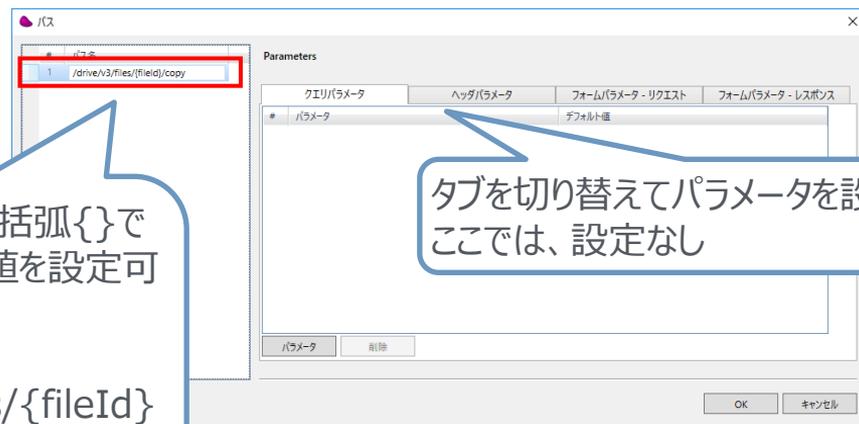


OUTPERFORM THE FUTURE™

2.5 Google Driveへファイルのコピー作成

【URIパスとパラメータの設定例】

- ・ 左ペイン下の「パラメータ」ボタンを押して、パスを入力します。
- ・ パスを選択した状態で、右ペイン下の「パラメータ」ボタンを押し、必要なパラメータを入力します。



パスパラメータは中括弧{ }で括ることで動的に値を設定可能です。
ここでは、
「/drive/v3/files/{fileId}/copy」



2.5 Google Driveへファイルのコピー作成

【ステップの設定例】

REST Client Configuration

接続

リソース名: REST_Client

オペレーション

パス: /drive/v3/files/{fileId}/copy

オペレーション: Post

セキュリティ: Yes

リクエスト Content Type: application/json

レスポンス Content Type: application/json

要求エンコーディング: UTF-8

出力

結果保存: Variable C.UserBlob

ヘッダフィルタ:

ヘッダ:

ステータスコード: C.UserCode

メッセージ: C.UserString

成功: Variable F.Success

パラメータ スキーマ更新 OK キャンセル

項目	値
リソース名	プロパティペインでREST Clientリソースを選択。
パス	リソース設定のリストからパスを選択。
オペレーション	「POST」を選択。
リクエスト Content-Type	ボタンを押して「application/json」を選択
レスポンス Content-Type	ボタンを押して「application/json」を選択
要求エンコーディング	リクエストの一部として送信されるデータのエンコードを指定。リストから「UTF-8」を選択
結果保存	「変数」を選択し、結果を格納する変数を指定。APIから戻されるJSONデータが格納されます
ステータスコード	HTTPステータスを格納する変数を指定
メッセージ	HTTPレスポンスメッセージを格納する変数を指定
成功	「Variable」を選択し、成否を格納する変数を指定。成功時は「True」、失敗時は「False」が格納されます。



2.5 Google Driveへファイルのコピー作成

【マップ画面の設定例 -送り元-】

- ・ ファイルのコピーには、対象となるファイルのファイルIDを知る必要があります。
下記の例では、アップロード時のレスポンスデータから取得したIDを送り元に設定しています。

アップロード時に取得したファイルIDをパスパラメータの「fileId」へマッピング。
不明な場合は事前にファイル一覧の取得APIでメタ情報を取得する必要があります。



OUTPERFORM THE FUTURE™

2.5 Google Driveへファイルのコピー作成

【マップ画面の設定例 -送り先-】

ルートノードは自動的に「変数」に設定され、
リクエスト用のデータがXML変数「C.UserXML」に格納されます。

Make a file copy (Make a copy of file V3)*

送り元

- FlatFile_1
 - Record (0)
 - FileID (0)

送り先

- RESTClient
 - Path
 - fileID (0)
 - RequestBody
 - Form
 - DataBlob
 - ResponseBody
 - Form

プロパティ

RESTClient XMLスキーマ

変数

C.UserXML

変数

送り元タイプ特性または送り先タイプ特性が変数に設定されている場合は、使用するデータ...

プロパティ

DataBlob Simple Field

計算値

'!(\"name\": \"Copy of Image Uploaded from Magic Xpi\")'

ドキュメンテーション

Make a copy of file V3 (GoogleDrive)

ファイル名を
「Copy of Image Uploaded from Magic Xpi」
でコピー

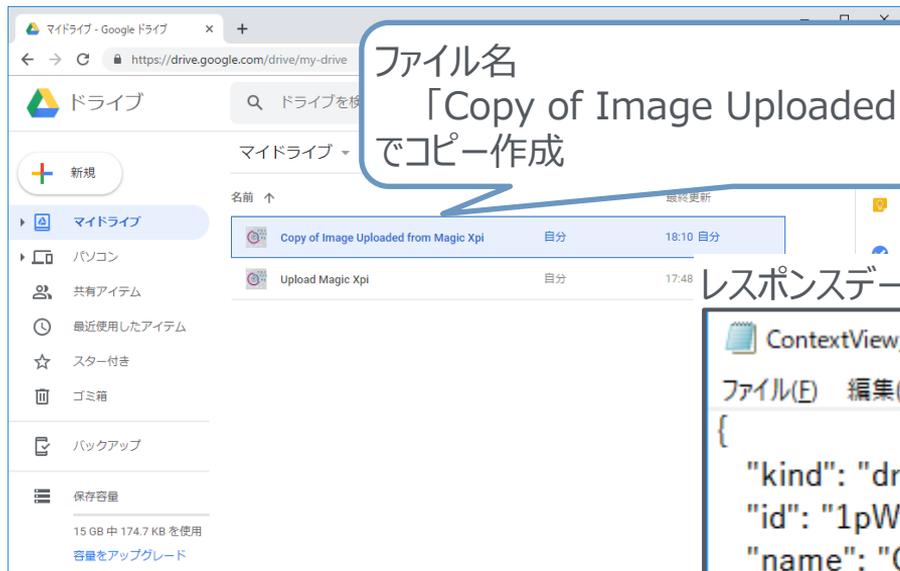


OUTPERFORM THE FUTURE™

2.5 Google Driveへファイルのコピー作成

【実行結果】

アップロードファイル



The screenshot shows the Google Drive interface. A callout box points to a file entry in the 'マイドライブ' (My Drive) section. The file is named 'Copy of Image Uploaded from Magic Xpi' and was uploaded by '自分' (me) at 18:10. Below it, another file 'Upload Magic Xpi' is visible, uploaded at 17:48.

名前	最終更新
Copy of Image Uploaded from Magic Xpi	自分 18:10
Upload Magic Xpi	自分 17:48

ファイル名
「Copy of Image Uploaded from Magic Xpi」
でコピー作成

レスポンスデータ

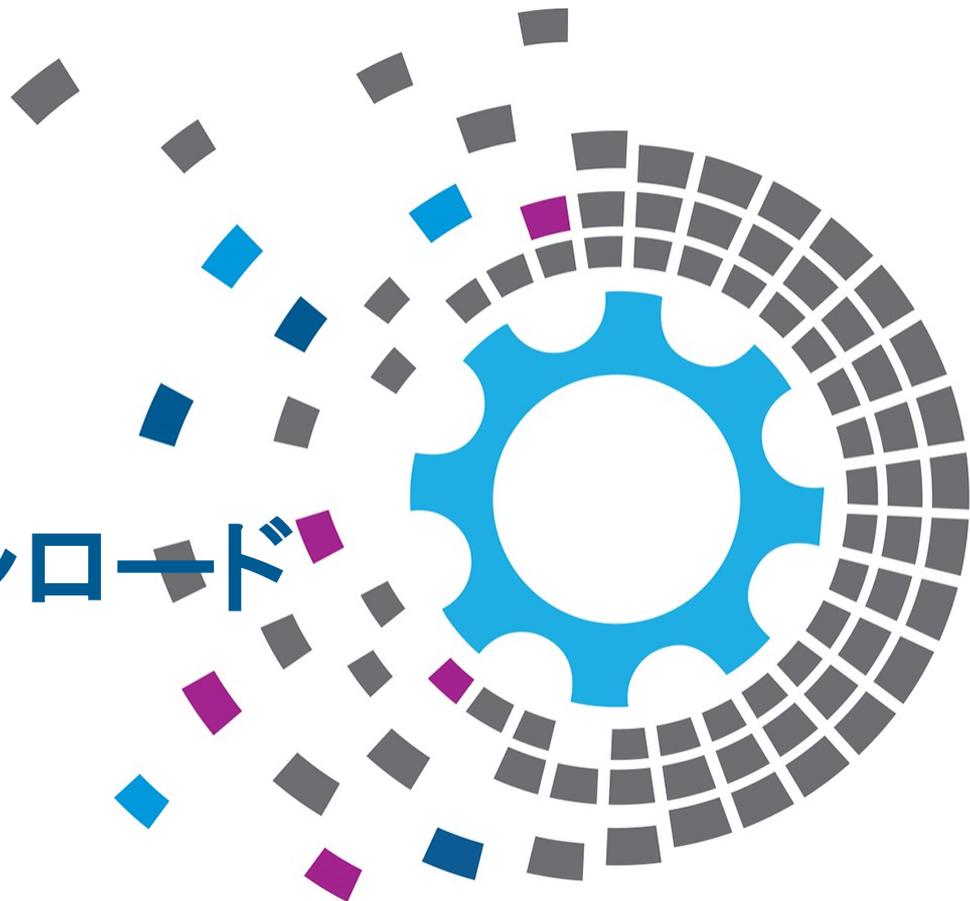
```
ContextView_BLOB.TXT - メモ帳
ファイル(E) 編集(E) 書式(O) 表示(V) ヘルプ(H)
{
  "kind": "drive#file",
  "id": "1pW3QFuH9ONAL2sOqSeXiZ3wzakEVJmEY",
  "name": "Copy of Image Uploaded from Magic Xpi",
  "mimeType": "image/jpeg"
}
```



OUTPERFORM THE FUTURE™

2.6

Google Driveの ファイルをダウンロード



OUTPERFORM THE FUTURE™

2.6 Google Driveのファイルをダウンロード

【API仕様】 ※Google Drive API V3 を元に記載しています。

公式リファレンス(<https://developers.google.com/drive/api/v3/reference>)「Files:get」より

- ・ メタデータ URI

GET <https://www.googleapis.com/drive/v3/files/{fileId}>

- ・ パスパラメータ(必須)

fileId (対象のファイルID)

