

IoT Platform
API リファレンス
(14.0_0 版)

< 改版履歴 >

版数	説明	日付
1.0 版	初版	2016/10/7
1.1 版	注釈の番号を振り直し(欠落を解消) 1.2 節にダイナミックリソースコントローラー(DRC)の参照先を記載 2.2 節を 2.2.1~2.2.5 に分割し、2.2.6 を追加 2.2 節に記載していた「リソース_Binary」に対する制約事項を削除 日本時間記載を全て UTC に変更 時刻凡例の「+0900」を全て「Z」に変更 2.3.1 節、2.4.1 節 SSL 通信にあたって TLS1.1 または TLS1.2 を使用するべきであることを記載 2.5.1 節 MQTT における QoS に関する記載の追記	2016/11/1
1.2 版	誤記の修正 リソースの検索における \$orderby の記述を削除 2.1 節「各パラメータの最大・最小値」に関する追記 3.2 節 エラー応答追記 7.2 節 Body 文における last_modified の日時に関する記載追記	2016/12/1
1.3 版	2.1 節 リソースパスの利用可能文字について修正	2017/1/12
4_1.0 版	機能エンハンス 2.3.1 節、2.3.2 節 部分取得機能(Range Request)を追加 2.4 節「CORS(Cross-Domain Resource Sharing)対応」を追加 3.3 節、5.3 節 hierarchy_get 権の利用方法を追加 8.1 節、8.2 節 クライアント証明書関連事項を追加 8.1 節 hierarchy_get 権、hierarchy_put 権を追加	2017/4/1
4_2.0 版	機能エンハンス 2.4 節、8.1 節、8.2 節 17 年夏以降エンハンスに関する記載を削除 2.2.7 節 通信頻度に対するシステム制限について を追加 第 11 章 システム格納データリファレンス を追加	2017/8/1
5.0_0 版	機能エンハンス 3.1 節、3.4 節 csv/テキスト/バイナリ形式のデータ登録、およびデータ登録時の圧縮送信を追加 3.3 節 配列形式に対する検索を追加 機能エンハンスに伴う記載内容の修正(2.1 節、付録 1)	2017/11/18
5.1_0 版	機能エンハンス 2.1 節、2.4.1 節、3.1 節、3.4 節、3.5 節、4.1 節、11.3 節 要求とエラー情報との関連づけ機能を追加 2.1 節 skip 件数の最小値を変更 3.3 節 検索結果の並び替え機能を追加 2.3.1 節、2.5.1 節 サポートする TLS バージョンの変更 補記追加 3.1 節、3.2 節、3.3 節、3.4 節、3.5 節 クエリに対する補記を追加	2018/3/17
5.1_1 版	誤記の修正 2.1 節 誤記の削除 11.3 節 誤記の修正	2018/4/11
5.1_2 版	記載の追加 第 10 章 章タイトルの変更 10.1 節 Request における注意事項、Response における要求仕様を追加 10.3 節 リソース(転送)_JSON の API 呼出し仕様を追加	2018/6/13
7.0_0 版	機能エンハンス 2.3.1 節、2.3.2 節、2.4.1 節、2.4.2 節 データ取得時の圧縮通信機能を追加	2018/9/15

	2.4.1 節、5.1 節、5.5 節 要求とエラー情報との関連づけ機能を追加 10.1 節 リーズンフレーズの要求を削除 11.3 節 失敗要因メッセージを追加	
9.0_0 版	機能エンハンス 変更なし	2019/8/22
11.0_0 版	仕様変更に関する追記 2.3.2 節 HTTP レスポンスの reason-phrase に関する注意書きを追記 以下の API 実行時は reason-phrase 付与されません。 第 5 章 リソース_Binary データ制御 (REST) 記載の全 API 第 7 章 リソース制御 (REST) 記載の全 API 第 8 章 アクセスコード制御 (REST) 記載の全 API 第 9 章 イベント制御 (REST) 記載の全 API	2021/1/16
14.0_0 版	機能エンハンス 変更なし	2022/4/1

はじめに

このたびは「IoT Platformサービス（以下、本サービス）」をご検討いただき、誠にありがとうございます。

この「IoT Platform APIリファレンス（以下、本書）」は、本サービスをご契約いただいたお客様、またはご採用いただくお客様のための資料です。下記につきましてご了承くださいませよう、よろしくお願いいたします。

1. ご検討中のお客様は、本書を本サービスご採用可否のご判断のためにのみ、ご参照願います。
2. 本書および本書の内容について、第三者へご開示、ご提供にはならないようお願いいたします。
3. 発行元の許可なく、本書の記載内容を複製、転写することを禁止します。

本書には本サービスをご採用いただくための重要な情報を記載しています。

ご契約いただいたお客様は、ご利用前に本書をよくお読みの上、本サービスをご利用ください。なお、本書は大切に保管してください。

ご採用いただけないお客様は、お客様の責任で本書をすみやかに廃棄願います。

本書の作成にあたって、細心の注意を払い可能な限り分かりやすい記載に努めていますが、本書の記述に誤りや欠落があっても弊社はいかなる責任も負わないものとします。本書及びその記述内容は予定なく変更される場合があります。

本書の内容の一部または全部を無断で複製・転載・改編しないでください。

免責事項について

- ・本サービスのマニュアルに記載されていない操作を行なったことで思わぬ誤動作や、予期せぬ課金が発生した場合であっても一切の責任を追いかねます。
- ・本サービスの利用、または利用できなかったことにより万が一損害（業務の中断・データの破損/損失・事故などによる損害や第三者からの賠償請求の可能性を含む）が生じたとしても、責任を一切負いかねます。

<用語>

用語	説明	備考
IoT	Internet of Things の略 「モノのインターネット」と言われるが、「モノがインターネットプロトコル(インターネット言語)でネットワークされている状態」のこと。	
REST	REpresentational State Transfer の略 複数のソフトウェア連携に適した設計原則を Web に適用したソフトウェアの設計様式を示す。 特定の URL にパラメータを指定して HTTP でアクセスすると XML で記述されたメッセージが送られてくるような、呼び出しインターフェース(「RESTful API」と呼ばれる)。 本サービスでは XML ではなく JSON 形式で返信する。	
MQTT	Message Queuing Telemetry Transport の略 MQTT とは、TCP/IP ネットワーク上の通信プロトコルで、多数の主体の間で短いメッセージを頻繁に送受信する用途に向けた軽量なプロトコル。M2M ネットワークや IoT (Internet of Things) 分野での普及が見込まれている。ヘッダ部分が最小で 2 バイトと小さく、HTTP で同様の通信を行う場合より、通信量や CPU 負荷・電力消費量などを数十分の一に抑える。	
ダイナミック リソース コントローラー	Dynamic Resource Controller 富士通の広域分散技術を用いた分散制御処理。データ収集においてトラフィックの変動に従い、限られたリソースの中で最適化した収集の実現を補佐する機能。	
リソース	リソースデータの収集単位	
リソースデータ	1つのデータ	
アクセスコード	リソースを対象に設定する認可情報	

- 目次 -

第1章	はじめに	3
1.1.	本書の目的	3
1.2.	ドキュメント構成	3
第2章	全体共通事項	4
2.1.	各パラメータの最大・最小値	4
2.2.	注意事項	8
2.2.1.	APIによる課金について	8
2.2.2.	リソース、アクセスコード及びイベントのAPI操作について	8
2.2.3.	同一リソースに対するREST/MQTTの両立について	9
2.2.4.	リソース/アクセスコード/イベント制御インターフェースの登録・更新時のJSONキーについて	9
2.2.5.	リソースパスについて	9
2.2.6.	リソース、アクセスコードおよびイベントの設定反映時間について	9
2.2.7.	通信頻度に対するシステム制限について	9
2.3.	REST(HTTP)共通事項	10
2.3.1.	Request	10
2.3.2.	Response	10
2.4.	CORS (Cross-Origin Resource Sharing)対応	12
2.4.1.	Preflight Request	12
2.4.2.	本要求リクエスト	14
2.5.	MQTT 共通事項	14
2.5.1.	Request	14
2.5.2.	Response	16
第3章	リソース_JSON / リソース(転送)_JSON のデータ制御(REST)	18
3.1.	リソース_JSON へのデータ登録 / リソース(転送)_JSON によるデータ転送	18
3.2.	リソース_JSON のデータ参照	23
3.3.	リソース_JSON のデータ検索	24
3.4.	リソース_JSON のデータ更新	28
3.5.	リソース_JSON のデータ削除	30
第4章	リソース_JSON / リソース(転送)_JSON へのデータ制御(MQTT)	32
4.1.	リソース_JSON へのデータ登録 / リソース(転送)_JSON へのデータ転送	32
4.2.	リソース_JSON / リソース(転送)_JSON のデータ参照	33
第5章	リソース_Binary データ制御(REST)	35
5.1.	リソース_Binary へのデータ登録	35
5.2.	リソース_Binary のデータ参照	36
5.3.	リソース_Binary のデータ検索	37
5.4.	リソース_Binary のデータ更新	40
5.5.	リソース_Binary のデータ削除	40
第6章	リソース_Binary データ制御(MQTT)	42
6.1.	リソース_Binary へのデータ登録	42
6.2.	リソース_Binary のデータ参照	42
第7章	リソース制御(REST)	44
7.1.	リソースの登録	44
7.2.	リソースのメタデータ参照	45
7.3.	リソースのメタデータ更新	48
7.4.	リソースの削除	48
第8章	アクセスコード制御(REST)	50
8.1.	アクセスコードの登録	50
8.2.	アクセスコードの参照	52
8.3.	アクセスコードの更新	54
8.4.	アクセスコードの削除	55

第 9 章	イベント制御 (REST)	57
9.1.	イベントの登録	57
9.2.	イベント情報の参照	60
9.3.	イベント情報の更新	64
9.4.	イベントの削除	65
第 10 章	外部システム呼出しリファレンス	66
10.1.	API 呼び出し (イベント機能)	66
10.2.	Email 送信 (イベント機能)	67
10.3.	API 呼出し (リソース (転送) _JSON 機能)	67
第 11 章	システム格納データリファレンス	69
11.1.	リコメンドリソース	69
11.2.	負荷リソース	69
11.3.	エラー収集用リソース	70
付録 1	Response エラー時のメッセージ一覧	72

第1章 はじめに

1.1. 本書の目的

本書はIoT Platform (以下: 本サービス) のご利用に際し、APIを利用しアプリケーションを開発する方向けの「APIリファレンス」です。

1.2. ドキュメント構成

サービスのご利用に際し、お客様を支援するための以下のマニュアルをご用意しております。

マニュアル名	説明
IoT Platform サービス詳細説明書	本サービスのサービス仕様を説明しています。
IoT Platform ユーザーガイド	本サービスを利用したシステムを設計するにあたって、具体的な例を交えて設計指針の説明を行うマニュアルです。
IoT Platform API リファレンス	本サービスで提供するサービスを利用したアプリケーションを設計するためのAPIのリファレンスマニュアルです。(本書)
IoT Platform サービスポータル操作マニュアル	Web インターフェース機能(以下: サービスポータル)に関するマニュアルです。

Memo

リソース、アクセスコードなど本サービス全般の定義・考え方については、「IoT Platform サービスポータル操作マニュアル」第3章をご参照ください。

また、ダイナミックリソースコントローラーについては「IoT Platform ユーザーガイド」を参照ください。

第2章 全体共通事項

本サービスの保持するデータに対して、REST(HTTP)/MQTT にてアクセスすることによる操作が可能です。本章ではその共通事項について説明します。

2.1. 各パラメータの最大・最小値

表 1: 外部インタフェース最大・最小値一覧に、REST(HTTP) のヘッダ、URI 及び MQTT の Topic において指定するパラメータの最大・最小値を説明します。

表 2: filter 条件詳細リソースデータ制御インタフェース 最大・最小値一覧に、REST(HTTP)にて検索時に指定する filter 条件の詳細パラメータの最大・最小値を説明します。

表 3: filter 条件詳細アクセスコード制御・イベント制御インタフェース 最大・最小値一覧に、API 操作によりアクセスコードやイベントを登録・参照・更新・削除する時に指定する filter 条件の詳細パラメータの最大・最小値を説明します。

表 4: select 条件詳細リソースデータ制御インタフェース 最大・最小値一覧に、select 条件に指定する詳細パラメータの最大・最小値を説明します。

表 5: Body 部リソースデータ制御インタフェース最大・最小値一覧に、REST の Body や MQTT の Payload に展開されるリソースデータに記載するパラメータの最大・最小値を説明します。

表 1：外部インタフェース最大値・最小値一覧

パラメータ名	説明	使用可能文字	最小	最大
リソースパス	リソースを一意に表現するためのパス。/(スラッシュ)区切りで階層的に表現できます。 先頭の「/」はリソースパスに含みません。	・半角英数字 ・「-」、「_」、「/」 「/-」、「/_」のように、 先頭や「/」の後に「-」と「_」 は指定不可。 「//」のように「/」の連続も不可	2 文字	128 文字 (但しリソース(転送用)の場合「_fwd」、リソース _Binary の場合、「_bin」を含めて 128 字)
拡張子	リソースデータのフォーマットを表す拡張子	json、csv、txt、bin	3 文字	4 文字
圧縮タイプ	Body データを圧縮して送信する際のデータの圧縮タイプ	gz のみ	2 文字	2 文字
アクセスコード	外部 IF アクセスの権限を示す識別子	半角英数字のみ	3 文字	48 文字
登録日時	登録対象データに付与する / 付与された登録日時。 ISO8601(基本表記としてのミリ秒表現を使用)に従います (20141225T103612.001Z など)。 精度はミリ秒(ミリ秒を省略した場合、0 ミリ秒とみなします)	・半角英数字および、+ or - ・英字は日付と時刻の間の T と、UTC を表す Z のみ許容します。	16 文字	24 文字

RETAIN	MQTT broker 側で本登録データを保持しておくかどうか。 ・ true : 保持する ・ false : 保持しない	true または false	4 文字	5 文字
filter 条件	処理対象を、<filter 条件>に一致するものに限定します。 <filter 条件>は、「プロパティ名 演算子 条件値」とし、and or で複数定義できます。登録日時を示すプロパティ名として _date を使用できます。	後述の name, value に使用可能な文字列に従います。 value では、「'」（シングルクォーテーション）で囲ったものを文字列、そうでないものを数値として扱います。	6 文字	256 文字
top 件数	検索結果として得られるデータ数を限定します	半角数字	1 件	1000 件
skip 件数	検索結果として得られたデータを指定件数分 skip します	半角数字	0 件	100,000 件
選択 key	Select 条件時の選択 key \$select=<選択 key> <選択 key>で指定されたフィールドのデータのみを返します。 <選択 key>は、JSON での name、XML での要素名、属性名に相当します。	・後述の name に使用可能な文字列に従います。 ・<選択 key>は「,」区切りで複数指定できます。 例) \$select=data.no,data.name	8 文字	256 文字
Content-Type	Body の MIME タイプ	RFC2046、RFC6838、RFC4289、RFC6657 に従います。	0 文字	128 文字
x-iotpf-request-id	リクエスト識別 ID	%x21-%x7E (印字可能な US - ASCII)	0 文字	64 文字
x-iotpf-meta-data1, x-iotpf-meta-data2, x-iotpf-meta-data3	リソースデータのメタデータ	%x21-%x7E (印字可能な US - ASCII)	1 文字	128 文字
JSON 配列要素数	Body 文に JSON 配列を記載する際における配列要素数	-	-	1,000