- ・ クエリパラメータ(必須)

alt=media

- ・ リクエスト Body

なし

- ・ レスポンス Body

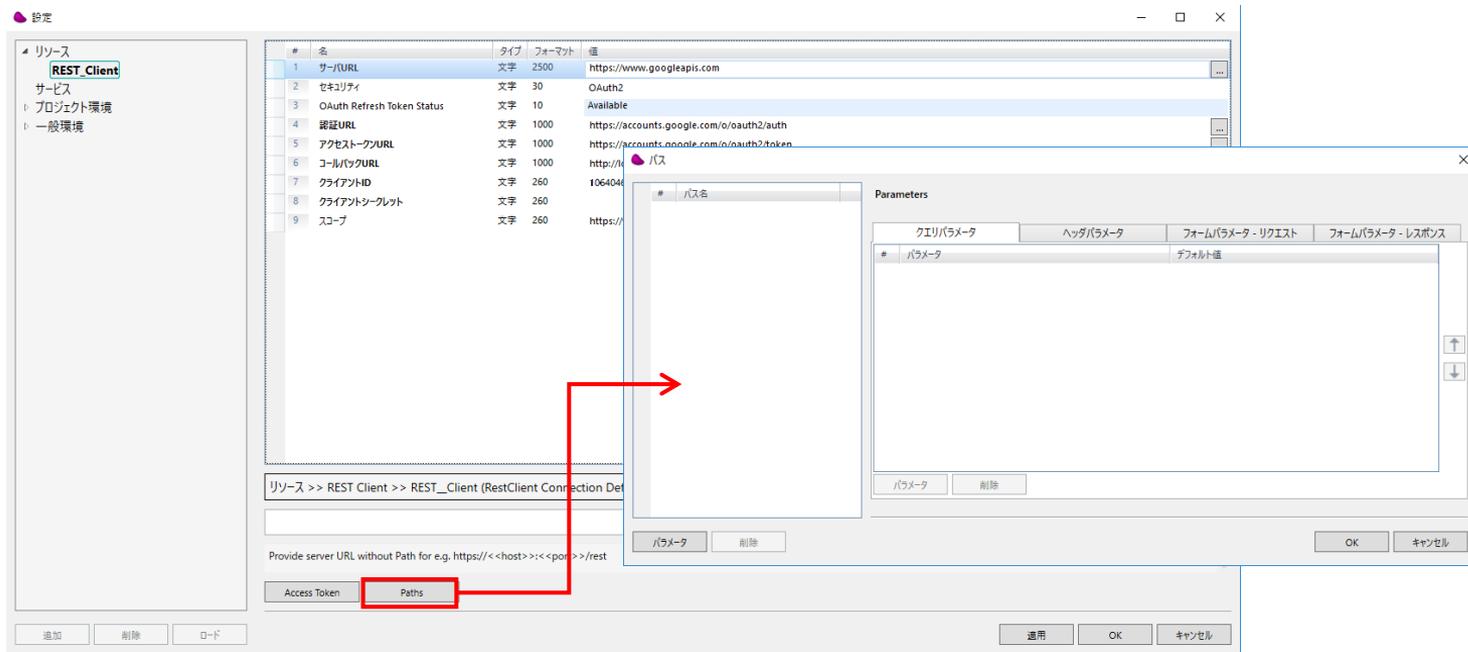
ファイルデータ(実体)



2.6 Google Driveのファイルをダウンロード

【URIパスとパラメータの設定】

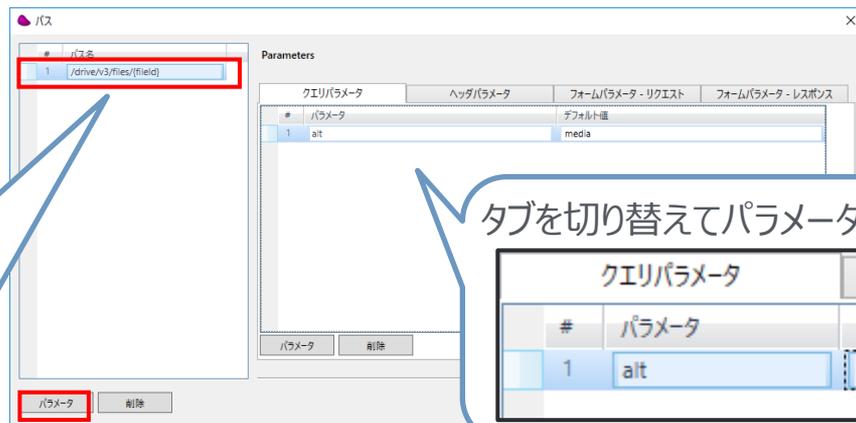
- ・ xpiスタジオのメニュー > プロジェクト > 設定より設定画面を開きます。
- ・ 左ペインで「リソース」> 「REST Client」を選択した状態で「Paths」ボタンを押します。



2.6 Google Driveのファイルをダウンロード

【URIパスとパラメータの設定例】

- ・ 左ペイン下の「パラメータ」ボタンを押して、パスを入力します。
- ・ パスを選択した状態で、右ペイン下の「パラメータ」ボタンを押し、必要なパラメータを入力します。



パスパラメータは中括弧{ }で括弧で動的に値を設定可能です。ここでは、「 /drive/v3/files/{fileId} 」

タブを切り替えてパラメータを設定します

クエリパラメータ		ヘッダパラメータ
#	パラメータ	デフォルト値
1	alt	media



OUTPERFORM THE FUTURE™

2.6 Google Driveのファイルをダウンロード

【ステップの設定例】

REST Client Configuration

接続

リソース名: REST_Client

オペレーション

パス: /drive/v3/files/{fileid}

オペレーション: Get

セキュリティ: Yes

リクエスト Content Type: ...

レスポンス Content Type: ...

要求エンコーディング: Keep existing Charset: ...

出力

結果保存: Variable C.UserBlob

ヘッダフィルタ: ...

ヘッダ: ...

ステータスコード: C.UserCode

メッセージ: C.UserString

成功: Variable F.Success

パラメータ スキーマ更新 OK キャンセル

項目	値
リソース名	プロパティペインでREST Clientリソースを選択。
パス	リソース設定のリストからパスを選択。
オペレーション	「GET」を選択。
結果保存	「変数」を選択し、結果を格納する変数を指定。 APIから戻されるJSONデータが格納されます ※下記参照
ステータスコード	HTTPステータスを格納する変数を指定
メッセージ	HTTPレスポンスメッセージを格納する変数を指定
成功	「Variable」を選択し、成否を格納する変数を指定。 成功時は「True」、失敗時は「False」が格納されます。

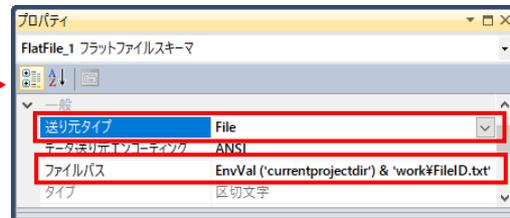
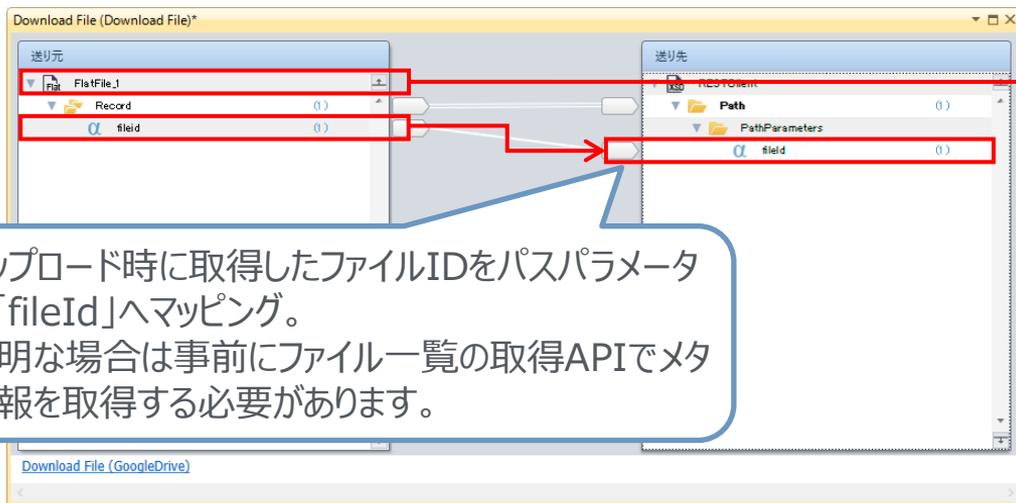
※ダウンロードファイルを直接保存する場合、結果保存に“File”を選択し、保存先のパスを指定します。



2.6 Google Driveのファイルをダウンロード

【マップ画面の設定例 -送り元-】

- ・ ファイルをのダウンロードするには、対象となるファイルのファイルIDを知る必要があります。下記の例では、アップロード時のレスポンスデータから取得したIDを送り元に設定しています。



2.6 Google Driveのファイルをダウンロード

【マップー画面の設定例 -送り先-】

ルートノードは自動的に「変数」に設定され、
リクエスト用のデータがXML変数「C.UserXML」に格納されます。

The screenshot shows the configuration interface for a 'Download File' action. On the left, the '送り元' (Source) tree shows a 'FlatFile_1' folder containing a 'Record' folder and a 'fileid' field. On the right, the '送り先' (Destination) tree shows a 'RESTClient' folder containing a 'Path' folder and a 'PathParameters' folder with a 'fileid' field. A red box highlights the 'RESTClient' folder, with an arrow pointing to the 'プロパティ' (Properties) window on the right. The 'プロパティ' window shows the 'RESTClient XMLスキーマ' (RESTClient XML Schema) and the '変数' (Variable) 'C.UserXML'.

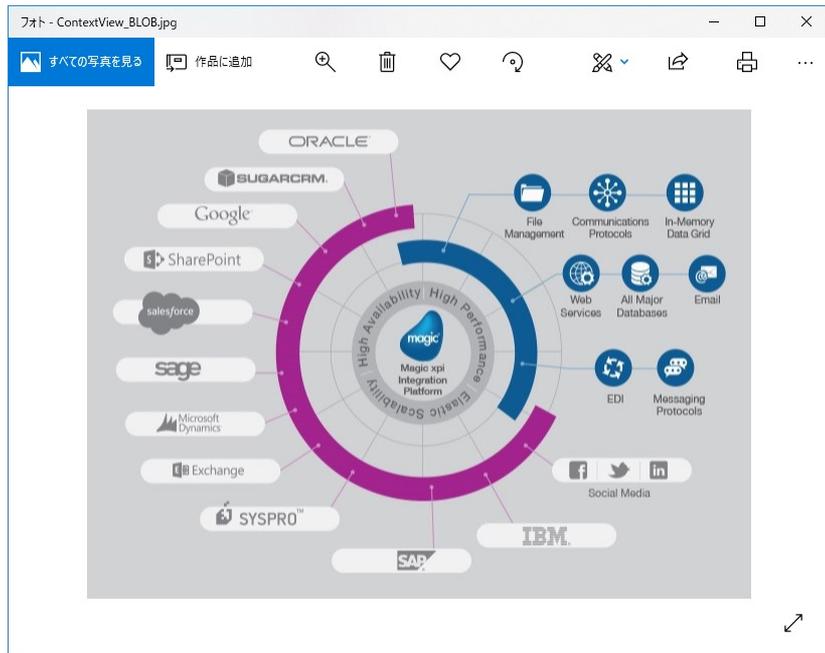


OUTPERFORM THE FUTURE™

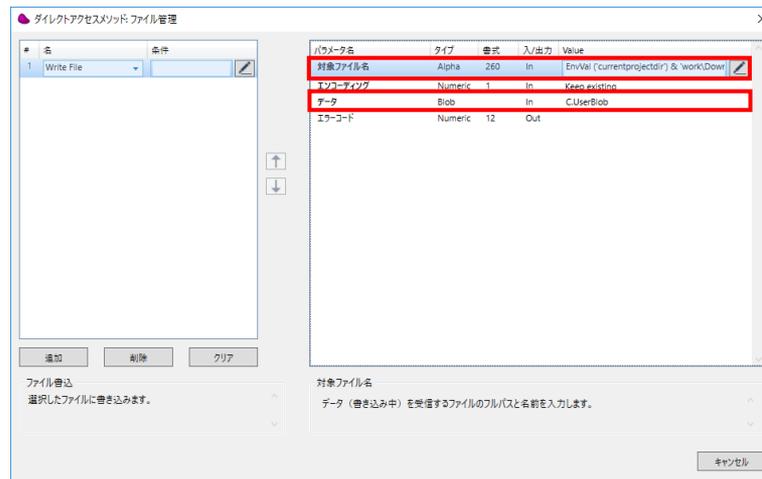
2.6 Google Driveのファイルをダウンロード

【実行結果】

レスポンスデータ



結果保存先に“Variable”を指定した場合、ダウンロードファイルはファイル管理コンポーネントを仕様して復元します。



2.7

Google Driveから ファイルを削除



OUTPERFORM THE FUTURE™

2.7 Google Driveからファイルを削除

【API仕様】 ※Google Drive API V3 を元に記載しています。

公式リファレンス(<https://developers.google.com/drive/api/v3/reference>)「Files:delete」より

- ・ メタデータ URI

DELETE <https://www.googleapis.com/drive/v3/files/{fileId}>

- ・ パスパラメータ(必須)

fileId (対象のファイルID)