表 2: filter 条件詳細リソースデータ制御インタフェース 最大・最小値一覧

パラメータ名	説明	使用可能文字	最小	最大
プロパティ名	JSON 形式のリソースデータにおける name	<ul style="list-style-type: none"> ・「”」、「¥(*1)」、「'」、「&」、「\$」、「(」、「)」制御コードを除く、全ての Unicode 文字 (UTF-8) ・URI の非予約文字 (「半角英数字」、「-」、「.」、「_」、「~」) 以外はパーセントエンコードすること。 ・先頭文字として「_」は使用不可 ・name として「and」、「or」、「eq」、「ne」、「lt」、「le」、「gt」、「ge」は未対応。 ・name が階層構造になっている場合は、<name>.<name>で表現する。深さは最大 15 	1 文字	128 文字
条件値	JSON 形式のリソースデータにおける value に対する条件値	<ul style="list-style-type: none"> ・「”」、「¥(*1)」、「'」、「\$」制御コードを除く、全ての Unicode 文字 (UTF-8) ・URI の非予約文字 (「半角英数字」、「-」、「.」、「_」、「~」) 以外はパーセントエンコードする必要があります。 ・「'」で囲ったものを文字列、そうでないものを数値として扱います。 ・null は、値が存在しないことを示します。 	文字列の場合、 最大 256 文字	
			整数の場合、 -9999999999999999 ~ 9999999999999999	
			実数の場合、 倍精度浮動小数点数	
比較演算子	eq(等号)、ne(不等号)、gt(より大きい)、ge(以上)、lt(より小さい)、le(以下)	eq、ne、gt、ge、lt、le のいずれか	1 回	8 回 (出現回数)
論理演算子	and(論理積)、or(論理和)。論理演算子は「()」を使用して、(A eq 1 and B eq 1)or(A eq 2 and B eq 2) のような表現ができます。ただしこの場合、()内に()は定義できません。例えば、((A eq 1 and B eq 1)or(A eq 2 and B eq 2))and(C eq 1)は誤りになります。	and、or のいずれか	0 回	8 回 (出現回数)

(*1) U+005C(REVERSE SOLIDUS)

表 3 : filter 条件詳細アクセスコード制御・イベント制御インタフェース 最大・最小値一覧

パラメータ名	説明	使用可能文字	最小	最大
プロパティ名	フィルタプロパティ名	_resource_path	各 1 回 (出現回数)	各 1 回 (出現回数)
条件値	各プロパティ名に対する条件値。 startwith()による前方一致を使用可能。	<ul style="list-style-type: none"> ・「”」、「¥(*1)」、「'」、 「\$」制御コードを除く、全ての Unicode 文字 (UTF-8 1) ・URI の非予約文字 (「半角英数字」、「-」、「.」、「_」、「~」) 以外はパーセントエンコードすること ・文字列のみを対象とし、必ず「'」で囲う 	文字列の場合、 最大 256 文字	
比較演算子	eq(等号)	eqのみ	-	3 回 (出現回数)
論理演算子	and(論理積) 論理演算子が and のみであるため、「()」は使用不可	andのみ	-	2 回 (出現回数)

(*1) U+005C(REVERSE SOLIDUS)

表 4 : select 条件詳細リソースデータ制御インタフェース 最大・最小値一覧

パラメータ名	説明	使用可能文字	最小	最大
選択 key	JSON 形式のリソースデータにおける name	<ul style="list-style-type: none"> ・「”」、「¥(*1)」、「'」、「&」、 「\$」、「(」、「)」制御コードを除く、全ての Unicode 文字 (UTF-8 1) ・URI の非予約文字 (「半角英数字」、「-」、「.」、「_」、「~」) 以外はパーセントエンコードすること ・先頭文字として「_」は使用不可 ・name が階層構造になっている場合は、<name>.<name>で表現する。深さは最大 15 	1 文字	128 文字
選択 key の複数指定	<選択 key>は「,」区切りで複数指定可	「,」	-	10 回 (出現回数)

(*1) U+005C(REVERSE SOLIDUS)

表 5 : Body 部リソースデータ制御インタフェース最大・最小値一覧

パラメータ名	説明	使用可能文字	最小	最大
-	Body 部に記載できる文字 (Bulk Insert 機能利用時)	JSON 形式の配列であること	0 Byte	16Mbyte (JSON を BSON 形式 に変換し た場合の サイズ)
-	Body 部に記載できる文字	拡張子に応じた内容であること ・ json または省略の場合 JSON 形式 ・ csv の場合 csv ファイル形式 ・ txt の場合 テキストファイル形式 ・ bin の場合 任意のバイト列	0 Byte	256kbyte (圧縮送 信時は伸 長後のサ イズ)
name	JSON 形式のリソースデータにお ける name	・「”」、「¥(*1)」、「'」、「&」、 「\$」、「(」、「)」制御コードを 除く、全ての Unicode 文字(UTF-8) ・先頭文字として「_」は使用不可 ・name として「and」、「or」、「eq」、 「ne」、「lt」、「le」、「gt」、 「ge」は未対応(検索不可)。 ・name の階層の深さは最大 15	1 文字	128 文字
value	JSON 形式のリソースデータにお ける value	・「”」、「¥(*1)」、「'」、「\$」 制御コードを除く、全ての Unicode 文字(UTF-8) ・「”」で囲ったものを文字列、 そうでないものを数値として扱 います。 ・null は、値が存在しないことを 示します。 ・true/false は、bool 値であるこ とを示します。	文字列の場合、 0 ~ Body 部としての 最大長	
			整数の場合、 -9999999999999999 ~ 9999999999999999	
			実数の場合、 倍精度浮動小数点 数(ただし、表示形 式は必ずしも保証 されません)	

(*1) U+005C(REVERSE SOLIDUS)

2.2. 注意事項

2.2.1. API による課金について

本サービスでは、リソース ID 数、API、SSL やイベントの利用回数に応じて従量課金させていただきます。

2.2.2. リソース、アクセスコード及びイベントの API 操作について

リソース、アクセスコード及びイベントの追加・参照・更新・削除を API 操作するためには、予め create、delete、list 権を持ったリソース及びアクセスコードが必要です。また、API 操作できるのは、create、delete、

list 権を持ったアクセスコードが指定するリソースパス配下のみです。

リソース及びアクセスコードの API 操作を行う場合、予めサービスポータルより、上階層のリソース及びアクセスコードの登録をお願いします。

2.2.3. 同一リソースに対する REST / MQTT の両立について

「リソース_JSON」では、REST で登録したリソースデータを MQTT にて SUBSCRIBE により参照することができます。

但し、「リソース_Binary」においては REST で登録したリソースデータを MQTT にて SUBSCRIBE により参照することはできません。

2.2.4. リソース/アクセスコード/イベント制御インターフェースの登録・更新時の JSON キーについて

本ドキュメントに記載した JSON キー以外のキーは使用不可とし、使用された場合は 400 応答します。また、他の制御インターフェースにおいて必要ない JSON キーが設定されていた場合も、400 応答します。例えば、「リソース_JSON」においてリソースデータを登録する際に、リクエストの Body に“ fwd_info ”キーを設定すると“ fwd_info ”は「リソース(転送)_JSON」に必要な設定であるため 400 応答されます。

2.2.5. リソースパスについて

リソースパスは以下のように定義しています。

- ・「リソース_JSON」: <お客様任意>
- ・「リソース(転送)_JSON」: _fwd/<お客様任意>
先頭の「_fwd/」制御文字も含みリソースパスとします。
- ・「リソース_Binary」: _bin/<お客様任意>
先頭の「_bin/」制御文字も含みリソースパスとします。

2.2.6. リソース、アクセスコードおよびイベントの設定反映時間について

サービスポータルや制御 API 操作にてリソース、アクセスコードおよびイベントの追加/変更/削除を実施した後、実施内容が利用できるようになるまで、最大 5 分程度の時間がかかります。(なお、この時間は将来のエンハンス等において変更される可能性があります。)

2.2.7. 通信頻度に対するシステム制限について

通信頻度が一定時間目安を超えた場合、すべてのリクエストがエラー返却またはデータ破棄されます。これらの事象の発生は下記から確認することができます。本事象となった場合には、通信頻度の調整、または料金プランの変更をご検討ください。

通信手段	発生する事象	確認する方法
REST(HTTP)	エラー返却	HTTP レスポンスにより確認できます。 429 Too Many Requests の場合に発生しています。
MQTT	データ破棄	エラーログ蓄積機能により確認できます。 “ trigger ” : “ MP ” , “ error_factor_message ” : “ Failed to execute publishing resource data, in the system internal processing. ” となるエラーログ蓄積の場合に発生しています。

2.3. REST(HTTP)共通事項

2.3.1. Request

Parameters	値	補足
Base URL	http://<zone>.fujitsu.com https://<zone>.fujitsu.com	<zone>に入る値はご契約後の通知内容に従ってください。

- http の場合は 80/tcp ポートを使用
- https の場合は 443/tcp ポートを使用(TLS1.2 に対応)
- 注)
 - URL のパス部(「?」より前の部分)は、パーセントエンコードが不要になります。(パーセントエンコード不可)

Headers	値	補足	M/O(*1)
Authorization	Bearer <アクセスコード>	アクセスコードはサービスポータルにて設定した値	M
Accept-Encoding	gzip	取得データ(ペイロード)の gzip 圧縮受信を希望する場合に指定	0(*2)
Range	bytes=<先頭位置>- bytes=<先頭位置>-<末尾位置> bytes=-<サイズ> コンマ区切りにより複数範囲の指定が可能	データの一部のみ取得を希望する場合に指定 複数範囲を指定した場合は、マルチパートで応答します。	0(*3)

(*1) M: 必須、O: オプション

(*2) 3.2 リソース_JSON のデータ参照、および 3.3 リソース_JSON のデータ検索のみで利用できます。

(*3) 5.2 リソース_Binary のデータ参照のみで利用できます。

2.3.2. Response

Status-Code	Reason-Phrase(*1)	説明
200	OK	成功、リソースデータ作成の成功
201	Created	リソース、アクセスコード、イベント作成の成功
204	No Content	以下の何れかに該当 ・リソースデータ参照時において、リソースは存在するが該当するリソースデータが存在しない ・リソースの削除において、削除成功 ・メタデータ・アクセスコード・イベント参照時において、各種情報が存在しない (将来変更する場合があります。) ・メタデータ・アクセスコード・イベント削除時において、削除成功
206	Partial content	部分取得の成功
400	Bad Request	リクエストデータに不正値があります
401	Unauthorized	リソースへのアクセス権がありません
403	Forbidden	アクセス権がありません
404	Not Found	リソースが存在しません
405	Method Not Allowed	該当のメソッドタイプは許可されていません
408	Request Time-out	リクエストタイムアウトです
409	Conflict	他のリソースと競合しています
411	Length Required	サーバアクセスを拒否しました(Content-Length 指定なし)
412	Precondition Failed	サーバアクセスを拒否しました(リクエスト条件が不正)
413	Payload Too Large	サーバアクセスを拒否しました(リクエストボディサイズがサーバ許容範囲超越)

414	URI Too Long	サーバアクセスを拒否しました (URI が長い)
415	Unsupported Media Type	サーバアクセスを拒否しました (未サポート Content-Type)
416	Requested Range Not Satisfiable	サーバアクセスを拒否しました (Range 要求の値が不正)
421	Misdirected Request	レスポンスを生成できないサーバに送信されました
423	Locked	リソースがロックされています
429	Too Many Requests	契約上のトラフィック上限を超えています
495	SSL Certificate Error	無効なクライアント証明書を受信しました
496	SSL Certificate Required	クライアントからクライアント証明書が送付されませんでした
497	HTTP Request Sent to HTTPS Port	HTTPS リクエストポートにて HTTP リクエストを受信しました
500	Internal Server Error	サーバ側の問題による失敗です
501	Not Implemented	サーバで未サポートリクエストのメソッドが送信されました
502	Bad Gateway	ゲートウェイサーバが起動していません
503	Service Unavailable	一時的にアクセスできません
504	Gateway Time-out	ゲートウェイサーバが時間内にレスポンスを返せませんでした

(*1) 以下の API 実行時は付与されません。

第 5 章 リソース_Binary データ制御 (REST) 記載の全 API

第 7 章 リソース制御 (REST) 記載の全 API

第 8 章 アクセスコード制御 (REST) 記載の全 API

第 9 章 イベント制御 (REST) 記載の全 API

Headers	値	補足
Content-Encoding	gzip	取得データ (ペイロード) が gzip 圧縮伝送された場合に限り付与します。 (*1)
Content-Length	Body サイズ (byte)	-
Content-Range	<先頭位置>-<末尾位置>/<全体長>	206 応答にて付与します。 Range 要求において複数範囲を指定するとマルチパート応答となり、それぞれのパートに付与します。

(*1) 3.2 リソース_JSON のデータ参照、および 3.3 リソース_JSON のデータ検索の Request にて Accept-Encoding が指定され、かつ取得したデータの圧縮前のサイズが 1024byte を超える場合に付与します。

1. エラー時は以下の形式で詳細情報を設定する

Headers	値	補足
Content-Type	application/json または application/json; charset=UTF-8	固定

1. Body

```
{
  "errors": [
    {
      "message": "<メッセージ>",
      "<任意の name>": "<任意の value>"
    }
  ]
}
```


パラメータ名	値	補足	M/O(*1)
メッセージ	エラーの詳細情報		M
任意の name	エラーの付加情報の name	エラーに応じて任意の name を設定	0
任意の value	<任意の name>に応じた value		0

(*1) M: 必須、0: オプション

エラー時は Body に詳細情報が設定されます。詳細情報については、付録 1 を参照してください。

2.4. CORS (Cross-Origin Resource Sharing)対応

Web ブラウザでは、同一生成源ポリシー (Same-Origin Policy) によって Web ページを生成したドメイン以外への REST (HTTP) リクエストは基本的にはできません。IoT Platform では、安全にクロスドメインアクセスを許可する仕組みとして CORS (Cross-Origin Resource Sharing)(*1) に対応しています。