- ・ リクエスト Body

なし

- ・ レスポンス Body

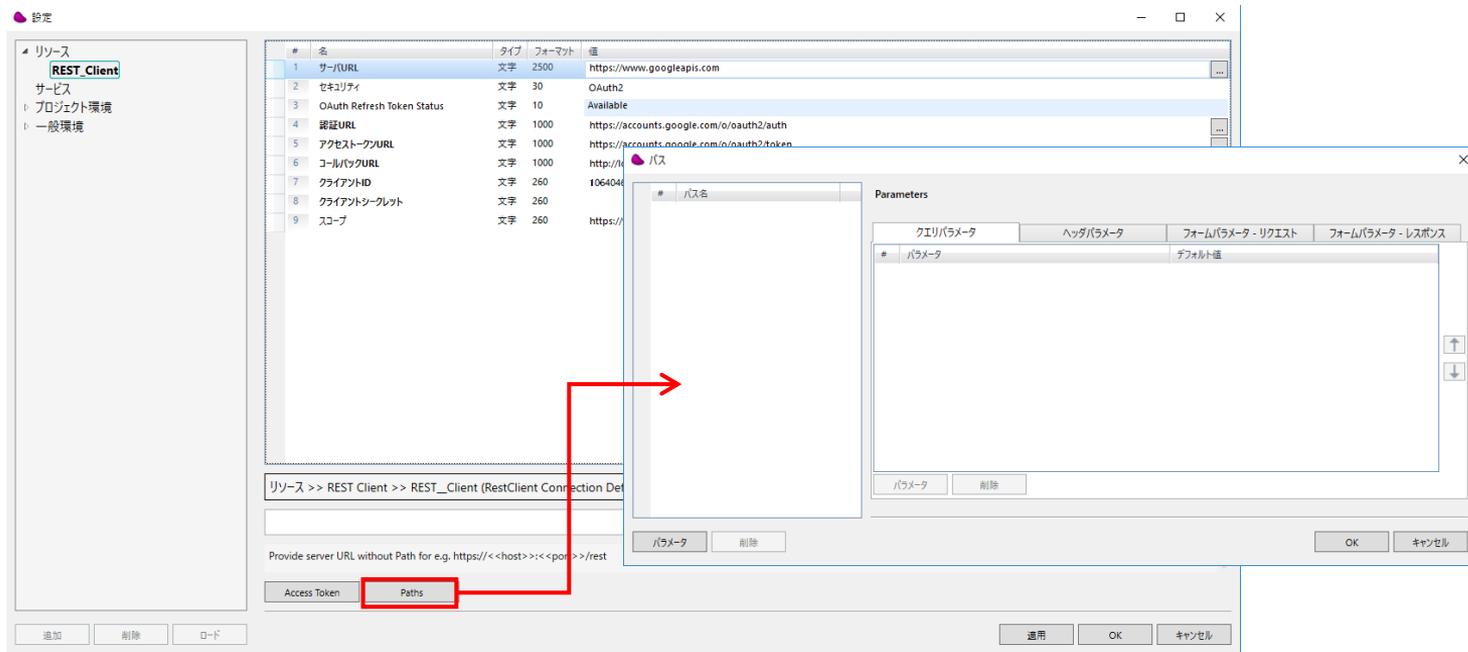
空のレスポンスBody



2.7 Google Driveからファイルを削除

【URIパスとパラメータの設定】

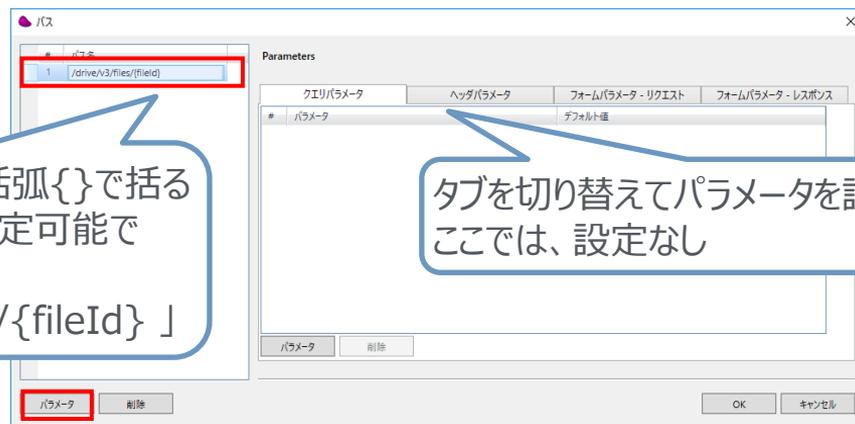
- ・ xpiスタジオのメニュー > プロジェクト > 設定より設定画面を開きます。
- ・ 左ペインで「リソース」> 「REST Client」を選択した状態で「Paths」ボタンを押します。



2.7 Google Driveからファイルを削除

【URIパスとパラメータの設定例】

- ・ 左ペイン下の「パラメータ」ボタンを押して、パスを入力します。
- ・ パスを選択した状態で、右ペイン下の「パラメータ」ボタンを押し、必要なパラメータを入力します。



OUTPERFORM THE FUTURE™

2.7 Google Driveからファイルを削除

【ステップの設定例】

REST Client Configuration

接続

リソース名: REST_Client

オペレーション

パス: /drive/v3/files/{fileid}

オペレーション: Delete

セキュリティ: Yes

リクエスト Content Type: ...

レスポンス Content Type: ...

要求エンコーディング: Keep existing Charset: ...

出力

結果保存: Variable C.UserBlob

ヘッダフィルタ: ...

ヘッダ: ...

ステータスコード: C.UserCode

メッセージ: C.UserString

成功: Variable F.Success

パラメータ スキーマ更新 OK キャンセル

項目	値
リソース名	プロパティペインでREST Clientリソースを選択。
パス	リソース設定のリストからパスを選択。
オペレーション	「DELETE」を選択。
結果保存	「変数」を選択し、結果を格納する変数を指定。 APIから戻されるJSONデータが格納されます
ステータスコード	HTTPステータスを格納する変数を指定
メッセージ	HTTPレスポンスメッセージを格納する変数を指定
成功	「Variable」を選択し、成否を格納する変数を指定。 成功時は「True」、失敗時は「False」が格納されます。



2.7 Google Driveからファイルを削除

【マップ画面の設定例 -送り元-】

- ・ ファイルの削除には、対象となるファイルのファイルIDを知る必要があります。
下記の例では、アップロード時のレスポンスデータから取得したIDを送り元に設定しています。

アップロード時に取得したファイルIDをパスパラメータの「fileId」へマッピング。
不明な場合は事前にファイル一覧の取得APIでメタ情報を取得する必要があります。



2.7 Google Driveからファイルを削除

【マップ画面の設定例 -送り先-】

ルートノードは自動的に「変数」に設定され、
リクエスト用のデータがXML変数「C.UserXML」に格納されます。

The screenshot shows the configuration interface for deleting files from Google Drive. The '送り先' (Destination) pane is set to 'RESTClient'. The 'Path' parameter is set to 'field'. The 'プロパティ' (Properties) pane shows the 'RESTClient XMLスキーム' (RESTClient XML Schema) set to '変数' (Variable) with the value 'C.UserXML'. The '変数' (Variable) section below it contains the text '送り元タイプ特性または送り先タイプ特性が変数に設定されている場合は、使用するデータ...' (If the source type characteristic or the destination type characteristic is set to a variable, use the data...). A red box highlights the 'RESTClient' node in the tree, and an arrow points to the '変数' (Variable) section in the properties pane.



OUTPERFORM THE FUTURE™

2.7 Google Driveからファイルを削除

【実行結果】

アップロードファイル



OUTPERFORM THE FUTURE™



第3章 使用例 (kintone REST API 編)



OUTPERFORM THE FUTURE™

3.1

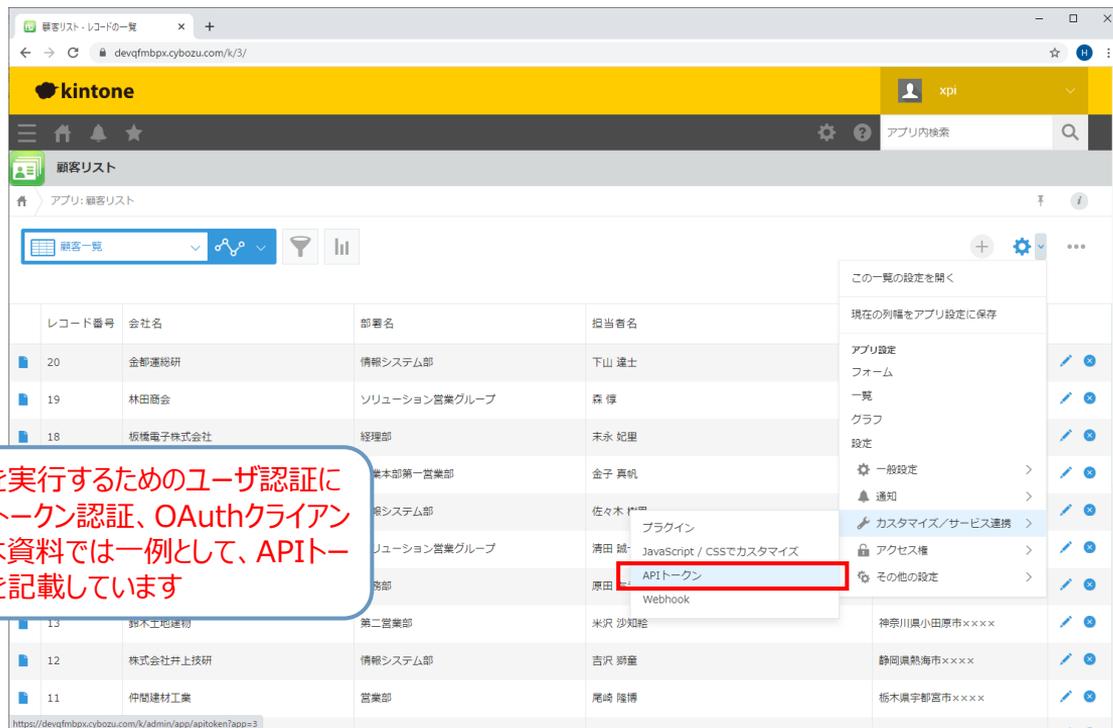
kintone REST API を 使用するための準備



OUTPERFORM THE FUTURE™

3.1 kintone REST API を使用するための準備

- kintone REST API の認証に使用するAPIトークンをアプリごとに生成します。
- アプリ設定ボタンから、「カスタマイズ/サービス連携」 - 「APIトークン」を選択します。



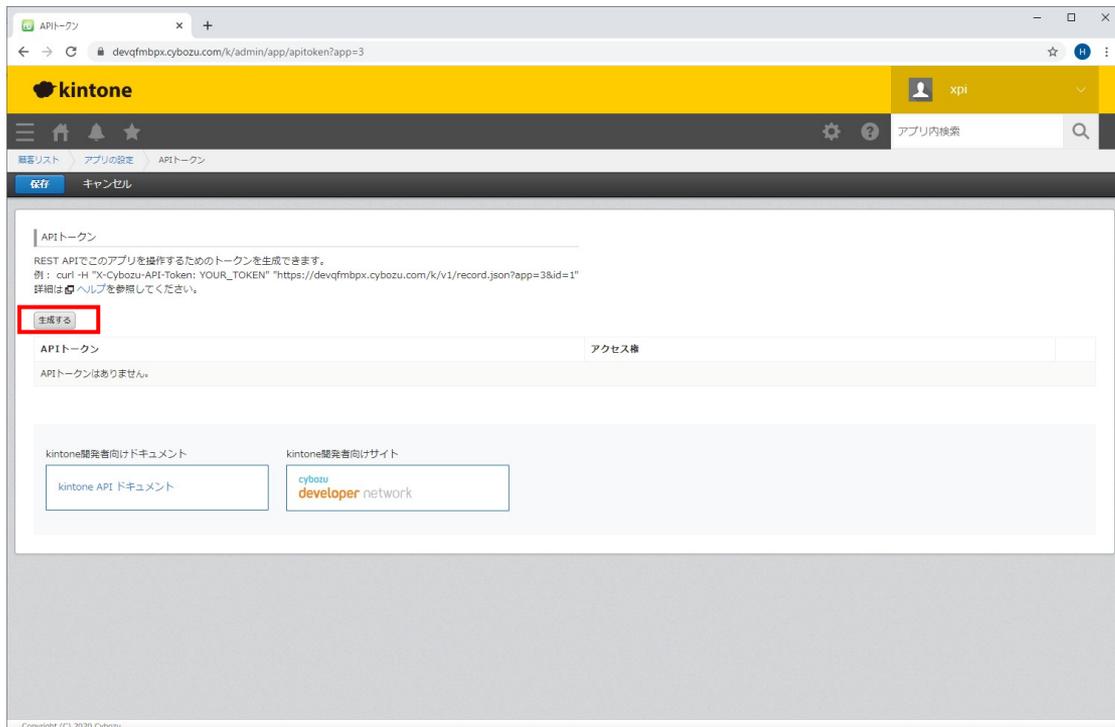
※ kintone REST APIを実行するためのユーザ認証には、パスワード認証やAPIトークン認証、OAuthクライアント認証などがありますが、本資料では一例として、APIトークン認証を使用する方法を記載しています



OUTPERFORM THE FUTURE™

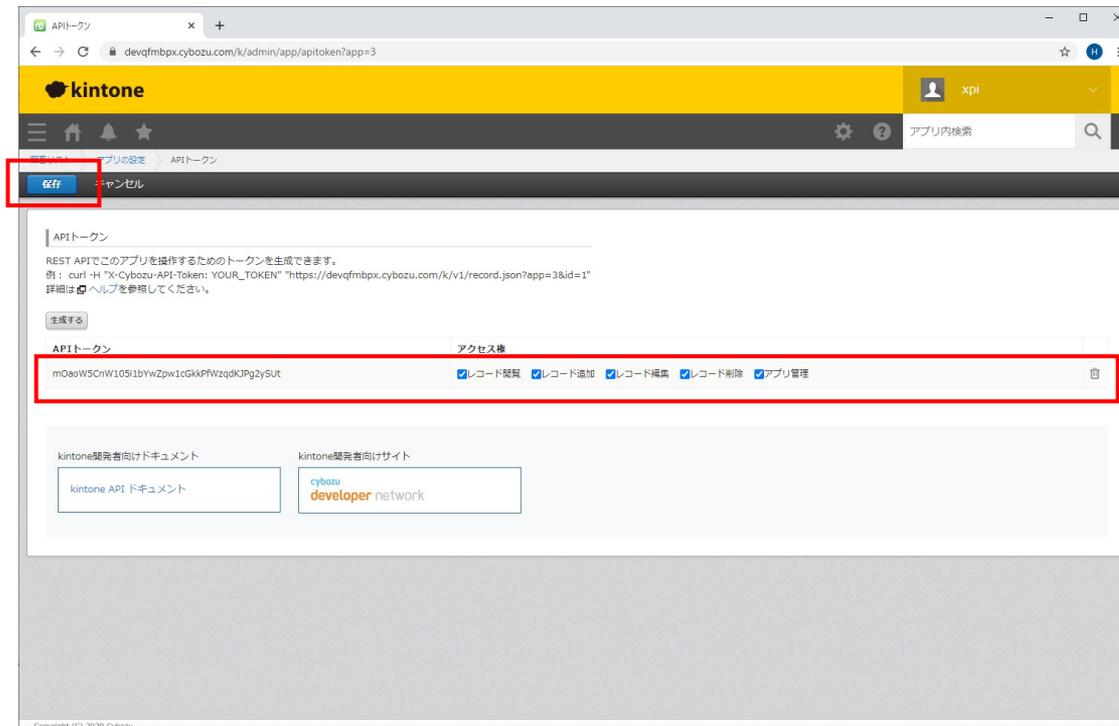
3.1 kintone REST API を使用するための準備

- APIトークン設定画面で、「生成する」ボタンをクリックします。



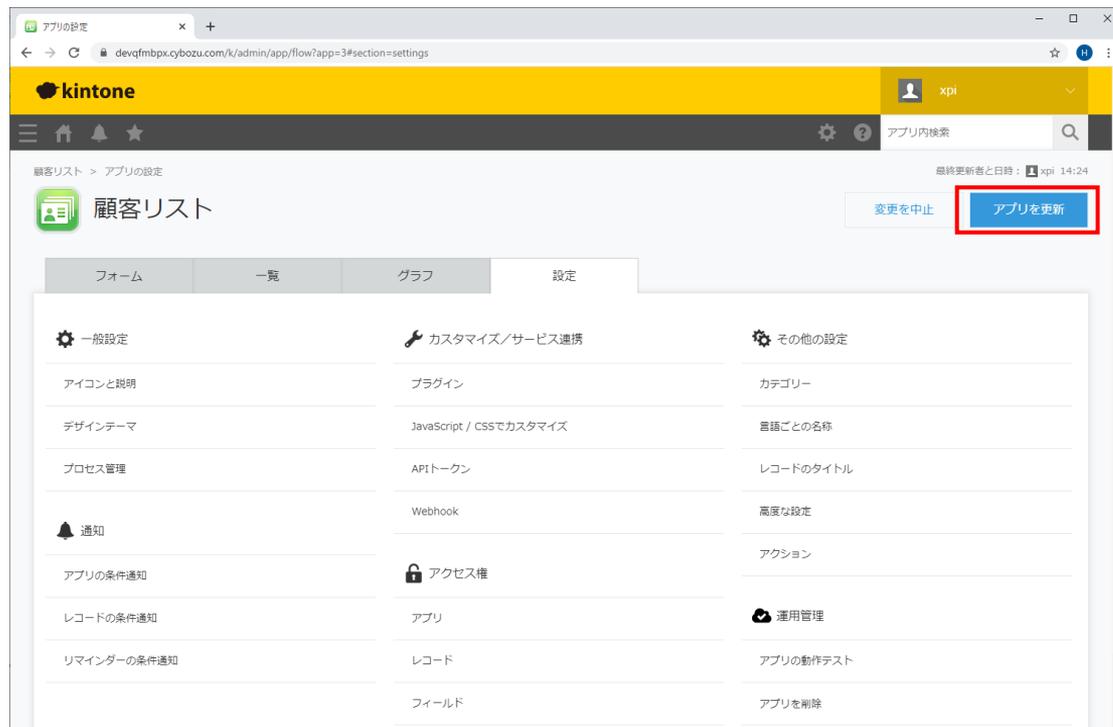
3.1 kintone REST API を使用するための準備

- 生成されたAPIトークンに対して、許可する操作のチェックボックスを選択します。
- 「保存」をクリックします。



3.1 kintone REST API を使用するための準備

- 「アプリを更新」をクリックします。



OUTPERFORM THE FUTURE™

3.1 kintone REST API を使用するための準備

- 設定メニューからアプリ管理画面を開きます。
- 接続先の「アプリのID」を確認しておきます。

アプリ管理

devqfmbpx.cybozu.com/k/admin/app/index

kintone xpi

全体検索

アプリ管理

アプリを作成する

ライセンスの使用状況 アップグレード

	上限	使用中
アプリ数	1000	3
1アプリで1日に実行できるAPIリクエスト数	10000	0
カスタマイズできるアプリ数	無制限	1

※「private」アプリグループに所属するアプリは除かれています。

アプリの一覧 CSV形式でダウンロードする

表示するアプリ 管理権限のあるアプリ (すべて) 1 - 3 / 3件

ID	アプリ名	設定	アプリグループ	ステータス	レコード数	フィールド数	1日のAPIリクエスト数	カスタマイズ	作成者	作成日時	設定の最終更新者	設定の最終更新日時
3	顧客リスト	⚙️	Public	変更あり	20	11	0	なし	橋本浩和	2019-02-14 13:00	xpi	2020-09-18 14:24
1	CustomerData	⚙️	Public	運用中	50	10	0	あり	橋本浩和	2019-02-14 12:58	橋本浩和	2020-09-03 18:02
2	案件管理	⚙️	Public	運用中	20	18	0	なし	橋本浩和	2019-02-14 13:00	橋本浩和	2019-02-14 13:00

Copyright (C) 2020 Cybozu



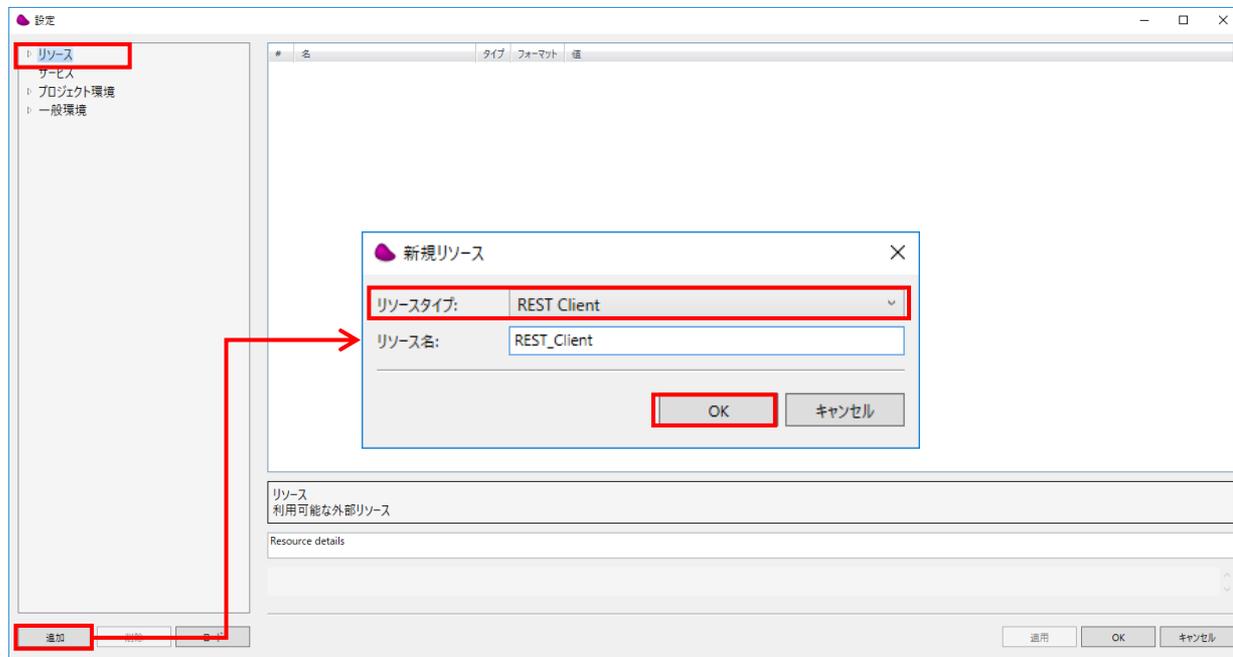
3.2 REST Client リソースの設定



OUTPERFORM THE FUTURE™

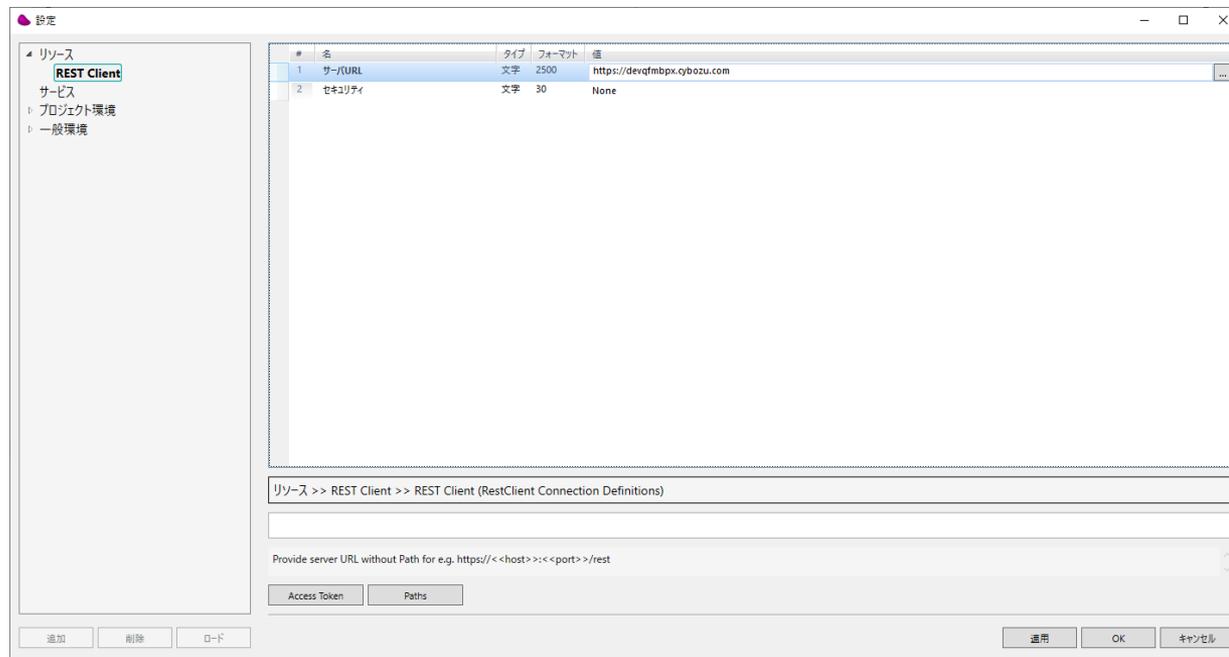
3.2 REST Clientリソースの設定

- ・ xpiスタジオのメニュー > プロジェクト > 設定より設定画面を開きます。
- ・ 左ペインで「リソース」を選択した状態で「追加」ボタンを押し、リソースタイプ「REST Client」のリソースを作成します。