(*1) W3C にて規定されています。(http://www.w3.org/TR/cors/)

CORS 仕様に準じてクロスドメインアクセスを行う際の通信手順には、以下の 2 種類があります。

- 直接、クロスドメインアクセスするリクエスト (本要求リクエスト) を送信する手順。
- Preflight Request (クロスドメインアクセスが可能か否かを確認するリクエスト) を送信し、そのレスポンスを受けた後に改めてクロスドメインアクセスするリクエスト (本要求リクエスト) を送信する手順。

なお、CORS 対応は REST (HTTP) アクセスに限り動作します。サービスポータルへのアクセスには動作しませんのでご注意ください。

2.4.1. Preflight Request

● Request

Method	値	補足
Options	-	-

Headers	値	補足	M/O(*1)
Origin	<オリジンサーバのドメイン>	-	M
Access-Control-Request-Method	POST、PUT、GET、DELETE のいずれか	本要求の REST (HTTP) で使用するメソッドを指定してください。	M
Access-Control-Request-Headers	Authorization	-	M
	Content-Type	下記参照ください。	0
	Range	下記参照ください。	0
	x-iotpf-meta-data1 x-iotpf-meta-data2 x-iotpf-meta-data3	下記参照ください。	0
	x-iotpf-request-id	下記参照ください	0
	Accept-Encoding	下記参照ください	0

(*1) M: 必須、0: オプション

- Access-Control-Request-Headers の補足事項について
 - ✧ 複数ヘッダを指定する場合は、「,」(コンマ) で区切って指定します。
 - ✧ Content-Type は、本要求リクエストが以下となる場合に必要です。
 1. リソースの登録 (7.1 節)

2. リソースのメタデータ更新 (7.3 節)
 3. アクセスコードの登録 (8.1 節)
 4. アクセスコードの更新 (8.3 節)
 5. イベントの登録 (9.1 節)
 6. イベント情報の更新 (9.3 節)
 7. リソース_JSON へのデータ登録 / リソース (転送)_JSON によるデータ転送 (3.1 節)
 8. リソース_JSON のデータ更新 (3.4 節)
 9. リソース_Binary へのデータ登録 (5.1 節)
- ◇ x-iotpf-meta-data1、x-iotpf-meta-data2、x-iotpf-meta-data3、Range は、本要求リクエストが以下となる場合に必要です。
 1. リソース_Binary データ制御 (REST) (第 5 章)
 - ◇ x-iotpf-request-id は、本要求リクエストが以下となる場合に必要です。
 1. リソース_JSON へのデータ登録 / リソース (転送)_JSON によるデータ転送 (3.1 節)
 2. リソース_JSON のデータ更新 (3.4 節)
 3. リソース_JSON のデータ削除 (3.5 節)
 4. リソース_Binary へのデータ登録 (5.1 節)
 5. リソース_Binary のデータ削除 (5.5 節)
 - ◇ Accept-Encoding は、本要求リクエストが以下となる場合に必要です。
 1. リソース_JSON のデータ参照 (3.2 節)
 2. リソース_JSON のデータ検索 (3.3 節)
- サービスポータルにて CORS が有効と設定されている場合において、Origin ヘッダが付与され、かつ必須ヘッダが付与されていない場合は、エラー応答 (400) を返します。

Body	値	補足
なし	-	-

● Response

Status-Code	Reason-Phrase	説明
2.3.2 節を参照	-	-

Headers	値	補足
Access-Control-Allow-Origin	*	固定 (全てのオリジンを許可します)
Access-Control-Allow-Methods	OPTIONS, POST, PUT, GET, DELETE	固定
Access-Control-Allow-Headers	Authorization	固定
	Content-Type	固定
	Range	リソース_Binary データ制御 (REST) 時
	x-iotpf-meta-data1 x-iotpf-meta-data2 x-iotpf-meta-data3	リソース_Binary データ制御 (REST) 時
	x-iotpf-request-id	リソース_JSON / リソース (転送)_JSON のデータ制御 (REST)、リソース_Binary データ制御 (REST) 時
	Accept-Encoding	リソース_JSON / リソース (転送)_JSON のデータ制御 (REST) 時
Access-Control-Max-Age	86400	固定 (24 時間)

- Access-Control-Request-Headers に複数ヘッダを指定する場合は「,」(コンマ)で区切って指定します。

Body	値	補足
------	---	----

ボディ	成功：なし 失敗：あり（失敗エラー情報）	-
-----	-------------------------	---

2.4.2. 本要求リクエスト

CORS を利用する場合の、通常リクエストとの差分を以下に示します。その他の共通事項は 2.3 節を参照ください。

● Request

クロスドメインアクセスする場合の Request には、以下のヘッダを付与してください。

Headers	値	補足	M/O(*1)
Origin	<オリジンサーバのドメイン>	-	M

(*1) M：必須、O：オプション

● Response

Headers	値	補足
Access-Control-Allow-Origin	*	固定（全てのオリジンを許可します） CORS リクエストに対して許可した場合に限り付与します。 CORS 機能は、サービスポータルから利用するか否かの設定ができます。利用する場合に限り CORS リクエストを許可します。
Access-Control-Expose-Headers	Location Content-Length Content-Range x-iotpf-meta-data1 x-iotpf-meta-data2 x-iotpf-meta-data3 Content-Encoding	固定 CORS リクエストに対して許可した場合に限り付与します。

2.5. MQTT 共通事項

本サービスでは、MQTT v3.1 をサポートしています。

2.5.1. Request

● MQTT ブローカーアドレス

MQTT ブローカーアドレス	ポート番号	SSL/TLS
<zone>.fujitsu.com	1883/TCP	
<zone>.fujitsu.com	8883/TCP	

- <zone>に入る値はご契約後の通知内容に従ってください。
- SSL/TLS は TLS1.2 に対応しています。

● 利用可能なメッセージタイプ

メッセージタイプ	値	説明	利用可否
予約済み	0	予約済み	×
CONNECT	1	クライアントからサーバへの接続要求	
CONNACK	2	接続確認応答	(送信不可)
PUBLISH	3	メッセージ発行	
PUBACK	4	PUBLISH の確認応答 (QoS=1)	

PUBREC	5	PUBLISH の確認応答 (QoS=2,part1)	
PUBREL	6	PUBLISH の解放要求 (QoS=2,part2)	
PUBCOMP	7	PUBLISH の解放応答 (QoS=2,part3)	
SUBSCRIBE	8	購読要求	
SUBACK	9	SUBSCRIBE 確認応答	(送信不可)
UNSUBSCRIBE	10	購読解除要求	
UNSUBACK	11	UNSUBSCRIBE 確認応答	(送信不可)
PINGREQ	12	PING 要求	
PINGRESP	13	PING 応答	(送信不可)
DISCONNECT	14	切断通知	
予約済み	15	予約済	×

● 固定ヘッダ設定値

Parameters	値	補足	適用対象のメッセージタイプ
DUP フラグ	0 or 1 を任意に設定できます。	0:初回メッセージ 1:再送メッセージ	PUBLISH PUBREL SUBSCRIBE UNSUBSCRIBE
QoS フラグ	0 or 1 or 2 を任意に設定できます。 但し、セッション断を跨いだ場合において、1 or 2 を設定していても0と同等の効果となります。 MQTT クライアント~ブローカー間の QoS であり、エンド~エンドではありません。	0:最高一回 1:最低一回 2:正確に一回	PUBLISH SUBSCRIBE
RETAIN フラグ	0 or 1 を任意に設定できます。	0:MQTT broker で最新情報を保持しない 1:保持する	PUBLISH

● 可変ヘッダ設定値

Parameters	値	補足	適用対象のメッセージタイプ
プロトコル名	MQIsdp	固定	CONNECT
プロトコルバージョン	3	固定	CONNECT
User name フラグ	1	固定	CONNECT
Password フラグ	1	固定	CONNECT
Will RETAIN	0	固定	CONNECT
Will QoS	0	固定	CONNECT
Will フラグ	0	固定	CONNECT
Clean Session フラグ	1	固定	CONNECT
キープアライブ・タイマー	1 秒以上 1800 秒以下	1800 秒より大きい値もしくは 0 秒を設定した場合、エラー (0x05: 接続拒否) として切断	CONNECT
トピック名	<アクセスコード>/v1/<Tenant ID>/<リソース		PUBLISH SUBSCRIBE UNSUBSCRIBE

	パス> (*1)	
--	----------	--

(*1) <アクセスコード>の部分は、アクセス可否の判定のために利用します。適切な<アクセスコード>が設定されている場合、/v1/<Tenant ID>/<リソースパス>の部分が一致すれば MQTT クライアント間でメッセージ交換をすることができます。

● ペイロード設定値

● CONNECT

Parameters	値	最大長	補足
クライアント識別子	クライアントを一意に識別する文字列	23 文字	クライアントごとに重複なく設定してください。
User Name	ユーザ名	10 文字	ご契約後の通知内容に従ってください。 (テナント ID 固定)
Password	パスワード	12 文字	サービスポータルより設定

● PUBLISH

◇ リソースデータ登録メッセージ

● SUBSCRIBE

◇ 購読対象のトピックのリスト

- トピック名は「<アクセスコード>/v1/<Tenant ID>/<リソースパス>」とします。

● UNSUBSCRIBE

◇ 購読解除対象のトピックのリスト

- トピック名は「<アクセスコード>/v1/<Tenant ID>/<リソースパス>」とします。

● MQTT のトピック名におけるワイルドカードについて

以下のパターンのワイルドカード表現に対応しています。未対応のワイルドカード表現を使用した SUBSCRIBE/UNSUBSCRIBE メッセージを受信した場合、エラーとしてコネクションを切断します。

No	パターン	説明	マッチするリソースパスの例
1	<アクセスコード>/v1/<Tenant ID>/#	<Tenant ID>のテナント内の全てのリソースパス	DC/ DC/aaa/Tokyo DC/bbb/Tokyo/1 DC/aaa/Fukuoka/1 DC/bbb/Fukuoka など全て。
2	<アクセスコード>/v1/<Tenant ID>/DC/aaa/#	リソースパスの前方一致	DC/aaa DC/aaa/Tokyo DC/aaa/Fukuoka/1
3	<アクセスコード>/v1/<Tenant ID>/DC/+Tokyo	リソースパスの部分一致	DC/aaa/Tokyo

◇ 「<アクセスコード>/v1/<Tenant ID>/」は必須とします。

◇ 以下のワイルドカードパターンには対応していません。

- 「+」が複数出現する
- 「#」と「+」の組み合わせ
- 最後に「+」が出現する

2.5.2. Response

● 固定ヘッダ

- 特になし

● 可変ヘッダ

Parameters	値	補足	適用対象のメッセージタイプ
接続戻りコード	0: 接続許可		CONNACK

	1: 接続拒否(プロトコルバージョン許容不可) 2: 接続拒否(識別子拒否) 3: 接続拒否(サーバ使用不可) 4: 接続拒否(ユーザ名/パスワードが不正) 5: 接続拒否(権限なし)		
--	--	--	--

- ペイロード
 - 特になし

第3章 リソース_JSON / リソース(転送)_JSON のデータ制御(REST)

リソース_JSON 及びリソース(転送)_JSON の制御が行える REST のインターフェースの一覧を以下に示します。

表 6 : データ蓄積 / 参照インターフェース一覧

プロトコル	種別	用途	章
REST	リソース_JSON へのデータ登録 / リソース(転送)_JSON によるデータ転送	リソースに対して新たにデータを登録(転送)します。 [オプション] リソース_JSON の場合： ・なし： 本サービスでの受信日時でデータを登録 ・日時指定： 指定された日時でデータを登録 ・Bulk Insert フラグ指定： Bulk Insert でデータを登録 Bulk Insert とは一回の API 起動で複数のデータを登録することを指します。 リソース(転送)_JSON の場合： ・なし：本サービスにおいて、限定された URL に対してデータを転送します。URL を指定しない場合、MQTT Broker 連携のみ機能します。	3.1 章
	リソースデータの参照	リソースに登録済みのデータを参照します。 [オプション] ・なし： 最新データを返します。 ・日時指定： 指定日時のデータを返します。	3.2 章
	リソースデータの検索	リソースに登録済みのデータを、条件を指定して検索します。検索結果として、HIT した登録済みデータそれぞれに、本サービスにおいて管理する登録日時情報を付加した上でデータを返します。 [オプション] ・QUERY にて、検索条件を指定できます。	3.3 章
	リソースデータの更新	リソースに登録済みのデータを更新します。 [オプション] ・新登録日時を指定： データの更新とともに、登録日時を変更します。 ・なし： データの更新のみを行い、登録日時は変更しません。	3.4 章
	リソースデータの削除	リソースに登録済みのデータを削除します。 [オプション] ・QUERY にて、削除条件を指定できます。	3.5 章

3.1. リソース_JSON へのデータ登録 / リソース(転送)_JSON によるデータ転送

・ Request

Parameters	値
Method	PUT
URI	データ蓄積 <Base URL>/v1/<Tenant ID>/<リソースパス>.<拡張子>.<圧縮タイプ>?\$date=<登録日時>&\$retain=<RETAIN>&\$bulk=<Bulk Insert フラグ>&\$charset=<Body データ文字コード>&\$skip=<Body データ削除指定行>&\$numconv=<数値変換>

データ転送	<Base URL>/v1/<Tenant ID>/_fwd/<リソースパス>.<拡張子>.<圧縮タイプ>?\$date=<登録日時>&\$retain=<RETAIN>&\$charset=<Body データ文字コード>&\$skip=<Body データ削除指定行>&\$numconv=<数値変換>
Headers	2.3.1 節 参照
Body	登録対象データ

Memo

IoT Platform のデータ登録では、「存在するリソースにデータを格納する」という考え方から、Method に PUT を使用する仕様です。リソースへの POST Method は「リソースの作成(7.1 リソースの登録)」にて使われます。