2.2 REST Clientリソースの設定

- ・ サーバURL「https://(サブドメイン名).cybozu.com」を入力し、セキュリティに「None」を選択します。
- ※ APIトークン認証の場合はリクエストヘッダ「X-Cybozu-API-Token」に指定します。



3.3

kintoneからレコード を取得する



OUTPERFORM THE FUTURE™

3.3 kintoneからレコードを取得する

【API仕様】

cybozu developer network (<https://developer.cybozu.io/hc/ja/articles/202331474>)より

- HTTPメソッド

GET

- URI

[https://\(サブドメイン名\).cybozu.com/k/v1/record.json](https://(サブドメイン名).cybozu.com/k/v1/record.json)

- リクエストパラメータ(必須)

app (数値、または文字列)

id (数値、または文字列)

- リクエスト Body

なし

- レスponse Body

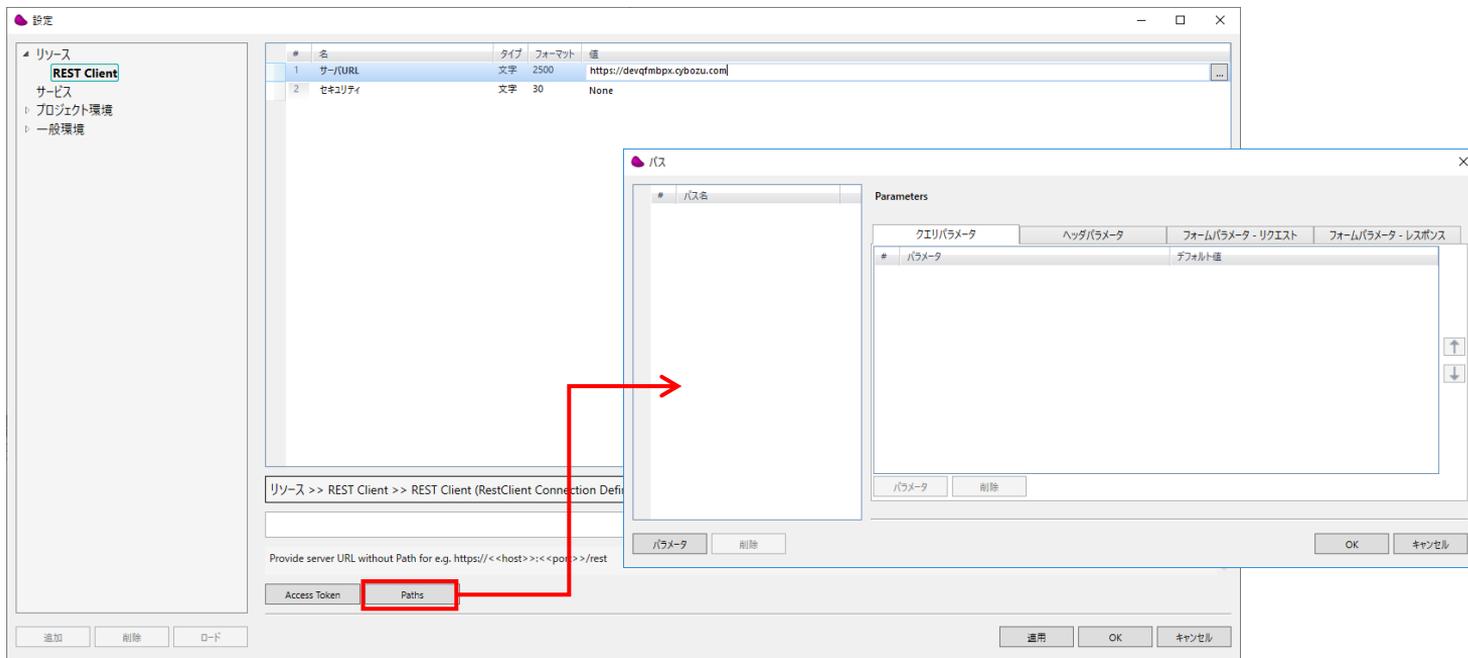
クエリ指定したデータを含む、レコード(JSONで返される)



3.3 kintoneからレコードを取得する

【URIパスとパラメータの設定】

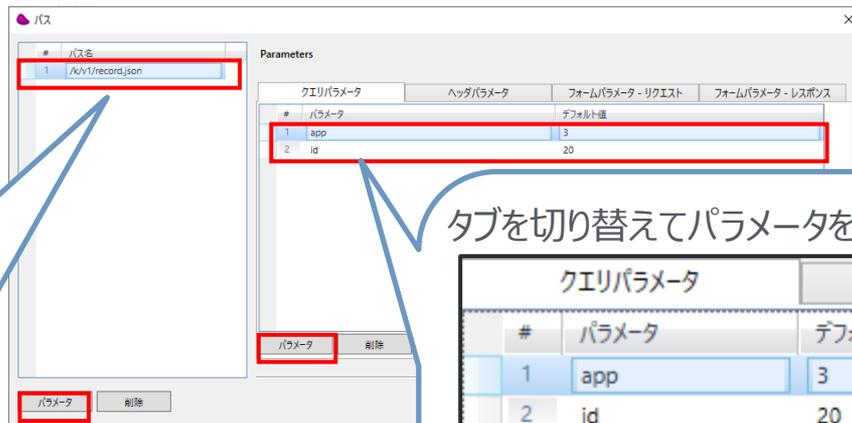
- ・ xpiスタジオのメニュー > プロジェクト > 設定より設定画面を開きます。
- ・ 左ペインで「リソース」> 「REST Client」を選択した状態で「Paths」ボタンを押します。



3.3 kintoneからレコードを取得する

【URIパスとパラメータの設定例】

- ・ 左ペイン下の「パラメータ」ボタンを押して、パスを入力します。
- ・ パスを選択した状態で、右ペイン下の「パラメータ」ボタンを押し、必要なパラメータを入力します。



サーバURLを除いた(APIバージョンやリソースを識別する部分)パスを指定します。ここでは、「/k/v1/record.json」

タブを切り替えてパラメータを設定します

クエリパラメータ		ヘッダパラメータ	
#	パラメータ	デフォルト値	
1	app	3	
2	id	20	

クエリパラメータ		ヘッダパラメータ		フォームパラメータ - リク	
#	Name	Default Value			
1	X-Cybozu-API-Token	mOaoW5CnW105i1bYwZpw1cGkkPfWzqdKJPg2ySUT			



OUTPERFORM THE FUTURE™

3.3 kintoneからレコードを取得する

【ステップの設定例】

項目	値
リソース名	プロパティペインでREST Clientリソースを選択。
パス	リソース設定のリストからパスを選択。
オペレーション	「GET」を選択。
レスポンス Content-Type	ボタンを押して「application/json」を選択
結果保存	「変数」を選択し、結果を格納する変数を指定。 APIから戻されるJSONデータが格納されます
ステータスコード	HTTPステータスを格納する変数を指定
メッセージ	HTTPレスポンスメッセージを格納する変数を指定
成功	「Variable」を選択し、成否を格納する変数を指定。 成功時は「True」、失敗時は「False」が格納されます。

※ 前頁のパラメータ値は、「パラメータ」ボタンで設定・変更します。(設定値を変数にすることで動的に反映)



3.3 kintoneからレコードを取得する

【マップ画面の設定例】

ルートノードは自動的に「変数」に設定され、
リクエスト用のデータがXML変数「C.UserXML」に格納されます。

クエリパラメータとしてURI文字列
に含めた形式でリクエストするた
め、ここでは何も指定しません。

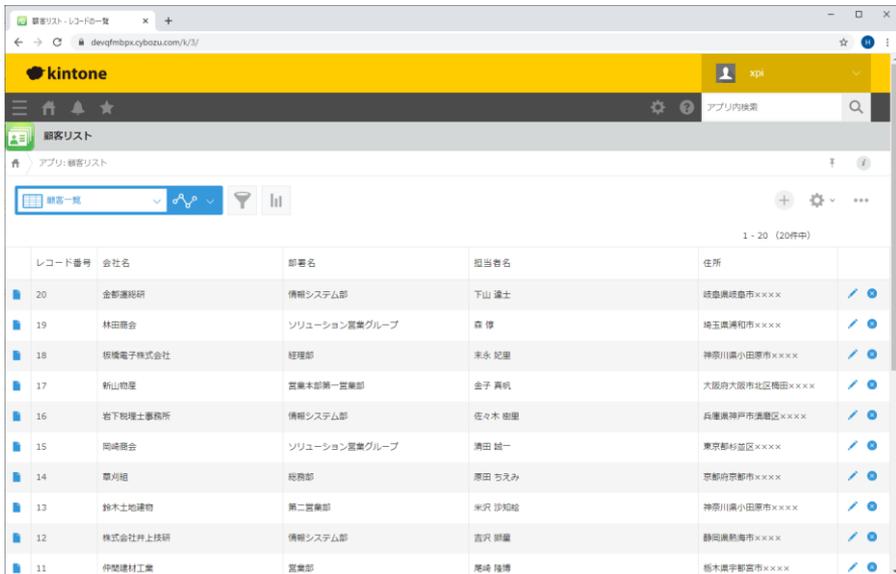


OUTPERFORM THE FUTURE™

3.3 kintoneからレコードを取得する

【実行結果】

kintoneアプリ



The screenshot shows a Kintone application titled '顧客リスト' (Customer List). The interface includes a search bar, filters, and a table of records. The table has columns for Record ID, Company Name, Department Name, Representative Name, and Address. The records are sorted by ID in descending order.

レコード番号	会社名	部署名	担当名	住所
20	金都運総研	情報システム部	下山 謙士	岐阜県岐阜市xxxx
19	林田商会	ソリューション営業グループ	森 博	埼玉県浦和市xxxx
18	板橋電子株式会社	経理部	末永 紀雄	神奈川県小田原市xxxx
17	新山電産	営業本部第一営業部	金子 真帆	大阪府大阪市北区梅田xxxx
16	岩下知理士事務所	情報システム部	佐々木 悠里	兵庫県神戸市灘区xxxx
15	岡崎商会	ソリューション営業グループ	渡田 雄一	東京都杉並区xxxx
14	藤川組	総務部	原田 ちえみ	京都府京都市xxxx
13	鈴木土地建物	第二営業部	米沢 沙知絵	神奈川県小田原市xxxx
12	株式会社井上技研	情報システム部	吉沢 朝慶	静岡県熱海市xxxx
11	仲間建材工業	営業部	尾崎 隆博	栃木県宇都宮市xxxx

レスポンスデータ

```
{
  "record": {
    "備考": {
      "type": "MULTI_LINE_TEXT",
      "value": ""
    },
    "レコード番号": {
      "type": "RECORD_NUMBER",
      "value": "20"
    },
    "更新者": {
      "type": "USER",
      "value": {
        "code": "090-xxxx-xxxx",
        "name": "yamashita"
      }
    },
    "作成者": {
      "type": "USER",
      "value": {
        "code": "050-xxxx-xxxx",
        "name": "kimura"
      }
    },
    "作成日時": {
      "type": "CREATED_TIME",
      "value": "2019-02-14T04:00:00Z"
    },
    "会社名": {
      "type": "SINGLE_LINE_TEXT",
      "value": "金都運総研"
    },
    "$id": {
      "type": "__ID__",
      "value": "20"
    }
  }
}
```



OUTPERFORM THE FUTURE™

3.4

kintoneへレコード を登録する



OUTPERFORM THE FUTURE™

3.4 kintoneへレコードを登録する

【API仕様】

cybozu developer network (<https://developer.cybozu.io/hc/ja/articles/202166160>)より

- HTTPメソッド

POST

- URI

[https://\(サブドメイン名\).cybozu.com/k/v1/record.json](https://(サブドメイン名).cybozu.com/k/v1/record.json)

- リクエストパラメータ(必須)

app (数値、または文字列)

- リクエスト Body

レコード情報(フィールドコードとフィールドの値)をJSON形式で指定

- レスポンス Body

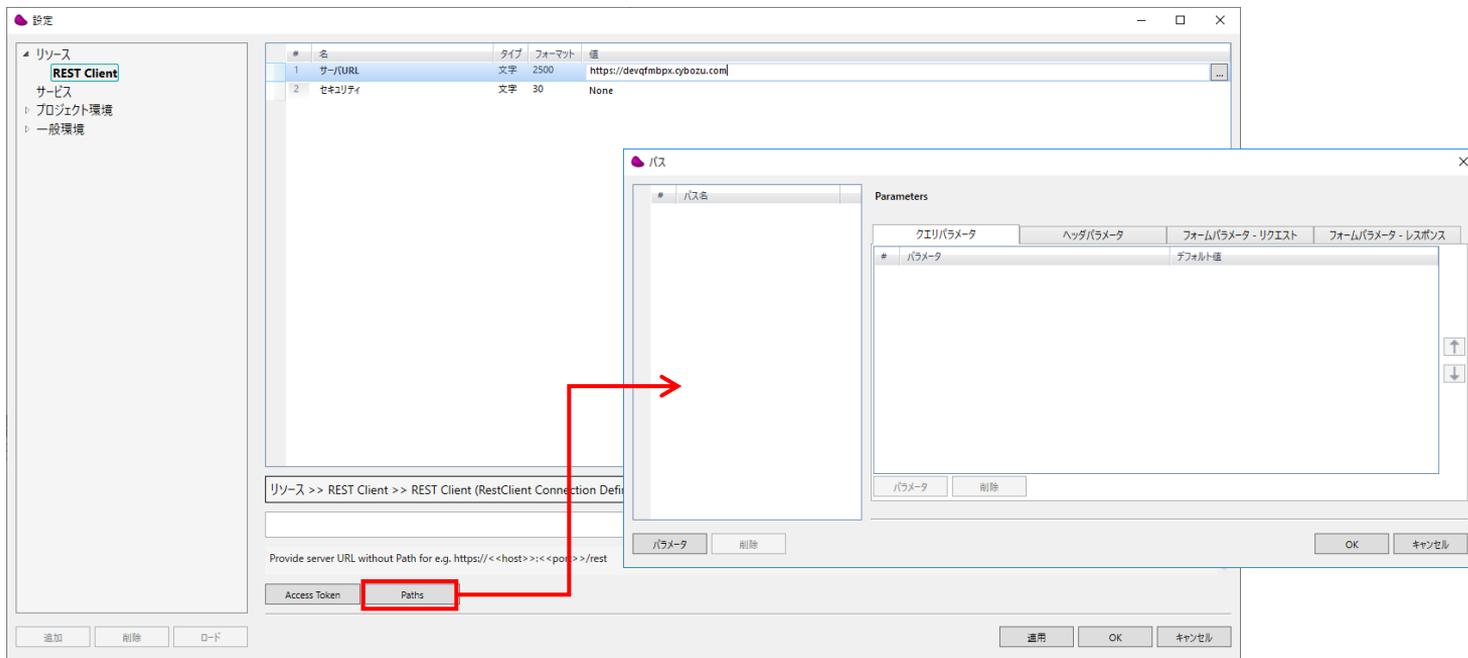
登録したレコードのレコードIDとリビジョン(JSONで返される)



3.4 kintoneへレコードを登録する

【URIパスとパラメータの設定】

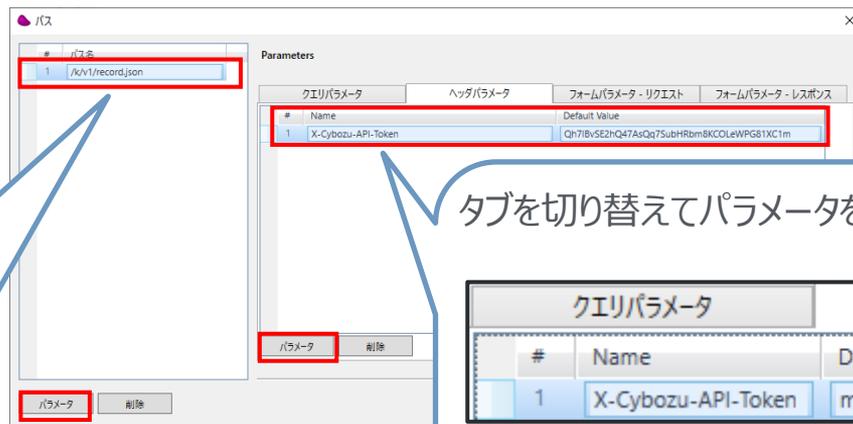
- ・ xpiスタジオのメニュー > プロジェクト > 設定より設定画面を開きます。
- ・ 左ペインで「リソース」> 「REST Client」を選択した状態で「Paths」ボタンを押します。



3.4 kintoneへレコードを登録する

【URIパスとパラメータの設定例】

- ・ 左ペイン下の「パラメータ」ボタンを押して、パスを入力します。
- ・ パスを選択した状態で、右ペイン下の「パラメータ」ボタンを押し、必要なパラメータを入力します。



3.4 kintoneへレコードを登録する

【ステップの設定例】

項目	値
リソース名	プロパティペインでREST Clientリソースを選択。
パス	リソース設定のリストからパスを選択。
オペレーション	「POST」を選択。
リクエスト Content-Type	ボタンを押して「application/json」を選択
レスポンス Content-Type	ボタンを押して「application/json」を選択
結果保存	「変数」を選択し、結果を格納する変数を指定。 APIから戻されるJSONデータが格納されます
ステータスコード	HTTPステータスを格納する変数を指定
メッセージ	HTTPレスポンスメッセージを格納する変数を指定
成功	「Variable」を選択し、成否を格納する変数を指定。 成功時は「True」、失敗時は「False」が格納されます。



3.4 kintoneへレコードを登録する

【マッパー画面の設定例】

ルートノードは自動的に「変数」に設定され、
リクエスト用のデータがXML変数「C.UserXML」に格納されます。

登録するレコードをJSON形式で指定
※予め、前ステップにてリクエスト用のJSONデータを生成し、値をマッピングさせることもできます



3.4 kintoneへレコードを登録する

【実行結果】

リクエストデータ

```
{
  "app": "3",
  "record": {
    "備考": {
      "value": ""
    },
    "メールアドレス": {
      "value": "magic_xpi@example.com"
    },
    "顧客ランク": {
      "value": "A"
    },
    "住所": {
      "value": "愛知県名古屋市xxxx"
    },
    "TEL": {
      "value": "090-xxxx-xxxx"
    },
    "FAX": {
      "value": "050-xxxx-xxxx"
    },
    "会社名": {
      "value": "マジックソフトウェア"
    }
  }
}
```

kintoneアプリ

レコード番号	会社名	部署名	担当者名	住所
22	マジックソフトウェア	名古屋営業所	魔術 印手肇	愛知県名古屋市xxxx
21	試栄興業	情報システム部	上川 崇雄	愛知県名古屋市xxxx
20	金都運研研	情報システム部	下山 達士	岐阜県岐阜市xxxx
19	林田商会	ソリューション営業グループ	森 尊	埼玉県浦和市xxxx
18	株式会社 株式会社	経理部	末永 紀雄	神奈川県小田原市xxxx
17	新山物産	営業本部第一営業部	金子 貴机	大阪府大阪市北区梅田xxxx
16	若下税理士事務所	情報システム部	佐々木 樹里	兵庫県神戸市須磨区xxxx
15	岡崎商会	ソリューション営業グループ	満田 誠一	東京都杉並区xxxx
14	豊川組	総務部	原田 ちえみ	京都府京都市xxxx
13	鈴木土地建物	第二営業部	米沢 沙知絵	神奈川県小田原市xxxx



OUTPERFORM THE FUTURE™

3.5

kintoneのレコード を更新する



OUTPERFORM THE FUTURE™

3.5 kintoneのレコードを更新する

【API仕様】

cybozu developer network (<https://developer.cybozu.io/hc/ja/articles/201941784>)より

- HTTPメソッド

PUT

- URI

[https://\(サブドメイン名\).cybozu.com/k/v1/record.json](https://(サブドメイン名).cybozu.com/k/v1/record.json)

- リクエストパラメータ(必須)

app (数値、または文字列)

id (数値、または文字列)

- リクエスト Body

更新するレコード情報(フィールドコードとフィールドの値)をJSON形式で指定

- レスポンス Body

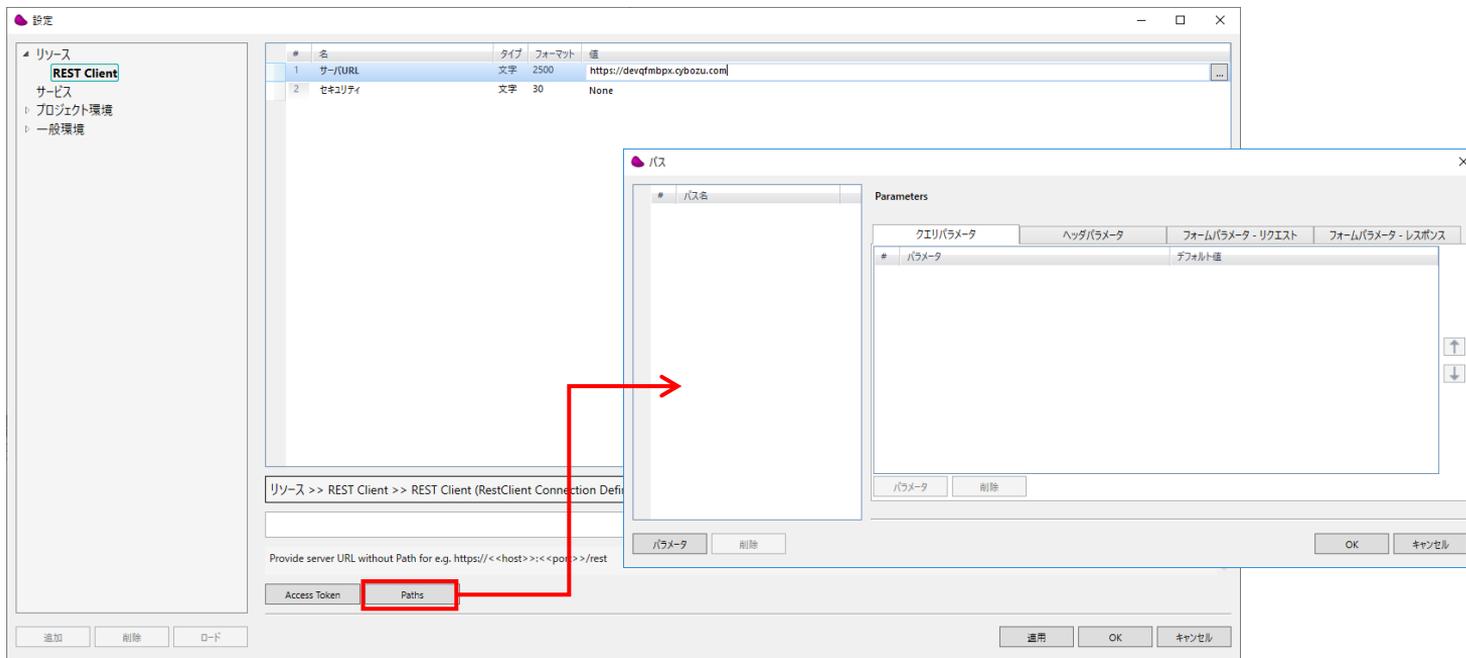
更新したレコードのリビジョン(JSONで返される)