◇ URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M
Tenant ID	リソース所有テナントの識別子	M
リソースパス	データ登録対象のリソースパス	M
拡張子	以下のいずれかを指定します。省略時は「.」を含め省略し、この場合は json が指定されたものとみなします。 ・ json ・ csv ・ txt ・ bin	0
圧縮タイプ	Body データを圧縮して送信する際の圧縮タイプを以下で指定します。省略時は「.」を含め省略し、この場合は Body データが無圧縮とみなします。 ・ gz	0
登録日時	登録対象データに付与する登録日時(*2) ・ 省略可(\$date=含む)で、省略時は本サービスでのリクエスト受信日時を採用します。 ・ 既に指定登録日時のデータが存在するかどうかはチェックせず、追記を行います。	0
RETAIN	MQTT broker 側で本登録データを保持しておくかどうか。 ・ true : 保持する ・ false : 保持しない false 時は、省略可(\$retain=含む) Bulk Insert 指定時は、RETAIN が指定されても無視します。	0
Bulk Insert フラグ	Bulk Insert (1 度に複数リクエストを送信すること。)を実行するか否かを指定します。拡張子が json、または省略の場合に限り有効です。 ・ none : Bulk Insert しません ・ single_resource_path : 単一リソースに対して Bulk Insert を実行します none 時は省略可(\$bulk=含む)	0
Body データ文字コード	Body データの文字コードを指定します。拡張子が csv、txt の場合に限り有効です。 ・ utf-8 : UTF-8 ・ shift_jis : Shift-JIS utf-8 時は省略可(\$charset=含む)	0
Body データ削除指定行	Body データの先頭から削除する行数を指定します。拡張子が csv の場合に限り有効です。省略することができ(\$skip=含む)、この場合は行削除を行いません。	0

数値変換	Body データ中の数値を文字列に変換するか否かを指定します。拡張子が csv の場合に限り有効です。 <ul style="list-style-type: none"> ・ true : 数値変換します。 ・ false : 数値変換しません。 true 時は省略可(\$numconv=含む)	0
------	--	---

(*1) M: 必須、0: オプション

(*2) ISO8601(基本表記としてのミリ秒表現を使用)に従います(20141225T103612.001Z など)。精度はミリ秒(ミリ秒を省略した場合、0ミリ秒とみなします)です。以降の「登録日時」はすべて同一仕様です。秒とミリ秒の区切りは「.」、タイムゾーン指定は「±hhmm」形式で省略時は「Z」を添えます。本サービスがレスポンスに格納する場合、UTC を用います。

1. URI パラメータのクエリに対し、上記一覧に記載がなく、かつ本章の他の要求(参照、検索、更新、削除)にて許容しているものを指定した場合には、エラーとならず無視します。許容されていないものは、解析失敗としてエラー応答します。

◇ HTTP ヘッダ

Headers	値	補足	M/O(*1)
Content-Type	・ 拡張子が json または省略の場合 application/json; charset=UTF-8	固定	0
	・ 拡張子が csv の場合 text/csv; charset=<charset> <charset>は以下のいずれか。 <ul style="list-style-type: none"> ・ UTF-8 ・ Shift_JIS 		
	・ 拡張子が txt の場合 text/plain; charset=<charset> <charset>は以下のいずれか <ul style="list-style-type: none"> ・ UTF-8 ・ Shift_JIS 		
	・ 拡張子が bin の場合 Body の MIME タイプ		
	・ 圧縮タイプが gz の場合(拡張子は任意) application/gzip または application/x-gzip		
x-iotpf-request-id	リクエスト識別 ID(任意文字列)(*2)	本サービス内および通信先の影響でリクエストが欠損した場合に出力するエラー情報に利用されます。エラー情報は 11.3 節を参照ください。	0

(*1) M: 必須、0: オプション

(*2) 値に「-」のみが設定された場合は、ヘッダ設定無しと扱います。

◇ Body (拡張子が json または省略の場合)

1. 共通事項

- メンバーの順序性は保証せず、登録時の並びで参照できるとは限りません。
- 重複した name を持つメンバは保存しません。
- JSON 配列の最大要素数は 1000 個です。

2. <Bulk Insert フラグ>なし または none の場合

- JSON 形式の登録対象のデータとして、配列([])での開始は許容しません。

3. <Bulk Insert フラグ> が single_resource_path の場合

- 各リソースデータを以下のJSONオブジェクトとし、JSON配列形式で複数格納できます。

Parameters	形式	説明	最大長	M/O(*1)
_date	日時(*2)	登録日時 ・設定されていなければ、クエリストリングの<登録日時>を設定します。クエリストリングの<登録日時>も設定されていなければ受信日時を設定します。	24 文字	0
_data	文字列	登録対象のリソースデータ本体 (JSON 形式)	全体の最大サイズに 従う	M

(*1) M: 必須、0: オプション

(*2) ISO8601(基本表記としてのミリ秒表現を使用)に従います(20141225T103612.001Z など)。精度はミリ秒(ミリ秒を省略した場合、0ミリ秒とみなします)です。以降の「登録日時」はすべて同一仕様です。秒とミリ秒の区切りは「.」、タイムゾーン指定は「±hhmm」形式で省略時は「Z」を添えます。本サービスがレスポンスに格納する場合、UTCを用います。

登録データ例:

```
[
  { "_date": "20160717T131520Z",
    "_data": { "name1": "value1" } },
  { "_date": "20160717T131521Z",
    "_data": { "name1": "value1" } }
]
```

- Bulk Insert フラグありのAPIはBulk Insert フラグなしのAPI料金と比較して高額となります。詳細はご契約時のサービスメニューをご参照ください。
 - Bulk Insert にて登録したデータに対してイベント処理は行いません。
 - Bulk Insert にて登録したデータはMQTT Subscriberに通知されません。
- ◇ Body (拡張子が csv の場合)
1. CSV形式をJSON形式に変換して格納します。変換後のデータ形式を以下に示します。
name : "csv"
value : 2重配列形式。
 2. CSVフォーマットはRFC4180に準ずる形式にしてください。
 - CSVの区切り文字「,(コンマ)」の直前/直後にある「" (ダブルクォーテーション)」は削除されます。
 - CSV中の改行コードは「CR LF」、「LF」に対応しています。
 3. CSV中の各データは以下のように解釈します。
 - 数字(小数点を含む)または指数表記であり、かつ\$numconv=falseの場合数値と解釈します。指数表記時、仮数部と指数部との記号は「e」または「E」です。
 - true、またはfalseの場合bool値と解釈します。
 - 上記以外の場合文字列と解釈します。\$charset=shit_jisが指定された場合は文字コードがShift-JISであると、UTF-8に変換します。
実際のBody中の文字コードによらず、Shift-JIS文字としてUTF-8へ変換します。
 4. Body先頭にUTF-8を示すバイトオーダーマーク(BOM)が付与されていた場合、削除されます。
 5. 転送リソースによる転送、イベント発生時の付与データ、およびMQTT Subscriberへの通知は、JSON形式に変換されたデータが通知されます。
 6. 以下に変換例を示します。

Body データ (CSV 形式)

```
node-a, true, 10.1
```

```
node-b, false, 20.0
```

格納データ(JSON形式)

```
{
  "csv": [
    [
      "node-a",
      true,
      10.1
    ],
    [
      "node-b",
      false,
      20
    ]
  ]
}
```

◇ Body (拡張子が txt の場合)

1. テキスト形式を JSON 形式に変換して格納します。変換後のデータ形式を以下に示します。
name : "txt"
value : 文字列
2. \$charset=shit_jis が指定された場合は文字コードが Shift_JIS であるとし、UTF-8 に変換します。
実際の Body 中の文字コードによらず、Shift-JIS 文字として UTF-8 へ変換します。
3. Body 先頭に UTF-8 を示すバイトオーダーマーク (BOM) が付与されていた場合、削除されます。
4. 転送リソースによる転送、イベント発生時の付与データ、および MQTT Subscriber への通知は、JSON 形式に変換されたデータが通知されます。
5. 以下に変換例を示します。

Body データ(テキスト形式)

```
#include <stdio.h>

int main(int argc, char **argv)
{
    printf("Hello, World!!\n");
}
```

格納データ(JSON形式)

```
{
  "txt": "#include <stdio.h>\n\nint main(int argc, char\n*argv){\nprintf(\"Hello, World!!\n\");\n}\n"
```

◇ Body (拡張子が bin の場合)

1. バイナリ形式を JSON 形式に変換して格納します。変換後のデータ形式を以下に示します。
name : "bin"
value : Base64 形式の文字列
2. 転送リソースによる転送、イベント発生時の付与データ、および MQTT Subscriber への通知は、JSON 形式に変換されたデータが通知されます。
3. 以下に変換例を示します。

Body データ(バイナリ形式)

```
1234567890
```

格納データ(JSON形式)

```
{
  "bin": "MTIzNDU2Nzg5MA=="
}
```

◇ Body の圧縮送信

1. 圧縮タイプに gz を指定した場合は、Body データを gzip にて圧縮し送信することができます。データ伸長し、拡張子に応じて JSON 形式に変換した後に格納します。
2. 処理可能な最大データ長は、伸長後のデータ長で判断します。
3. 転送リソースによる転送、イベント発生時の付与データ、および MQTT Subscriber への通知は、伸長および JSON 形式に変換されたデータが通知されます。

2. Response

◇ 正常応答時

Parameters	値
Status-Code	200 OK
Headers	2.3.2 節 参照
Body	なし

◇ エラー応答時については 2.3.2 節を参照ください。

3.2. リソース_JSON のデータ参照

・ Request

Parameters	値
Method	GET
URI	最新データの参照 <Base URL>/v1/<Tenant ID>/<リソースパス>/_present.<拡張子>?<QUERY> 過去データの参照 (*) <Base URL>/v1/<Tenant ID>/<リソースパス>/_past(<登録日時>).<拡張子>?<QUERY>
Headers	2.3.1 節 参照
Body	なし

(*) <登録日時>のデータが複数存在した場合、全てのデータを返却します。

◇ URI パラメータ

パラメータ名	説明	M/O(*)
Base URL	2.3.1 節 参照	M
Tenant ID	リソース所有テナントの識別子	M
リソースパス	データ参照対象のリソースパス	M
登録日時	参照対象データの登録日時(*)。	M
拡張子	以下を指定。省略時は「.」を含め省略します。 ・ json	O
QUERY	参照条件(後述)。省略時(?含む)は、条件なし(全て)として扱います。	O

(*) M: 必須、O: オプション

(*) ISO8601(基本表記としてのミリ秒表現を使用)に従います(20141225T103612.001Z など)。精度はミリ秒(ミリ秒を省略した場合、0ミリ秒とみなします)です。秒とミリ秒の区切りは「.」、タイムゾーン指定は「±hhmm」形式で省略時は「Z」を添えてください。本サービスがレスポンスに格納する場合、UTC を用います。

1. URI パラメータのクエリに対し、上記一覧に記載がなく、かつ本章の他の要求(登録、検索、更新、削除)にて許容しているものを指定した場合には、エラーとならず無視します。許容されていないものは、解析失敗としてエラー応答します。

◇ QUERY 一覧

書式	説明
\$select=<選択 key>	<選択 key>で指定されたフィールドのデータのみを返します。 ・ <選択 key>は、JSON での name、XML での要素名、属性名に相当し、登録デ

	<p>ータ内の任意の key を指定できます。フィールドの階層は「.」で表現します。</p> <ul style="list-style-type: none"> ・<選択 key>は「,」区切りで複数指定できます。 ・本サービスの管理データである、_date/_resource_path/_data は<選択 key>に使用できません。 <p>例) \$select=sensor.id,sensor.name,sensor.data.temp</p>
--	--

3. Response

◇ 正常応答時

Parameters		値
Status-Code		200 OK
Headers	Content-Type	<拡張子>に従った MIME タイプ。 ・ json の場合： application/json; charset=UTF-8
Body		一致したリソースデータ

◇ Body

以下の形式で、複数の登録済みデータを返却します。

Name	Value
_resource_path	対象データのリソースパス
_date	対象データの登録日時(*1)
_data	JSON 形式の登録済みのデータ

(*1) ISO8601(基本表記としてのミリ秒表現を使用)に従います(20141225T103612.001Z など)。精度はミリ秒(ミリ秒を省略した場合、0 ミリ秒とみなします)です。秒とミリ秒の区切りは「.」、タイムゾーン指定は「±hhmm」形式で省略時は「Z」を添えてください。本サービスがレスポンスに格納する場合、UTC を用います。

◇ エラー応答時

1. 参照対象のリソースが存在しなかった場合、404 Not Found を返します。
2. 条件に一致したリソースデータが存在しなかった場合、204 No Content を返します。
3. 条件に一致したリソースデータ数が 2 以上でデータサイズの合計が 16MB を超える場合、または、一致したリソースデータ数が 1000 を超える場合、400 Bad Request を返します。
4. その他エラーについては、2.3.2 節参照

3.3. リソース_JSON のデータ検索

・ Request

Parameters		値
Method		GET
URI	データ本体の取得(*1)	<Base URL>/v1/<Tenant ID>/<リソースパス(/\$all 利用可)>/_past.<拡張子>?<QUERY>
	HIT 数の取得	<Base URL>/v1/<Tenant ID>/<リソースパス(/\$all 利用可)>/_past/_count?<QUERY>
Headers		2.3.1 節 参照
Body		なし

(*1) 検索結果として得られるデータの順番は、<リソースパス>および<登録日時>でソートします。

◇ URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M

Tenant ID	リソース所有テナントの識別子	M
リソースパス(/\$all 利用可)	検索対象のリソースパス。記述方法は後述します。	M
拡張子	以下を指定。省略時は「.」を含め省略します。 ・json	0
QUERY	検索条件(後述)。省略時(?含む)は、条件なし(全て)として扱います。	0

(*1) M: 必須、0: オプション

- URI パラメータのクエリに対し、上記一覧に記載がなく、かつ本章の他の要求(登録、参照、更新、削除)にて許容しているものを指定した場合には、エラーとならず無視します。許容されていないものは、解析失敗としてエラー応答します。

◇ <リソースパス(/\$all 利用可)>の記述方法

- 1)リソースパスをフルパスで指定:
 - 指定したリソースパスのリソースデータを返却します。
- 2)リソースパスの途中までを指定し、その後に「/\$all」を付加:
 - 指定したパス配下全てのリソースパスのリソースデータを返却します。
 - 例
 - 「AX」「A/B」「A/B/C」の3つのリソースが存在する状態で、「A/\$all」を指定した場合、「A/B」および「A/B/C」の2つのリソースを対象とします。

Memo

対象テナント内全てのリソース_JSONを対象とした検索は以下で指定できますが、検索母数が多い場合タイムアウトするため非推奨です。

<テナントID>/\$all/_past

- ◇ \$all にて複数のリソースパスを対象にする場合、リクエストの Authorization ヘッダに格納するアクセスコードは、以下のいずれかを使用する必要があります。
1. 対象リソースパス全てに read 権を持つアクセスコードを使用する。
 - 1つでも read 権のないリソースパスを含む場合、エラー応答(401 Unauthorized)します。
 2. 対象リソースパスの上位パスに hierarchy_get 権を持つアクセスコードを使用する。
 - hierarchy_get 権を利用する際の\$all指定は、リソースパスの途中までを指定し、その後に「\$all」を付加してください。
 - 「\$all」のみが指定された場合、hierarchy_get 権は利用されず、read 権のみが利用されます。1つでも read 権のないリソースパスを含む場合、エラー応答(401 Unauthorized)します。