3.5 kintoneのレコードを更新する

【URIパスとパラメータの設定】

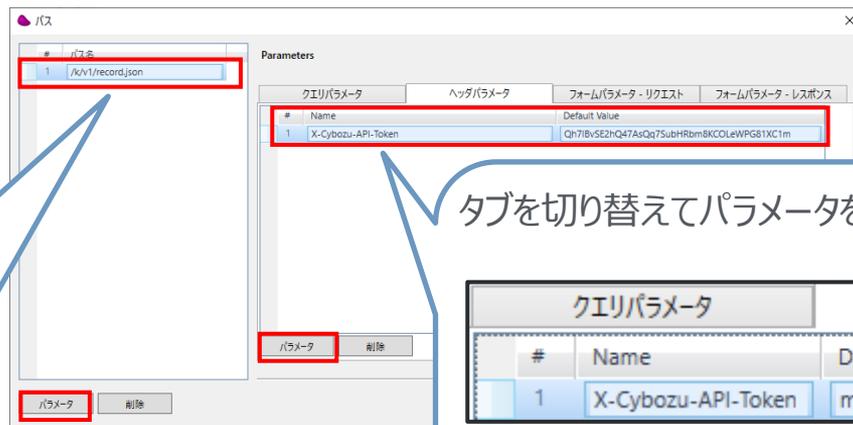
- ・ xpiスタジオのメニュー > プロジェクト > 設定より設定画面を開きます。
- ・ 左ペインで「リソース」> 「REST Client」を選択した状態で「Paths」ボタンを押します。



3.5 kintoneのレコードを更新する

【URIパスとパラメータの設定例】

- ・ 左ペイン下の「パラメータ」ボタンを押して、パスを入力します。
- ・ パスを選択した状態で、右ペイン下の「パラメータ」ボタンを押し、必要なパラメータを入力します。



3.5 kintoneのレコードを更新する

【ステップの設定例】

REST Client Configuration

接続
リソース名: REST Client

オペレーション
パス: /k/v1/record.json
オペレーション: Put
セキュリティ: Yes
リクエスト Content Type: application/json
レスポンス Content Type: application/json
要求エンコーディング: Keep existing Charset:

出力
結果保存: Variable C.UserBlob
ヘッダフィルタ:
ヘッダ:
ステータスコード: C.UserCode
メッセージ: C.UserString
成功: Variable F.Success

パラメータ スキーマ更新 OK キャンセル

項目	値
リソース名	プロパティペインでREST Clientリソースを選択。
パス	リソース設定のリストからパスを選択。
オペレーション	「PUT」を選択。
リクエスト Content-Type	ボタンを押して「application/json」を選択
レスポンス Content-Type	ボタンを押して「application/json」を選択
結果保存	「変数」を選択し、結果を格納する変数を指定。APIから戻されるJSONデータが格納されます
ステータスコード	HTTPステータスを格納する変数を指定
メッセージ	HTTPレスポンスメッセージを格納する変数を指定
成功	「Variable」を選択し、成否を格納する変数を指定。成功時は「True」、失敗時は「False」が格納されます。



3.5 kintoneのレコードを更新する

【マップパー画面の設定例】

ルートノードは自動的に「変数」に設定され、
リクエスト用のデータがXML変数「C.UserXML」に格納されます。

更新するレコードID、フィールド情報をJSON形式で指定
※予め、前ステップにてリクエスト用のJSONデータを生成し、値をマッピングさせることもできます



OUTPERFORM THE FUTURE™

3.5 kintoneのレコードを更新する

【実行結果】

リクエストデータ

```
{  
  "app": "3",  
  "id": "22",  
  "record": {  
    "担当者名": {  
      "type": "SINGLE_LINE_TEXT",  
      "value": "更新 島舌"  
    }  
  }  
}
```

kintoneアプリ

The screenshot shows the Kintone application interface with a list of records. The record with ID 22 is highlighted with a red box, indicating it was successfully updated. The record details are as follows:

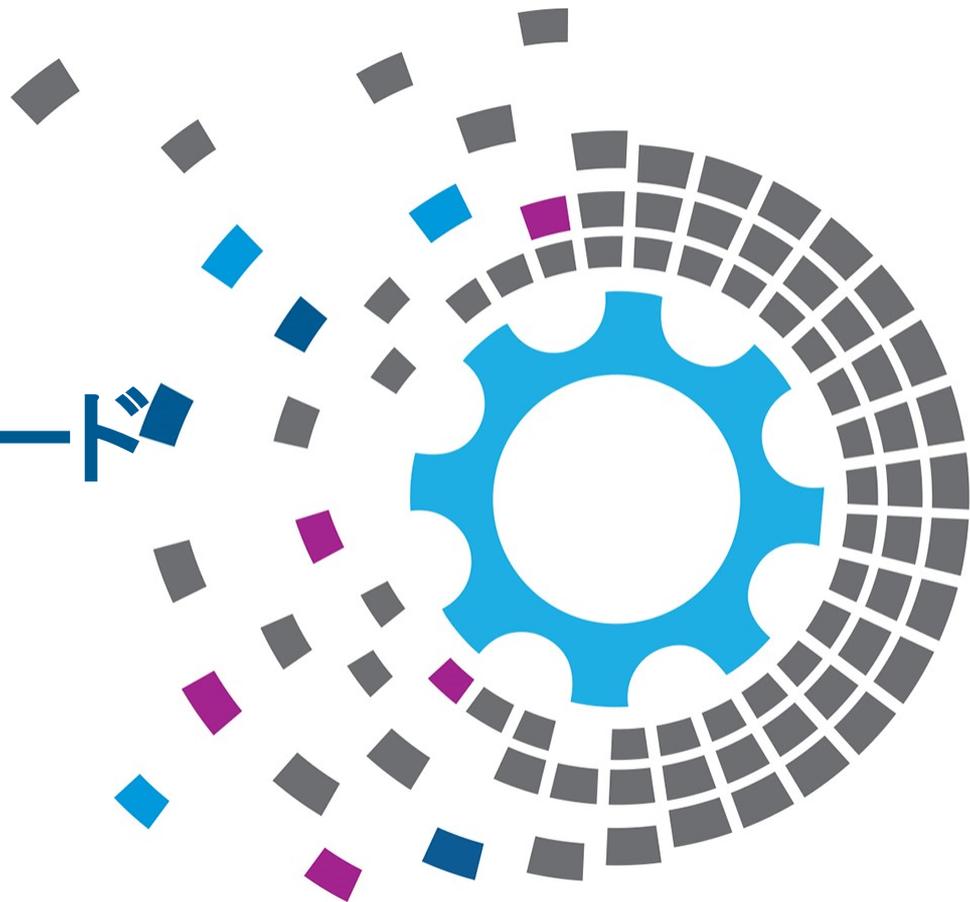
レコード番号	会社名	部署名	担当者名	住所
22	マジックソフトウェア	名古屋営業所	更新 島舌	愛知県名古屋××××
21	試栄興業	コンサル部	上川 英雄	愛知県名古屋××××
20	金都運研研	情報システム部	下山 達士	岐阜県岐阜××××
19	林田商会	ソリューション営業グループ	森 尊	埼玉県浦和××××
18	株式会社 株式会社	経理部	木永 紀里	神奈川県小田原市××××
17	新山物産	営業本部第一営業部	金子 貴帆	大阪府大阪市北区梅田××××
16	若下税理士事務所	情報システム部	佐々木 樹里	兵庫県神戸市灘区××××
15	岡崎商会	ソリューション営業グループ	満田 誠一	東京都杉並区××××
14	翠川組	総務部	原田 ちえみ	京都府京都市××××
13	鈴木土地建物	第二営業部	米沢 沙知絵	神奈川県小田原市××××



OUTPERFORM THE FUTURE™

3.6

kintoneからレコード を削除する



OUTPERFORM THE FUTURE™

3.6 kintoneからレコードを削除する

【API仕様】

cybozu developer network (<https://developer.cybozu.io/hc/ja/articles/201941794>)より

- HTTPメソッド
DELETE
- URI
`https://(サブドメイン名).cybozu.com/k/v1/records.json`
- リクエストパラメータ(必須)
app (数値、または文字列)
ids (数値配列)
- リクエスト Body
なし(クエリパラメータ指定時) or 削除するレコード番号をJSON形式で指定
- レスponse Body



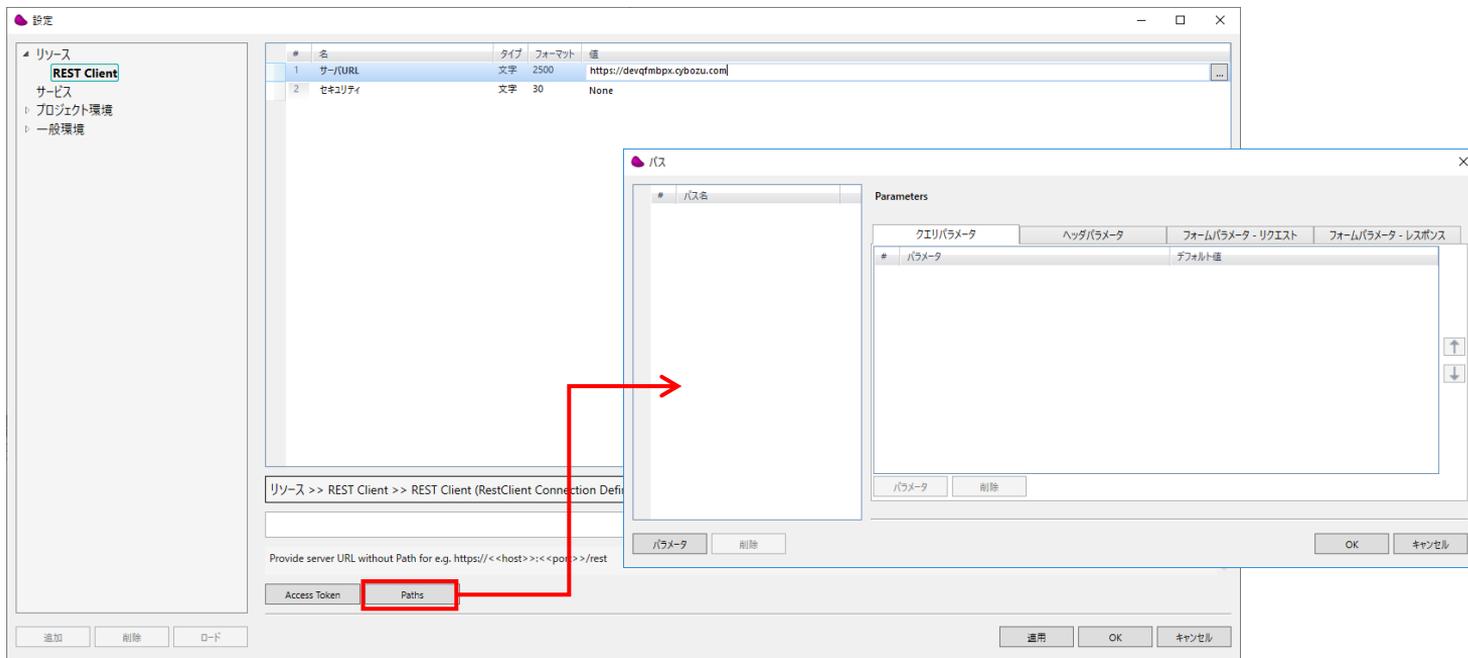
OUTPERFORM THE FUTURE™

空のJSONデータ

3.6 kintoneからレコードを削除する

【URIパスとパラメータの設定】

- ・ xpiスタジオのメニュー > プロジェクト > 設定より設定画面を開きます。
- ・ 左ペインで「リソース」> 「REST Client」を選択した状態で「Paths」ボタンを押します。

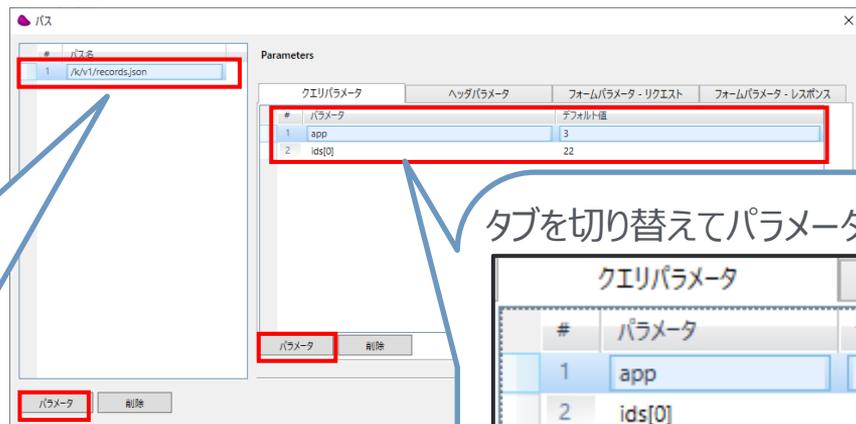


3.6 kintoneからレコードを削除する

【URIパスとパラメータの設定例】

- ・ 左ペイン下の「パラメータ」ボタンを押して、パスを入力します。
- ・ パスを選択した状態で、右ペイン下の「パラメータ」ボタンを押し、必要なパラメータを入力します。

サーバURLを除いた(APIバージョンやリソースを識別する部分)パスを指定します。ここでは、「 /k/v1/records.json 」



タブを切り替えてパラメータを設定します

クエリパラメータ		ヘッダパラメータ	
#	パラメータ	デフォルト値	
1	app	3	
2	ids[0]	22	

クエリパラメータ		ヘッダパラメータ		フォームパラメータ - リクエスト	
#	Name	Default Value			
1	X-Cybozu-API-Token	mOaoW5CnW105i1bYwZpw1cGkkPfWzqdKJPg2ySUT			



3.6 kintoneからレコードを削除する

【ステップの設定例】

REST Client Configuration

接続
リソース名: REST Client

オペレーション
パス: /k/v1/records.json
オペレーション: Delete

セキュリティ: Yes

リクエスト Content Type: ...

レスポンス Content Type: ...

要求エンコーディング: Keep existing Charset: ...

出力
結果保存: Variable C.UserBlob
ヘッダフィルタ: ...
ヘッダ: ...
ステータスコード: C.UserCode
メッセージ: C.UserString
成功: Variable F.Success

パラメータ スキーマ更新 OK キャンセル

項目	値
リソース名	プロパティペインでREST Clientリソースを選択。
パス	リソース設定のリストからパスを選択。
オペレーション	「GET」を選択。
結果保存	「変数」を選択し、結果を格納する変数を指定。 APIから戻されるJSONデータが格納されます
ステータスコード	HTTPステータスを格納する変数を指定
メッセージ	HTTPレスポンスメッセージを格納する変数を指定
成功	「Variable」を選択し、成否を格納する変数を指定。 成功時は「True」、失敗時は「False」が格納されます。

※ 前頁のパラメータ値は、「パラメータ」ボタンで設定・変更します。(設定値を変数にすることで動的に反映)



3.6 kintoneからレコードを削除する

【マップ画面の設定例】

ルートノードは自動的に「変数」に設定され、
リクエスト用のデータがXML変数「C.UserXML」に格納されます。

クエリパラメータに指定する場合、リクエスト用のJSONデータは不要なため、空文字を指定します。

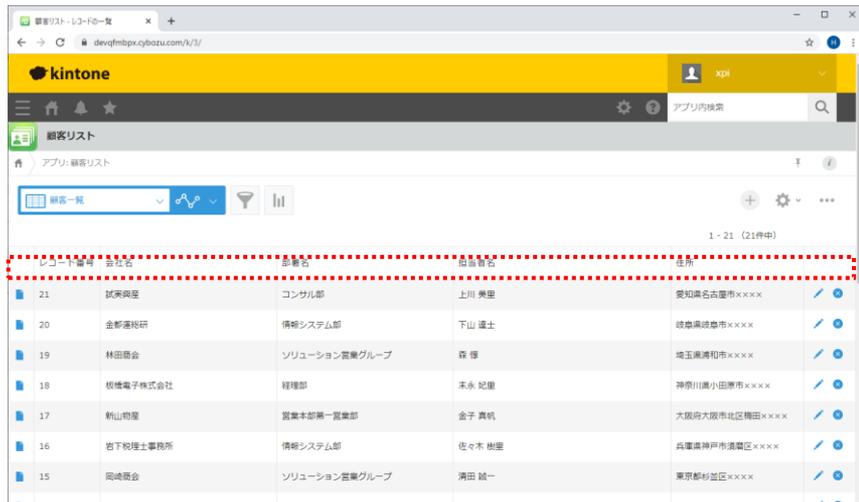


OUTPERFORM THE FUTURE™

3.6 kintoneからレコードを削除する

【実行結果】

レスポンスデータ



The screenshot shows a Kintone application interface with a table of records. The table has columns for 'レコード番号' (Record Number), '会社名' (Company Name), '部署名' (Department Name), '担当者名' (Staff Name), and '住所' (Address). The first two rows are highlighted with a red dashed box.

レコード番号	会社名	部署名	担当者名	住所
21	試栄商店	コンサル部	上川 美里	愛知県名古屋市xxxx
20	金野建設	情報システム部	下山 達士	岐阜県岐阜市xxxx
19	林田商会	ソリューション営業グループ	森 博	埼玉県浦和市xxxx
18	板橋電子株式会社	経理部	末永 紀里	神奈川県小田原市xxxx
17	新山物産	営業本部第一営業部	金子 真帆	大阪府大阪市北区鶴田xxxx
16	岩下税理士事務所	情報システム部	佐々木 悠里	兵庫県神戸市須磨区xxxx
15	岡崎商会	ソリューション営業グループ	濱田 結一	東京都杉並区xxxx



OUTPERFORM THE FUTURE™

3.7

リクエスト/レスポンス データの扱い方



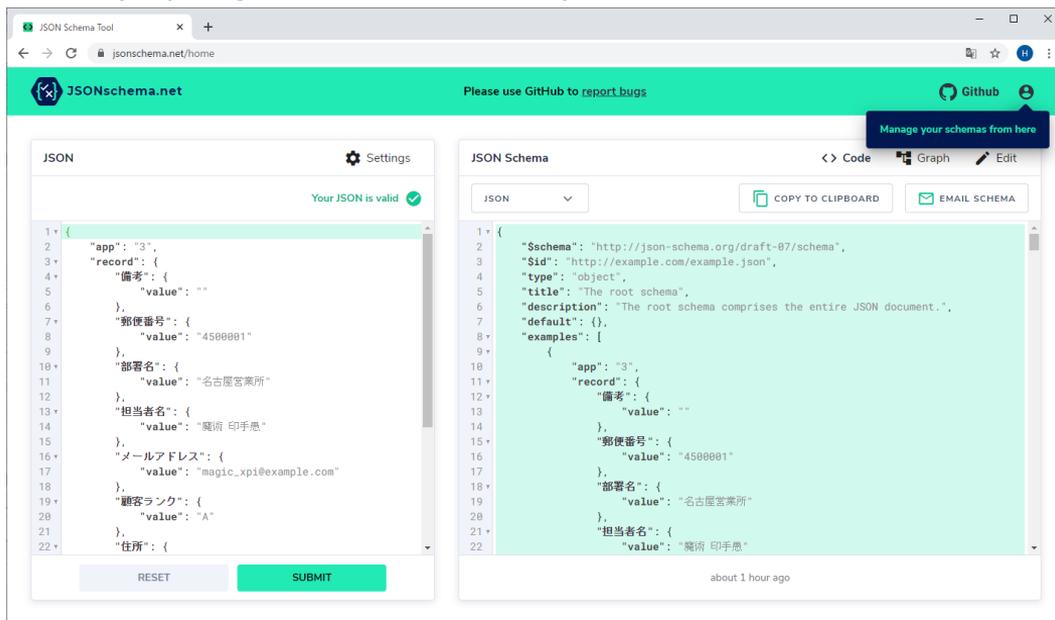
OUTPERFORM THE FUTURE™

3.7 リクエスト/レスポンスデータの扱い方

【JSONスキーマの作成】

- ・ リクエスト/レスポンスのサンプルデータを元にツール等を利用してJSONスキーマを作成します。
- ・ 作成したJSONスキーマは、プロジェクトフォルダへ保存します。

※ JSON schema Tool (<https://jsonschema.net/home>)の例



The screenshot displays the JSON Schema Tool interface. On the left, a JSON document is shown with a 'Your JSON is valid' status. On the right, the generated JSON Schema is displayed, including fields like 'app', 'record', '郵便番号', '部署名', '担当者名', 'メールアドレス', '顧客ランク', and '住所'.

```
1 {
2   "app": "3",
3   "record": {
4     "備考": {
5       "value": ""
6     },
7     "郵便番号": {
8       "value": "4500001"
9     },
10    "部署名": {
11      "value": "名古屋営業所"
12    },
13    "担当者名": {
14      "value": "魔術 印手愚"
15    },
16    "メールアドレス": {
17      "value": "magic_xpi@example1e.com"
18    },
19    "顧客ランク": {
20      "value": "A"
21    },
22    "住所": {
```

```
1 {
2   "$schema": "http://json-schema.org/draft-07/schema",
3   "$id": "http://example.com/example.json",
4   "type": "object",
5   "title": "The root schema",
6   "description": "The root schema comprises the entire JSON document.",
7   "default": {},
8   "examples": [
9     {
10      "app": "3",
11      "record": {
12        "備考": {
13          "value": ""
14        },
15        "郵便番号": {
16          "value": "4500001"
17        },
18        "部署名": {
19          "value": "名古屋営業所"
20        },
21        "担当者名": {
22          "value": "魔術 印手愚"
```



3.7 リクエスト/レスポンスデータの扱い方

【リクエストデータの設定例】

- 前ステップでデータマッパーを使用し、リクエストBODYとなるJSONデータを作成します。
下記の例では、CSVファイルからレコード登録用データを設定しています。

事前に作成・保存したJSONスキーマを指定します。
データはBlob型の変数に格納し、REST Clientコンポーネントのマッパー画面でRequestBodyのDataBlobへ指定します。



3.7 リクエスト/レスポンスデータの扱い方

【レスポンスデータの設定例】

- 後ステップでデータマッパーを使用し、レスポンスデータから必要なデータを抜き出します。下記の例では、レコード取得データからCSVファイルに書き出しています。

The screenshot shows the 'データマッパー (フロー-6)' configuration window. On the left, the '送り元' (Source) pane shows a tree structure for 'JSON_1' with fields like 'レコード番号' (Record ID), '郵便番号' (Postal Code), '会社名' (Company Name), etc. On the right, the '送り先' (Destination) pane shows a 'FlatFile-2' configuration with corresponding fields. A red box highlights 'JSON_1' in the source pane, and a red arrow points to the 'FlatFile-2' pane. To the right, the 'プロパティ' (Properties) window for 'JSON_1 JSONスキーマファイル' is shown, with a red box around the 'JSONスキーマファイル' property set to 'schema#response.json' and the '変数' (Variable) property set to 'C.UserBlob'.

事前に作成・保存したJSONスキーマを指定します。
データはREST Clientコンポーネントのステップ設定で指定した結果保存先の変数を指定します。



THANK YOU!



OUTPERFORM THE FUTURE™