◇ QUERY 一覧

書式	説明
\$filter=<filter 条件>	<filter 条件>に一致するもののみを返すよう、結果を限定します。 <filter 条件>は、「プロパティ名 演算子 条件値」とし、and or で複数定義可能です。使用可能な演算子、プロパティ名は後述します。
\$top=n	検索結果として得られるデータを n 件に限定します。最大取得件数は 1000 件です。
\$skip=n	検索結果として得られたデータを n 件 skip します。検索結果として得られるデータの順番は、第一ソート key : <リソースパス>、第二ソート key : <登録日時>でソートします。
\$select=<選択 key>	<選択 key>で指定されたフィールドのデータのみを返します。 ・<選択 key>は、JSONでの name に相当し、登録データ内の任意の name を指定可能です。フィールドの階層は「.」で表現します。

	<ul style="list-style-type: none"> ・ <選択 key>は「,」区切りで複数指定できます。 ・ <code>_date/_resource_path/_data</code> は<選択 key>に使用できません。 <p>例) <code>\$select=sensor.no,sensor.name,sensor.data.temp</code></p>
<code>\$orderby=<ソート key></code> <ソート順>	<p>検索結果の並べ替え方を指定します。</p> <ul style="list-style-type: none"> ・ <ソート key>は、並べ替え基準となる情報を指定します。以下のいずれかが指定可能です。 <ul style="list-style-type: none"> <code>_resource_path</code> <code>_date</code> ・ <ソート順>は、並べ替え順を指定します。以下のいずれかが指定可能です。 <ul style="list-style-type: none"> <code>asc</code> : 昇順で並べ替えを行います。 <code>desc</code> : 降順で並べ替えを行います。 ・ <ソート key>と<ソート順>の間は、半角スペースを置きます。 ・ <ソート key> <ソート順>のペアは、「,」区切りで複数指定できます。 ・ 指定されなかった<ソート key>における並べ替え順は不定です。 ・ <code>\$orderby</code> が指定されなかった場合は、以下が指定されたものとして動作します。 <p><code>\$orderby=_resource_path asc,_date desc</code></p>

1. `/_count` が指定された場合、`$stop`、`$skip`、`$select`、`$orderby` は無視します。(`$filter` のみ併用可能)

◇ filter 条件の演算子

演算子	説明	例
<code>eq</code>	等号	<code>/temperature?\$filter=Owner eq 'Tom'</code>
<code>ne</code>	不等号	<code>/temperature?\$filter=Owner ne null</code>
<code>gt</code>	より大きい	<code>/temperature?\$filter=Floor1.Value gt 1000</code>
<code>ge</code>	以上	<code>/temperature?&\$filter=Floor1.Value ge 1000</code>
<code>lt</code>	より小さい	<code>/temperature?\$filter=Floor1.Value lt 1000</code>
<code>le</code>	以下	<code>/temperature?\$filter=Floor1.Value le 1000</code>
<code>and</code>	論理積	<code>/temperature?\$filter=Floor1.Value ge 1000 and Owner eq 'Tom'</code>
<code>or</code>	論理和	<code>/temperature?\$filter=Id eq 2 or Id eq 1</code>

1. `null` は、値が存在しないことを示します。
2. データの `name` が階層構造になっている場合は、`<name>.<name>` で表現します。
3. `(A eq 1 and B eq 1)or(A eq 2 and B eq 2)` のような表現も可能です。
 - ただしこの場合、`()` 内に `()` は定義できません。例えば、`((A eq 1 and B eq 1)or(A eq 2 and B eq 2))and(C eq 1)` はエラーとなります。
4. 正規表現によるマッチングには対応していません。
5. 文字列はシングルクォートで囲みます。シングルクォートで囲まないものは数値とみなします。
6. JSON 配列の内容を指定する場合は、`<name>.<配列 index>` で表現します。配列 `index` は数字です。

例：

検索対象データ	<pre>{ "Owners" : ["Taro", "Jiro"] }</pre>
filter 条件指定例	<code>\$filter=Owners.0 eq 'Taro'</code>

- 配列内にオブジェクト構造があり、かつ `name` が数字のみで構成されている場合、配列 `index` 指定と `name` の識別ができない場合があります。下記例を参考に配列 `index` と `name` の双方を指定してください。

検索対象データ例：

<pre>{ "data" : [{ "0" : "Taro" }, { "0" : "Jiro" }] }</pre> <pre>{ "data" : [{ "0" : "Jiro" }, { "0" : "Taro" }] }</pre> <pre>{ "data" : { "0" : "Taro" } }</pre>
--

検索例：

filter 指定条件	適合するデータ
\$filter=data.0 eq 'Taro'	、
\$filter=data.1.0 eq 'Taro'	

◇ filter 条件に利用可能なプロパティ名

プロパティ名	説明	備考
_date	登録日時	検索対象データの登録日時(*1) _date で条件指定する登録日時についてはシングルクォートを使って囲みません。
任意の name	登録データに含まれる任意の name	URI の非予約文字 (「半角英数字」、「-」、「.」、「_」、「~」以外) はパーセントエンコードしてください。

(*1) ISO8601(基本表記としてのミリ秒表現を使用)に従います(20141225T103612.001Z など)。精度はミリ秒(ミリ秒を省略した場合、0ミリ秒とみなします)です。秒とミリ秒の区切りは「.」、タイムゾーン指定は「±hhmm」形式で省略時は「Z」を添えてください。本サービスがレスポンスに格納する場合、UTC を用います。

4. Response

[データ本体(リソースデータ)の取得]

◇ 正常応答時

Parameters	値
Status-Code	200 OK
Headers	Content-Type: application/json; charset=UTF-8
Body	一致したリソースデータ

1. Body

➤ 以下の形式で、複数の登録済みデータを返却します。形式は<拡張子>に従います。

Name	Value
_resource_path	対象データのリソースパス
_date	対象データの登録日時(*1)
_data	<拡張子>に従った形式の、登録済みのデータ。

(*1) ISO8601(基本表記としてのミリ秒表現を使用)に従います(20141225T103612.001Z など)。精度はミリ秒(ミリ秒を省略した場合、0ミリ秒とみなします)です。秒とミリ秒の区切りは「.」、タイムゾーン指定は「±hhmm」形式で省略時は「Z」を添えてください。本サービスがレスポンスに格納する場合、UTC を用います。

➤ 対象リソースに検索条件に合致するデータが存在しない場合、そのリソースに関するデータは返却しません。

◇ エラー応答時

1. 対象のリソースが存在しなかった場合、404 Not Found を返します。
2. 条件に一致するリソースデータが存在しなかった場合、204 No Content を返します。
3. 条件に一致したリソースデータ数が2以上でデータサイズの合計が16MBを超える場合、または、一致したリソースデータ数が1000を超える場合、400 Bad Request を返します。

- この時、エラーレスポンスに受付可能な top 数として、acceptable_top=n を Body に格納します。
4. その他エラーについては、2.3.2 節を参照ください。

[HIT 数の取得]

◇ 正常応答時

Parameters	値	
Status-Code	200 OK	
Headers	Content-Type	text/plain
Body	一致したリソースデータ数	

1. Body

- 一致したリソースデータ数をそのまま返却します。

◇ エラー応答時については、2.3.2 節を参照ください。

3.4. リソース_JSON のデータ更新

・ Request

Parameters	値
Method	PUT
URI	過去データの更新(*1) <Base URL>/v1/<Tenant ID>/<リソースパス>/_past(<登録日時(旧)>.<拡張子>.<圧縮タイプ>?\$newdate=<登録日時(新)> &\$charset=<Body データ文字コード>&\$skip=<Body データ削除指定行>&\$numconv=<数値変換>
Headers	2.3.1 節 参照
Body	更新対象データ

(*1) 同じ登録日時を持つデータが複数存在した場合、ひとつのみ更新(どのデータになるかは不定)されます。

◇ URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M
Tenant ID	リソース所有テナントの識別子	M
リソースパス	データ更新対象のリソースパス	M
拡張子	以下のいずれかを指定します。省略時は「.」を含め省略し、この場合は json が指定されたものとみなします。 ・ json ・ csv ・ txt ・ bin	0
圧縮タイプ	Body データを圧縮して送信する際の圧縮タイプを以下で指定します。省略時は「.」を含め省略し、この場合は Body データが無圧縮とみなします。 ・ gz	0
登録日時(旧)	更新対象データの登録日時(*2)	M
登録日時(新)	更新後の登録日時(*2) ・ 省略可(\$newdate=含む)で、省略時は登録日時を更新しません。 ・ 既に指定登録日時のデータが存在するかどうかはチェックせず、上書きを行います。	0
Body データ文字コード	Body データの文字コードを指定します。拡張子が csv、txt の場合に限り有効です。	0

	<ul style="list-style-type: none"> ・ utf-8 : UTF-8 ・ shift_jis : Shift-JIS utf-8時は省略可(\$charset=含む)	
Body データ削除指定行	Body データの先頭から削除する行数を指定します。拡張子が csv の場合に限り有効です。省略することができ(\$skip=含む)、この場合は行削除を行いません。	0
数値変換	Body データ中の数値を文字列に変換するか否かを指定します。拡張子が csv の場合に限り有効です。 <ul style="list-style-type: none"> ・ true : 数値変換します。 ・ false : 数値変換しません。 true時は省略可(\$numconv=含む)	0

(*1) M: 必須、0: オプション

(*2) ISO8601(基本表記としてのミリ秒表現を使用)に従います(20141225T103612.001Z など)。精度はミリ秒(ミリ秒を省略した場合、0ミリ秒とみなします)です。秒とミリ秒の区切りは「.」、タイムゾーン指定は「±hhmm」形式で省略時は「Z」を添えてください。本サービスがレスポンスに格納する場合、UTC を用います。

- URI パラメータのクエリに対し、上記一覧に記載がなく、かつ本章の他の要求(登録、参照、検索、削除)にて許容しているものを指定した場合には、エラーとならず無視します。許容されていないものは、解析失敗としてエラー応答します。

◇ HTTP ヘッダ

Headers	値	説明	M/O(*1)
Content-Type	・ 拡張子が json または省略の場合 application/json; charset=UTF-8	固定	0
	・ 拡張子が csv の場合 text/csv; charset=<charset> <charset>は以下のいずれか。 <ul style="list-style-type: none"> ・ UTF-8 ・ Shift_JIS 		
	・ 拡張子が txt の場合 text/plain; charset=<charset> <charset>は以下のいずれか <ul style="list-style-type: none"> ・ UTF-8 ・ Shift_JIS 		
	・ 拡張子が bin の場合 Body の MIME タイプ		
	・ 圧縮タイプが gz の場合(拡張子は任意) application/gzip または application/x-gzip		
x-iotpf-request-id	リクエスト識別 ID(任意文字列)(*2)	本サービス内および通信先の影響でリクエストが欠損した場合に出力するエラー情報に利用されます。エラー情報は 11.3 節を参照ください。	0

(*1) M: 必須、0: オプション

(*2) 値に「-」のみが設定された場合は、ヘッダ設定無しと扱います。

◇ Body

拡張子に従った形式の、登録対象のデータです。詳細は 3.1 節を参照ください。

- JSON の場合、配列([])での開始は許容しません。

- Response
 - 正常応答時

Parameters	値
Status-Code	200 OK
Headers	2.3.2 節 参照
Body	なし

- エラー応答時については 2.3.2 節を参照ください。

3.5. リソース_JSON のデータ削除

- Request

Parameters	値
Method	DELETE
URL	過去データの削除 <Base URL>/v1/<Tenant ID>/<リソースパス>/_past?<QUERY>
Headers	2.3.1 節、および下記参照
Body	なし

- URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M
Tenant ID	リソース所有テナントの識別子	M
リソースパス	データ削除対象のリソースパス	M
QUERY	削除条件(後述)	M

(*1) M: 必須、O: オプション

- HTTP ヘッダ

Headers	値	説明	M/O(*1)
x-iotpf-request-id	リクエスト識別 ID(任意文字列)(*2)	本サービス内の影響でリクエストが欠損した場合に出力するエラー情報に利用されます。エラー情報は 11.3 節を参照ください。	O

(*1) M: 必須、O: オプション

(*2) 値に「-」のみが設定された場合は、ヘッダ設定無しと扱います。

- QUERY 一覧

書式	説明
\$filter=<filter 条件>	<filter 条件>に一致するもののみを削除するよう、結果を限定します。<filter 条件>は、「プロパティ名 演算子 条件値」とし、and or で複数定義可能です。使用可能な演算子、プロパティ名は後述します。

- URI パラメータのクエリに対し、上記一覧に記載がなく、かつ本章の他の要求(登録、参照、検索、更新)にて許容しているものを指定した場合には、エラーとならず無視します。許容されていないものは、解析失敗としてエラー応答します。

- ・ filter 条件の演算子
3.3 節を参照ください。
- ・ filter 条件に利用可能なプロパティ名
3.3 節を参照ください。
- ・ Response
 - ・ 正常応答時

Parameters	値
Status-Code	200 OK
Headers	2.3.2 節 参照
Body	なし

- ・ エラー応答時については 2.3.2 節を参照ください。

第4章 リソース_JSON / リソース(転送)_JSON へのデータ制御(MQTT)

リソース_JSON 及びリソース(転送)_JSON の制御が行える MQTT のインターフェースの一覧を以下に示します。

表 7: データ蓄積 / 参照インターフェース一覧

プロトコル	種別	用途	章
MQTT	リソースへのデータ登録	リソースに対して新たにデータを追加登録(転送)します。 [オプション] ・なし: 本サービスでの受信日時でデータを登録 ・日時指定: 指定された日時でデータを登録	4.1 章
	リソースデータの参照	リソースに登録されるデータを参照します。	4.2 章

4.1. リソース_JSON へのデータ登録 / リソース(転送)_JSON へのデータ転送

・ Request

Parameters	値
メッセージタイプ	PUBLISH
Topic	データ蓄積 <アクセスコード>/v1/<Tenant ID>/<リソースパス> データ転送 <アクセスコード>/v1/<Tenant ID>/_fwd/<リソースパス>
他固定ヘッダおよび可変ヘッダ	2.5.1 節 参照
ペイロード	本サービスヘッダ用および、登録対象データペイロード内先頭の「---IoT-PF<CRLF>」以降を本サービス用ヘッダとします。 Header 名と値は「<Header 名> : <値><CRLF>」形式で記述します。 本サービス用ヘッダの終了「<CRLF><CRLF>」にて判定するものとし、[<CRLF>]は、Header のものを含みます。 本サービス用ヘッダは省略することができます。

・ 本サービス用ヘッダ

Headers	値	M/O(*1)
Date	<登録日時>	0
x-iotpf-request-id	<リクエスト識別 ID(任意文字列)>(*2)	0

(*1) M: 必須、0: オプション

(*2) 本サービス内の影響でリクエストが欠損した場合に出力するエラー情報に利用されます。エラー情報は 11.3 節を参照ください。

・ パラメータ

パラメータ名	説明	M/O(*1)
アクセスコード	アクセス権限を表す文字列。管理者から入手します。	M
Tenant ID	リソース所有テナントの識別子	M
リソースパス	データ登録対象のリソースパス	M
登録日時	登録対象データに付与する登録日時(*2)。 ・省略可で、省略時は本サービスでのリクエスト受信日時を採用しま	0

	す。 ・既に指定登録日時のデータが存在するかどうかはチェックせず、上書きを行います。	
リクエスト識別 ID	本サービス内および通信先の影響でリクエストが欠損した場合に出力するエラー情報に利用されます。エラー情報は 11.3 節を参照ください。値に「-」のみが設定された場合は、設定無しと扱います。	0

(*1) M: 必須、0: オプション

(*2) ISO8601 (基本表記としてのミリ秒表現を使用)に従います (20141225T103612.001Z など)。精度はミリ秒 (ミリ秒を省略した場合、0 ミリ秒とみなします) です。秒とミリ秒の区切りは「.」、タイムゾーン指定は「±hhmm」形式で省略時は「Z」を添えます。本サービスがレスポンスに格納する場合、UTC を用います。

- ・ 登録対象のデータ
 1. JSON 形式の任意のデータ
 - 配列 ([])での開始は許容しません。
 - メンバーの順序性は保証せず、登録時の並びで参照できるとは限りません。
 - 重複した name を持つメンバは保存しません。
- ・ Response
 - ・ QoS レベルに応じたメッセージ (なし/PUBACK メッセージ/PUBREC メッセージ)

4.2. リソース_JSON / リソース (転送)_JSON のデータ参照

MQTT でのリソースデータの参照では、SUBSCRIBE メッセージを使用します。

対象の Topic (リソースパス) を SUBSCRIBE しておくことで、以下のメッセージを MQTT PUBLISH にて受信することが可能です。

プロトコル	操作	SUBSCRIBE による受信	Retain オプション
REST	リソースデータの登録	(*1)	
	リソースデータの参照	×	
	リソースデータの更新	×	
	リソースデータの削除	×	
MQTT	リソースデータの登録		
	リソースデータの参照	×	

(*1) Bulk Insert にて登録されたデータは、MQTT PUBLISH にて受信することはできません。

- ・ Request

Parameters	値
メッセージタイプ	SUBSCRIBE
固定ヘッダおよび可変ヘッダ	2.5.1 節 参照
ペイロード	Topic 名 「<アクセスコード>/v1/<Tenant ID>/<リソースパス>」 または 「<アクセスコード>/v1/<Tenant ID>/_fwd/<リソースパス>」 のリスト

- ・ パラメータ

パラメータ名	説明	M/O(*1)
アクセスコード	アクセス権限を表す文字列。管理者から入手します。	M
Tenant ID	リソース所有テナントの識別子	M
リソースパス	データ登録対象のリソースパス	M

(*1) M：必須、O：オプション

- ・ Response
 - ・ SUBACK メッセージ

第5章 リソース_Binary データ制御 (REST)

リソース_Binary に対するデータ蓄積 / リソースのデータ参照インターフェース一覧を以下に示します。

表 8 : データ蓄積 / 参照インターフェース一覧

プロトコル	種別	用途	章
REST	リソースへのデータ登録	リソースに対して新たにデータを追加登録します。 [オプション] ・なし： 本サービスでの受信日時でデータを登録 ・日時指定： 指定された日時でデータを登録	5.1 章
	リソースデータの参照	リソースに登録済みのデータの実体を参照します。 [オプション] ・なし： 最新データを返します ・日時指定： 指定日時のデータを返します	5.2 章
	リソースデータの検索	リソースに登録済みのデータを、条件を指定して検索します。検索結果として、HITした登録済みデータの情報を返します。 (検索できるのはメタデータのみです。登録したデータの実体は検索できません。) [オプション] ・QUERY にて、検索条件を指定できます	5.3 章
	リソースデータの削除	リソースに登録済みのデータを削除します [オプション] ・QUERY にて、削除条件を指定できます	5.5 章

◇ REST / MQTT の連携動作について

「リソース_Binary」においては REST で登録したリソースデータを MQTT の SUBSCRIBE により参照することはできません。

5.1. リソース_Binary へのデータ登録

・ Request

Parameters	値
Method	PUT
URI	データ蓄積 <Base URL>/v1/<Tenant ID>/_bin/<リソースパス>?\$date=<登録日時>
Headers	2.3.1 節 及び、下記を参照ください。
Body	登録対象データ

Memo

IoT Platform のデータ登録では、「存在するリソースにデータを格納する」という考え方から、Method に PUT を使用する仕様です。リソースへの POST Method は「リソースの作成 (7.1 リソースの登録)」にて使われます。

・ URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M
Tenant ID	リソース所有テナントの識別子	M
リソースパス	データ登録対象のリソースパス	M
登録日時	登録対象データに付与する登録日時(*2)。	O

	<ul style="list-style-type: none"> 省略できます(\$date=含む)。省略時は本サービスにおけるリクエスト受信日時を採用します。 すでに指定登録日時のデータが存在する場合、上書き登録を行います。 	
--	---	--

(*1) M: 必須、0: オプション

(*2) ISO8601(基本表記としてのミリ秒表現を使用)に従います(20141225T103612.001Z など)。精度はミリ秒(ミリ秒を省略した場合、0ミリ秒とみなします)です。秒とミリ秒の区切りは「.」、タイムゾーン指定は「±hhmm」形式で省略時は「Z」を添えます。本サービスがレスポンスに格納する場合、UTC を用います。

・ HTTP ヘッダ

Headers	値	補足	M/O(*1)
Content-Type	Body の MIME タイプ	各リソースデータのメタデータとして管理し、レスポンスに付与します	0
x-iotpf-meta-data1 x-iotpf-meta-data2 x-iotpf-meta-data3	任意のメタデータ	各リソースデータのメタデータとして管理し、レスポンスに付与します	0
x-iotpf-request-id	リクエスト識別 ID(任意文字列)(*2)	本サービス内の影響でリクエストが欠損した場合に出力するエラー情報に利用されます。エラー情報は 11.3 節を参照ください。	0

(*1) M: 必須、0: オプション

(*2) 値に「-」のみが設定された場合は、ヘッダ設定無しと扱います。

・ Body

1. 登録対象のデータ。

- 最大サイズ: 100MB
- REST にて登録した Binary データは、MQTT では参照できません。

・ Response

・ 正常応答時

Parameters	値
Status-Code	200
Headers	2.3.2 節 参照
Body	なし

- ・ エラー応答時については 2.3.2 節を参照ください。

5.2. リソース_Binary のデータ参照

・ Request

Parameters	値				
Method	GET				
URI	<table border="1"> <tr> <td>最新データの参照</td> <td><Base URL>/v1/<Tenant ID>/_bin/<リソースパス>/_present</td> </tr> <tr> <td>過去データの参照</td> <td><Base URL>/v1/<Tenant ID>/_bin/<リソースパス>/_past(<登録日時>)</td> </tr> </table>	最新データの参照	<Base URL>/v1/<Tenant ID>/_bin/<リソースパス>/_present	過去データの参照	<Base URL>/v1/<Tenant ID>/_bin/<リソースパス>/_past(<登録日時>)
最新データの参照	<Base URL>/v1/<Tenant ID>/_bin/<リソースパス>/_present				
過去データの参照	<Base URL>/v1/<Tenant ID>/_bin/<リソースパス>/_past(<登録日時>)				
Headers	2.3.1 節 参照				

Body	なし
------	----

・ URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節参照	M
Tenant ID	リソース所有テナントの識別子	M
リソースパス	データ参照対象のリソースパス	M
登録日時	参照対象データの登録日時(*2)。	M

(*1) M: 必須、O: オプション

(*2) ISO8601(基本表記としてのミリ秒表現を使用)に従います(20141225T103612.001Z など)。精度はミリ秒(ミリ秒を省略した場合、0ミリ秒とみなします)です。秒とミリ秒の区切りは「.」、タイムゾーン指定は「±hhmm」形式で省略時は「Z」を添えます。本サービスがレスポンスに格納する場合、UTC を用います。

・ Response

・ 正常応答時

Parameters	値	説明	
Status-Code	200	成功(全体取得)	
	206	成功(部分取得)	
Headers	field name	Value	M/O(*1)
	Content-Type	Body の MIME タイプ。 リソースデータ登録時に指定した値を設定します。 リソースデータ登録時に指定しなかった場合、「application/octet-stream」を設定します。	M
	x-iotpf-meta-data1 x-iotpf-meta-data2 x-iotpf-meta-data3	メタデータ。 リソースデータ登録時に設定した場合に付与します。	O
	Body	データ本体	

(*1) M: 必須、O: オプション

・ エラー応答時

1. 参照対象のリソースが存在しなかった場合、404 を返します。
2. 条件に一致したリソースデータが存在しなかった場合、204 を返します。
3. その他エラーについては、2.3.2 節を参照ください。

5.3. リソース_Binary のデータ検索

・ Request

Parameters	値	
Method	GET	
URI	データリストの取得(*1)	<Base URL>/v1/<Tenant ID>/_bin/<リソースパス(\$all 利用可)>/_past?<QUERY>
	HIT 数の取得	<Base URL>/v1/<Tenant ID>/_bin/<リソースパス(\$all 利用可)>/_past/_count?<QUERY>
Headers	2.3.1 参照	
Body	なし	

(*1) 検索結果として得られるデータの順番は、<登録日時>でソートします。

・ URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M
Tenant ID	リソース所有テナントの識別子	M
リソースパス	検索対象のリソースパス。記述方法は後述します。	M
QUERY	検索条件（後述）。省略時(?含む)は、条件なし(全て)として扱います。	O

(*1) M：必須、O：オプション

- ・ <リソースパス(/\$all 利用可)>の記述方法
 - 1) リソースパスをフルパスで指定：
 - 指定したリソースパスのリソースデータ情報を返却します。
 - 2) リソースパスの途中までを指定し、その後に「/\$all」を付加：
 - 指定したパス配下全てのリソースパスのリソースデータ情報を返却します。
 - 例
 - 「AX」「A/B」「A/B/C」の3つのリソースが存在する状態で、「A/\$all」を指定した場合、「A/B」および「A/B/C」の2つのリソースを対象とします。
 - 対象テナント内の全ての Binary リソースを対象にする場合、次の通り設定します。
 - <テナント ID>/_bin/\$all/_past
- ・ \$all にて複数のリソースパスを対象にする場合、リクエストの Authorization ヘッダに格納するアクセスコードは、以下のいずれかを使用する必要があります。
 1. 対象リソースパス全てに read 権を持つアクセスコードを使用する。
 - 1つでも read 権のないリソースパスを含む場合、エラー応答(401)します。
 2. 対象リソースパスの上位パスに hierarchy_get 権を持つアクセスコードを使用する。
 - hierarchy_get 権を利用する際の\$all 指定は、リソースパスの途中までを指定し、その後に「\$all」を付加してください。
 - 「\$all」のみが指定された場合、hierarchy_get 権は利用されず、read 権のみが利用されます。1つでも read 権のないリソースパスを含む場合、エラー応答(401)します。

・ QUERY 一覧

書式	説明
\$filter=<filter 条件>	<filter 条件>に一致するもののみを返すよう、結果を限定します。 <filter 条件>は、「プロパティ名 演算子 条件値」とし、and で複数定義可能。 使用可能な演算子、プロパティ名は後述します。
\$top=n	検索結果として得られるデータを n 件に限定します。 最大取得件数は 1000 件です。
\$skip=n	検索結果として得られたデータを n 件 skip します。検索結果として得られるデータの順番は、第一ソート key：<リソースパス>、第二ソート key：<登録日時>でソートします。

1. /_count が指定された場合、\$top、\$skip は無視する（\$filter のみ併用可能）

・ filter 条件の演算子

演算子	説明	例
eq	等号	/image?\$filter=_date eq 20160101T000000.001Z
gt	より大きい	/image?\$filter=_date gt 20160101T000000.001Z
ge	以上	/image?\$filter=_date ge 20160101T000000.001Z
lt	より小さい	/image?\$filter=_date lt 20160101T000000.001Z
le	以下	/image?\$filter=_date le 20160101T000000.001Z
and	論理積	/image?\$filter=_date ge 20160101T000000.001Z and _date le 20160529T101010.001Z

1. and は上記例ののように、一度のみ使用できます。

- filter 条件に利用可能なプロパティ名

プロパティ名	説明	備考
_date	登録日時	検索対象データの登録日時(*1) _dateで条件指定する登録日時についてはシングルクォートで囲みません

(*1) ISO8601(基本表記としてのミリ秒表現を使用)に従います(20141225T103612.001Z など)。精度はミリ秒(ミリ秒を省略した場合、0ミリ秒とみなします)です。秒とミリ秒の区切りは「.」、タイムゾーン指定は「±hhmm」形式で省略時は「Z」を添えます。本サービスがレスポンスに格納する場合、UTCを用います。

- Response

[データリストの取得]

- 正常応答時

Parameters	値
Status-Code	200
Headers	Content-Type: application/json; charset=UTF-8
Body	一致したリソースデータのメタデータ情報

1. Body

- 以下の形式で、複数の登録済みデータの情報を JSON 形式で返却します。

Name	Value	M/O(*1)
name	リソースデータの名前。URI のパスで表現。 _bin/<リソースパス>/_past(<登録日時>)	M
content_type	リソースデータの MIME タイプ。 リソースデータ登録時に指定した Content-Type を設定。リソースデータ登録時に未指定の場合は、「application/octet-stream」を設定します	M

(*1) M: 必須、O: オプション

- 対象リソースに検索条件に合致するデータが存在しない場合、そのリソースに関するデータは返却しません
- エラー応答時
 - 対象のリソースが存在しなかった場合、404 を返します。
 - 条件に一致するリソースデータが存在しなかった場合、204 を返します
 - 条件に一致したリソースデータ数が 1000 を超える場合、400 を返します。
 - この時、エラーレスポンスに受付可能な top 数として、acceptable_top=n を Body に格納します
 - その他エラーについては、2.3.2 節参照

[HIT 数の取得]

- 正常応答時

Parameters	値
Status-Code	200
Headers	Content-Type: text/plain; charset=UTF-8
Body	一致したリソースデータ数

1. Body

- 一致したリソースデータ数をそのまま返却します。

- ・ エラー応答時には、2.3.2 節を参照ください。

5.4. リソース_Binary のデータ更新

リソース_Binary を更新するためのインターフェースはありません。

ただし、5.1 節にあるリソース_Binary のデータ登録を用い、既に登録をした同時刻を指定して登録することにより、上書き更新が可能です。

5.5. リソース_Binary のデータ削除

- ・ Request

Parameters	値
Method	DELETE
URI	過去データの削除 <Base URL>/v1/<Tenant ID>/_bin/<リソースパス>/_past?<QUERY>
Headers	2.3.1 節 参照
Body	なし

- ・ URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M
Tenant ID	リソース所有テナントの識別子	M
リソースパス	データ削除対象のリソースパス	M
QUERY	削除条件(後述)	M

(*1) M: 必須、O: オプション

- ・ HTTP ヘッダ

Headers	値	補足	M/O(*1)
x-iotpf-request-id	リクエスト識別 ID(任意文字列)(*2)	本サービス内の影響でリクエストが欠損した場合に出力するエラー情報に利用されます。エラー情報は 11.3 節を参照ください。	O

(*1) M: 必須、O: オプション

(*2) 値に「-」のみが設定された場合は、ヘッダ設定無しと扱います。

- ・ QUERY 一覧

書式	説明
\$filter=<filter 条件>	<filter 条件>に一致するもののみを削除するよう、結果を限定します。 <filter 条件>は、「プロパティ名 演算子 条件値」とし、and で複数定義できます。使用可能な演算子、プロパティ名は後述します。

- ・ filter 条件の演算子
3.3 節 参照

- ・ filter 条件に利用可能なプロパティ名
3.3 節 参照

- Response
 - 正常応答時

Parameters	値
Status-Code	200
Headers	2.3.2 節 参照
Body	なし

- エラー応答時については 2.3.2 節を参照ください。

第6章 リソース_Binary データ制御(MQTT)

リソース_Binary へのデータ蓄積 / リソースのデータ参照インターフェース一覧を以下に示します。

表 9 : データ蓄積 / 参照インターフェース一覧

プロトコル	種別	用途	章
MQTT	リソースへのデータ登録	リソースに対して新たにデータを追加登録します。	6.1 章
	リソースデータの参照	リソースに登録されるデータを参照します。	6.2 章

6.1. リソース_Binary へのデータ登録

・ Request

Parameters	値
メッセージタイプ	PUBLISH
Topic	<アクセスコード>/v1/<Tenant ID>/_bin/<リソースパス>
他固定ヘッダおよび可変ヘッダ	2.5.1 節 参照
ペイロード	登録対象データ

・ パラメータ

パラメータ名	説明	M/O(*1)
アクセスコード	アクセス権限を表す文字列。管理者から入手します。	M
Tenant ID	リソース所有テナントの識別子	M
リソースパス	データ登録対象のリソースパス	M

(*1) M : 必須、O : オプション

・ 登録対象のデータ

1. 最大サイズ: 16MB

・ Response

- ・ QoS レベルに応じたメッセージ(なし/PUBACK メッセージ/PUBREC メッセージ)

6.2. リソース_Binary のデータ参照

MQTT でのリソースデータの参照では、SUBSCRIBE メッセージを使用します。

対象の Topic (リソースパス) を SUBSCRIBE しておくことで、以下のメッセージを MQTT PUBLISH にて受信することが可能です。

REST IF にて登録した Binary データは、MQTT では参照できません。

プロトコル	操作	SUBSCRIBE による受信	Retain オプション
REST	リソースデータの登録	×	
	リソースデータの参照	×	
	リソースデータの更新	×	
	リソースデータの削除	×	
MQTT	リソースデータの登録		
	リソースデータの参照	×	

- Request

Parameters	値
メッセージタイプ	SUBSCRIBE
固定ヘッダおよび可変ヘッダ	2.5.1 節 参照
ペイロード	Topic 名「<アクセスコード>/v1/<Tenant ID>/_bin/<リソースパス>」のリスト

- パラメータ

パラメータ名	説明	M/O(*1)
アクセスコード	アクセス権限を表す文字列。管理者から入手する。	M
Tenant ID	リソース所有テナントの識別子	M
リソースパス	データ登録対象のリソースパス	M

(*1) M: 必須、O: オプション

- Response

- SUBACK メッセージ

第7章 リソース制御(REST)

リソース制御が行えるインターフェースの一覧を以下に示します。

リソース制御 API を利用するにはリソース及びアクセスコードをサービスポータルにて事前に設定する必要があります。2.2 節を参照願います。

表 10：リソース制御インターフェース一覧

種別	用途	章
リソースの登録	リソースを新規に登録する。 [オプション] ・ Body なし： リソースのみ作成 ・ Body にメタデータ格納： リソース登録時にメタデータの初期設定を行う	7.1 章
リソースのメタデータ参照	リソースのメタデータを参照する。 [オプション] ・ リソースパスの一部を\${} で指定： 指定パス配下のリソースのメタデータを返す	7.2 章
リソースのメタデータ更新	リソースのメタデータを更新する。 [オプション] なし	7.3 章
リソースの削除	リソースを削除する。 [オプション] なし	7.4 章

7.1. リソースの登録

・ Request

Parameters		値
Method		POST
URI	JSON	蓄積 <Base URL>/v1/<Tenant ID>/<リソースパス>
		転送 <Base URL>/v1/<Tenant ID>/_fwd/<リソースパス>
	Binary	<Base URL>/v1/<Tenant ID>/_bin/<リソースパス>
Headers		2.3.1 節 および下記参照
Body		メタデータ

・ URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M
Tenant ID	リソース所有テナントの識別子	M
リソースパス	登録対象のリソースパス	M

(*1) M：必須、O：オプション

・ HTTP ヘッダ

Headers	値	補足	M/O(*1)
Content-Type	application/json; charset=UTF-8	固定	O

(*1) M：必須、0：オプション

- Body
 - 次の通り、JSON 形式でメタデータを格納することでメタデータの初期化ができます。

Parameters		形式	M/O(*1)	説明	最大長
resource			M	開始タグ	
	retention_period	数値	0(*2)	リソースデータの保存期間(日) ・設定されていない場合、保存期間は1日とします。 ・値の範囲は1～9999	9999
fwd_info			0(*3)	転送先情報	
	http		M	HTTP の転送先情報	
	method	string	M	“ GET ”, “ POST ”, “ PUT ”, “ DELETE ”, “ HEAD ”, “ OPTIONS ”, “ TRACE ” のいずれか	7 文字
	uri	string	M	URI。 “ http:// ~ ” または “ https:// ~ ”	256 文字
	basic_auth_id	string	0	Basic 認証用の ID	20 文字
	basic_auth_pass	string	0	Basic 認証用のパスワード	20 文字
	header_fields	(配列)	0	HTTP ヘッダ (配列の要素は最大 10 個)	
	field_name	string	M	ヘッダフィールド名。“ : ” は含みません。	20 文字
	field_value	string	M	上記ヘッダフィールドに格納する値	512 文字

(*1) M：必須、0：オプション。各子要素の M/O は、親要素を設定した場合の可否を表します。

(*2) リソース_JSON、リソース_Binary の場合に限り有効です。

(*3) 対象リソースが JSON 形式の転送リソースの場合のみ設定できます。

- Response
 - 正常応答時

Parameters		値
Status-Code		201
Headers	Location	<Base URL>/v1/<Tenant ID>/<リソースパス> or <Base URL>/v1/<Tenant ID>/_fwd/<リソースパス> or <Base URL>/v1/<Tenant ID>/_bin/<リソースパス>
Body		なし

- エラー応答時については 2.3.2 節を参照ください。

7.2. リソースのメタデータ参照

- Request

Parameters				値
Method				GET
URI	データリストの取得	JSON	蓄積	<Base URL>/v1/<Tenant ID>/<リソースパス(/\$all)>/_resources?<QUERY>

	HIT 数の取得	転送	<Base URL>/v1/<Tenant ID>/_fwd/<リソースパス (</\$all>)/_resources?<QUERY>
		バイナリ	<Base URL>/v1/<Tenant ID>/_bin/<リソースパス (</\$all>)/_resources?<QUERY>
		蓄積	<Base URL>/v1/<Tenant ID>/<リソースパス (</\$all>)/_resources/_count?<QUERY>
		転送	<Base URL>/v1/<Tenant ID>/_fwd/<リソースパス (</\$all>)/_resources/_count?<QUERY>
		バイナリ	<Base URL>/v1/<Tenant ID>/_bin/<リソースパス (</\$all>)/_resources/_count?<QUERY>
Headers			2.3.1 節 参照
Body			なし

・ URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M
Tenant ID	リソース所有テナントの識別子	M
QUERY	参照条件(後述)。省略時(?含む)は、条件なし(全て)として扱う。	0

(*1) M: 必須、0: オプション

・ <リソースパス(</\$all>)>の記述方法

1. リソースパスをフルパスで指定:

- 指定したリソースパスのメタデータを返却します。

2. リソースパスの途中までを指定し、その後ろに「</\$all>」を付加:

- 指定したパス配下全てのリソースパスのメタデータを返却します。

➢ 例

- 「A」「AX」「A/B」「A/B/C」の4つのリソースが存在する状態で、「A/</\$all>」を指定した場合、「A/B」および「A/B/C」の2つのリソースを対象とします。

- 各リソース種別の全てのリソースパスを対象にする場合、次の通り設定します。

- リソース_JSON: <テナント ID>/</\$all>/_resources
- リソース(転送)_JSON: <テナント ID>/_fwd/</\$all>/_resources
- リソース_Binary: <テナント ID>/_bin/</\$all>/_resources

異なるリソース種別のリソースパスを一度に参照することはできません。

・ QUERY 一覧

書式	説明
\$top=n	検索結果として得られるデータを n 個に限定します(最大値は 1,000)
\$skip=n	検索結果として得られたデータを n 件 skip します。検索結果として得られるデータの順番は、リソースパスにてソートします。

1. /_count が指定された場合、\$top、\$skip は無視する

・ Response

[データリストの取得]

・ 正常応答時

Parameters	値
Status-Code	200
Headers	Content-Type: application/json; charset=UTF-8
Body	対象リソースのメタデータ(複数)

1. Body

- 次の通り、各リソースのメタデータを JSON オブジェクトとし、JSON 配列形式で複数格納します。
- 各リソースの情報は、リソースパスでソートします（昇順）

Parameters	形式	M/O(*1)	説明	最大長
resources		M	開始タグ	
resource_path	string	M	リソースパス	128 文字
retention_period	数値	0	リソースデータの保存期間 ・設定されていない場合、保存期間は 1 日とします。 ・値は 1-9999 とします。	9999
last_modified	日時	M	最新データの<登録日時>(*2)(*3)	20 文字
fwd_info		O(*4)	転送先情報	
http		M	HTTP の転送先情報	
method	string	M	“ GET ”, “ POST ”, “ PUT ”, “ DELETE ”, “ HEAD ”, “ OPTIONS ”, “ TRACE ” のいずれか。	6 文字
uri	string	M	URI。 “ http:// ~ ” または “ https:// ~ ”	256 文字
basic_auth_id	string	0	Basic 認証用の ID	20 文字
basic_auth_pass	string	0	Basic 認証用のパスワード	20 文字
header_fields	(配列)	0	HTTP ヘッダ (配列の要素は最大 10 個まで設定できます)	
field_name	string	M	ヘッダフィールド名。“ : ” は含まない。	20 文字
field_value	string	M	上記ヘッダフィールドに格納する値	512 文字

(*1) M: 必須、0: オプション。各子要素の M/O は、親要素を設定した場合の有無を表します。

(*2) ISO8601(基本表記としてのミリ秒表現を使用)に従います(20141225T103612.001Z など)。精度はミリ秒(ミリ秒を省略した場合、0ミリ秒とみなします)です。秒とミリ秒の区切りは「.」、タイムゾーン指定は「±hhmm」形式で省略時は「Z」を添えます。本サービスがレスポンスに格納する場合、UTC を用います。

(*3) リソースデータ登録/更新にて<登録日時>を指定できるため、サーバの最終更新日時と最新データの<登録日時>は一致しないケースがあります。

(*4) 対象リソースが JSON 形式の転送リソースの場合のみ設定します。

- パラメータに設定値がない場合、そのパラメータはレスポンスに格納されません。
 - リソース内にデータが存在しない場合、last_modified パラメータも格納されません。
- リクエストの Authorization ヘッダに指定されるアクセスコードが、対象となるリソースへの list 権を持っていない場合、参照することはできません。

・ エラー応答時

1. 条件に一致するリソースが 1 つも存在しなかった場合、404 を返します。
2. 条件に一致したリソース数が 1000 を超える場合、400 を返します。このときのエラーレスポンスの Body 部に、受付可能な top 数として acceptable_top=<n>を格納します。
3. その他エラーについては、2.3.2 節を参照ください。

[HIT 数の取得]

- ・ 正常応答時

Parameters	値
Status-Code	200
Headers	Content-Type
	text/plain

Body	一致したリソース数
------	-----------

1. Body

- 一致したリソース数をそのまま返却します。
- リクエストの Authorization ヘッダに格納したアクセスコードが list 権限を持たないリソースは、カウント対象としません。
- ・ エラー応答時については、2.3.2 節を参照ください。

7.3. リソースのメタデータ更新

・ Request

Parameters			値
Method			PUT
URI	JSON	蓄積	<Base URL>/v1/<Tenant ID>/<リソースパス>/_resources
		転送	<Base URL>/v1/<Tenant ID>/_fwd/<リソースパス>/_resources
	バイナリ		<Base URL>/v1/<Tenant ID>/_bin/<リソースパス>/_resources
Headers			2.3.1 節 および下記参照
Body			メタデータ

・ URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M
Tenant ID	リソース所有テナントの識別子	M
リソースパス	メタデータ更新対象のリソースパス	M

(*1) M: 必須、O: オプション

・ HTTP ヘッダ

Headers	値	補足	M/O(*1)
Content-Type	application/json; charset=UTF-8	固定	O

(*1) M: 必須、O: オプション

・ Body

1. 7.1 節を参照ください。
2. 対象リソースのメタデータを、更新データで全て上書きします。
 - 更新前に設定パラメータとして A,B が登録されており、更新で A だけを指定した場合、B は削除指示と判断して B を削除します。

・ Response

・ 正常応答時

Parameters	値
Status-Code	200
Headers	2.3.2 節 参照
Body	なし

- ・ エラー応答時については 2.3.2 節を参照ください。

7.4. リソースの削除

・ Request

Parameters	値
Method	DELETE

URI	JSON	蓄積	<Base URL>/v1/<Tenant ID>/<リソースパス>
		転送	<Base URL>/v1/<Tenant ID>/_fwd/<リソースパス>
	バイナリ	<Base URL>/v1/<Tenant ID>/_bin/<リソースパス>	
Headers			2.3.1 節 参照
Body			なし

- URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節参照	M
Tenant ID	リソース所有テナントの識別子	M
リソースパス	削除対象のリソースパス	M

(*1) M: 必須、O: オプション

- Response
 - 正常応答時

Parameters	値
Status-Code	204
Headers	2.3.2 節 参照
Body	なし

- エラー応答時
 - 削除対象リソースにアクセスコードが紐付いていた場合削除できません。この場合、423 を返却します。
 - その他エラーについては 2.3.2 節を参照ください。

第8章 アクセスコード制御(REST)

アクセスコード制御のインターフェースを以下に示します。プロトコルは全て REST(HTTP)とします。

表 11：アクセスコード制御インターフェース一覧

種別	用途	章
アクセスコードの登録	アクセスコードを新規登録します。 [オプション] なし	8.1 章
アクセスコード情報の参照	アクセスコードの情報を参照(検索)します。 [検索オプション] ・アクセスコード指定 ・リソースパス指定(前方一致検索可)	8.2 章
アクセスコード情報の更新	アクセスコードの情報を更新します。 [オプション] なし	8.3 章
アクセスコードの削除	アクセスコードを削除します。 [オプション] なし	8.4 章

8.1. アクセスコードの登録

Request

Parameters	値
Method	POST
URI	<Base URL>/v1/<Tenant ID>/_access_codes/<Access Code>
Headers	2.3.1 節、及び下記を参照ください。
Body	アクセスコード情報

URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M
Tenant ID	対象テナントの識別子	M
Access Code	アクセスコード	M

(*1) M：必須、O：オプション

HTTP ヘッダ

Headers	値	補足	M/O(*1)
Content-Type	application/json; charset=UTF-8	固定	M

(*1) M：必須、O：オプション

Body： アクセスコード情報

次の通り、JSON 形式でアクセスコード情報を登録します。

Parameters	形式	M/O(*1)	説明	最大長
access_code		M	開始タグ	
permissions		M	権限情報	
ip_filter	string	0	サービスポータルで定義したアクセス	35 文字

		(配列)		制限(アクセスコード)設定の IP アドレス範囲情報を指定してください。未定義の IP アドレス範囲情報を設定した場合、エラーレスポンス(400)を返却します。 ・ [“ 開始 IP アドレス ” , “ 終了 IP アドレス ”] の形式で 2 つの IP アドレスを指定できます。 ・ 1 つの IP アドレスを設定する場合、開始 IP アドレスと終了 IP アドレスを同じアドレスにしてください。 ・ 「 開始 IP アドレス > 終了 IP アドレス (IP アドレスを 32bit とみなして比較した場合) 」 となる設定はできません。	
	resource_operations	(配列)	M	リソースに紐づく権限情報	
	resource_path	string	M	リソースパス	128 文字
	operations	string (配列)	M	“ hierarchy_get ” , “ hierarchy_put ” , “ create ” , “ read ” , “ update ” , “ delete ” , “ list ” のいずれか。配列で複数指定できます。	72 文字
	certification_info		0	クライアント証明書情報	
	certification	string	M	クライアント証明書 ・ PEM 形式 「 -----BEGIN CERTIFICATE----- 」 , 「 -----END CERTIFICATE----- 」 を含めてください。また、改行コードは「 \n 」として登録してください。	10000 文字
	certificate_usage	string	M	クライアント証明書の利用用途。 以下を指定してください。 ・ auth : クライアント認可で使用する。	4 文字
	protocols	string (配列)	0	プロトコル指定情報 “ http ” , “ https ” , “ mqtt ” , “ mqtt_s ” のいずれか。配列で複数指定できます。	29 文字

(*1) M : 必須、0 : オプション。各子要素の M/O は、親要素を設定した場合の可否を表します。

1. 権限割り当てのないアクセスコード情報の登録はエラーとします。
 - すなわち、resource_operations に 1 つ以上の resource_path、operations を設定する必要があります。最大 1000 組設定できます。
 - 設定可能な権限設定パターンを以下の表に示します。

パターン	権限				
	create	read, hierarchy_get(*1)	update, hierarchy_put(*1)	delete	list
1					
2					
3					
4					
5					
6					
7					
8					

9					
10					
11					

(*1) どちらか、または両方の権限を設定できます。

- create/delete/list/hierarchy_get/hierarchy_put は、指定したリソースパス配下全てのリソースパスへの権限として扱います。
- create/delete/list/hierarchy_get/hierarchy_put 権を、階層の異なるパスに設定した場合、より上位パスの権限に従って動作します。
 - 例：
 - リソースパス「A/B」に create,delete,list,hierarchy_get.hierarchy_put 権、
 - リソースパス「A/B/C」に list 権のみを設定した場合、
 - リソースパス「A/B/C/D」には、create,delete,list,hierarchy_get,hierarchy_put が可能
- certification_info を設定する場合には、certification、certificate_usage の両方の設定が必ず必要です。ない場合にはエラー応答(400)を返します。
- protocols の設定について、以下に説明します。

パターン	動作
“ protocols ” のメンバなし	全てのプロトコルが指定されたものと解釈します。
“ protocols ”のメンバは存在するが、値がない。 ・ "protocols":[] ・ "protocols":[""]	フォーマットエラーとしてエラー応答(400)します。

- Response
 - 正常応答時

Parameters	値
Status-Code	201
Headers	Location
Body	なし

- エラー応答時については、2.3.2 節を参照ください。

8.2. アクセスコードの参照

- Request

Parameters	値
Method	GET
URI	データリストの取得 HIT数の取得
Headers	2.3.1 節 参照
Body	なし

- URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M

Tenant ID	対象テナントの識別子	M
QUERY	検索条件(後述)。省略時(?含む)は、条件なし(全て)として扱います。	0

(*1) M: 必須、0: オプション

・ QUERY 一覧

書式	説明
\$filter=<filter 条件>	<filter 条件>に一致するもののみを返すよう、結果を限定します。 <filter 条件>は、「プロパティ名 演算子 条件値」とし、and で複数定義可能。使用可能な演算子、プロパティ名は後述します。
\$top=n	検索結果として得られるデータを n 件に限定します。最大は 1000 件まで指定できます。
\$skip=n	検索結果として得られたデータを n 件 skip します。検索結果として得られるデータの順番は、アクセスコードでソートします。

1. /_count が指定された場合、\$top、\$skip は無視します。(\$filter のみ併用可能)

・ filter 条件の演算子

演算子	説明	例
eq	等号	\$filter=_resource_path eq 'hoge'

・ filter 条件に利用可能なプロパティ名

プロパティ名	説明	備考
_resource_path	リソースパス	「/」も含めてパーセントエンコードは不要です

・ filter 条件のサポート関数

Function	説明	例
bool startswith(string p0, string p1)	前方一致	\$filter=startswith(_resource_path, 'hoge') eq true

1. false 指定は未対応です。
2. \$filter 条件直下で _resource_path eq hoge と指定されたときは完全指定となり、startswith() を利用して指定された際は前方一致指定となります。

・ Response

[データリストの取得]

・ 正常応答時

Parameters	値
Status-Code	200
Headers	Content-Type: application/json; charset=UTF-8
Body	一致したアクセスコード情報

1. Body

- 次の通り、各アクセスコードの情報を JSON オブジェクトとし、JSON 配列形式で複数格納します。
- 各アクセスコード情報は、アクセスコード(access_code)でソートします。(昇順)

Parameters	形式	説明	最大長
access_codes	(配列)	開始タグ	
access_code	string	アクセスコード	48 文字
permissions		権限情報	
ip_filter	string (配列)	アクセス制限定義を行う許可 IP アドレスの範囲情報。[“開始 IP アドレス”, “終了 IP アド	35 文字

			レス]の形式で2つのIPアドレスを指定し ます。	
		resource_operations	(配列) リソースに紐づく権限情報	
		resource_path	string リソースパス	128文字
		operations	string (配列) “ hierarchy_get ”、“ hierarchy_put ”、 “ create ”、“ read ”、“ update ”、“ delete ”、 “ list ”のいずれか。配列で複数指定できます。	72文字
		certification_info	クライアント証明書情報	
		certification	string クライアント証明書 ・ PEM形式 「 -----BEGIN CERTIFICATE----- 」, 「 -----END CERTIFICATE----- 」を含みます。また、改行コ ードは「 \n 」として表示されます。	10000文字
		certificate_usage	string クライアント証明書の利用用途。 以下を指定してください。 ・ auth : クライアント認可で使用する。	4文字
		protocols	string (配列) プロトコル指定情報 “ http ”、“ https ”、“ mqtt ”、“ mqtt_s ”のい ずれか。配列で複数指定できます。	29文字

- パラメータに設定値がない場合、レスポンスにはパラメータそのものを格納しませ
ん。
- リクエストの Authorization ヘッダに格納したアクセスコードの権限を超える権限
を持つアクセスコードは、レスポンスに格納しません。
- アクセスコード登録時に“ protocols ”の指定が無かった場合には、“ protocols ”は
通知されません。

・ エラー応答時

1. 条件に一致するアクセスコード情報が存在しなかった場合、404 を返します。
2. Authorization ヘッダのアクセスコードが登録されていない場合、401 を返します。
3. 条件に一致したアクセスコード情報数が1000を超える場合、400 を返します。
 - この時、エラーレスポンスに受付可能な top 数として、acceptable_top=n を Body に格
納します。
4. その他エラーについては、2.3.2 節を参照ください。

[HIT 数の取得]

・ 正常応答時

Parameters	値
Status-Code	200
Headers	Content-Type text/plain
Body	一致したアクセスコード数

1. Body

- 一致したアクセスコード数をそのまま返却
- リクエストの Authorization ヘッダに格納したアクセスコードの権限を超える権限
を持つアクセスコードは、カウント対象としません

・ エラー応答時については、2.3.2 節を参照ください。

8.3. アクセスコードの更新

・ Request

Parameters	値
Method	PUT
URI	<Base URL>/v1/<Tenant ID>/_access_codes/<Access Code>
Headers	2.3.1 節 および下記参照
Body	アクセスコード情報

・ URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M
Tenant ID	対象テナントの識別子	M
Access Code	アクセスコード	M

(*1) M: 必須、O: オプション

・ HTTP ヘッダ

Headers	値	補足	M/O(*1)
Content-Type	application/json; charset=UTF-8	固定	M

(*1) M: 必須、O: オプション

・ Body: アクセスコード情報

1. 8.1 節を参照ください。
2. 対象のアクセスコード情報を、更新データで全て上書きします。
 - 更新前に設定パラメータとして A, B が登録されており、更新で A だけを指定した場合、B は削除指示と判断して B を削除します。

・ Response

・ 正常応答時

Parameters	値
Status-Code	200
Headers	2.3.2 節 参照
Body	なし

- ・ エラー応答時には、2.3.2 節を参照ください。

8.4. アクセスコードの削除

・ Request

Parameters	値
Method	DELETE
URI	<Base URL>/v1/<Tenant ID>/_access_codes/<Access Code>
Headers	2.3.1 節 参照
Body	なし

・ URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M
Tenant ID	対象テナントの識別子	M

Access Code	アクセスコード	M
-------------	---------	---

(*1) M：必須、O：オプション

- ・ Response
 - ・ 正常応答時

Parameters	値
Status-Code	204
Headers	2.3.2 節 参照
Body	なし

- ・ エラー応答時
 1. 削除対象のアクセスコードがイベントに紐付いていた場合、アクセスコード削除不可とし、ステータスコード 423 と、紐付いてるイベント ID を返します。
 2. その他エラーについては 2.3.2 節を参照ください。

第9章 イベント制御(REST)

イベント制御のインターフェースを以下に示します。プロトコルは全て REST(HTTP)とします。

表 12：イベント制御インターフェース一覧

種別	用途	章
イベントの登録	イベントを新規登録します。 [オプション] なし	9.1 章
イベント情報の参照	イベントの情報を参照(検索)します。 [検索オプション] ・イベント ID 指定 ・リソースパス指定(前方一致検索可)	9.2 章
イベント情報の更新	イベントの情報を更新します。 [オプション] なし	9.3 章
イベントの削除	イベントを削除します。 [オプション] なし	9.4 章

9.1. イベントの登録

・ Request

Parameters	値
Method	POST
URI	<Base URL>/v1/<Tenant ID>/_events
Headers	2.3.1 節、および下記を参照ください
Body	イベント情報

・ URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M
Tenant ID	対象テナントの識別子	M

(*1) M：必須、O：オプション

・ HTTP ヘッダ

Headers	値	補足	M/O(*1)
Content-Type	application/json; charset=UTF-8	固定	M

(*1) M：必須、O：オプション

・ Body： イベント情報

次の通り、JSON 形式でイベント情報を登録する。

Parameters	形式	M/O(*1)	説明	最大長
event		M	開始タグ	
conditions		M	イベント条件	
targets		M	対象	
resource_path	string	M	対象のリソースパス	128 文字

					バイナリデータへの操作の場合は対象リソースパスを “_bin/ ” から設定してください。	
		operations	string (配列)	M	対象のリソースデータ操作 (アクセスコードのアクセス権とは異なります。) ・通常のリソースの場合、 [“ create ” , “ update ”] を指定してください。 ・リソース_Binary の場合、 “ create ” を指定してください。	33 文字
		read_access_code	string	M	対象リソースパスの read 権または hierarchy_get 権 を持つアクセスコード	48 文字
		notification_condition		0	通知条件	
		start_time	日時 または時刻	0(*2)	開始日時または、時刻(*3)	20 文字
		end_time	日時 または時刻	0(*4)	終了日時または、時刻(*3)	20 文字
		body_conditions		0	データボディ条件 Binary リソースに対して は設定不可	
		path_type	string	M	path の形式 “ JSONPath ” のみ指定 できます	8 文字
		path	string	M	ボディ要素を指定する ためのパス	1902 文字
		comparing_operator	string	M	比較演算子 “ eq ” (=), “ ne ” (≠), “ gt ” (>), “ ge ” (≥), “ lt ” (<), “ le ” (≤), “ substring_of ” (部分一 致)のいずれか。(*5)	12 文字
		value	string または 数値	M	比較対象の値。	文字列の場合、 1 ~ 128 文字 整数の場合、 -2,147,483,648 ~ 2,147,483,647 実数の場合、 整数部分 10 桁、 小数部分 5 桁
		awake_condition		0	イベント抑止解除条件。 Binary リソースに対して は設定できません。 本設定を行うと、	

					notification_condition を満たしイベント通知実施後、イベント抑止状態になります。イベント抑止状態を解除するための条件をここで設定してください。	
		body_conditions		M	データボディ条件	
		path_type	string	M	pathの形式。 "JSONPath"のみ指定できます。	8文字
		path	string	M	ボディ要素を指定するためのパス	1902文字
		comparing_operator	string	M	比較演算子。 "eq" (=), "ne" (), "gt" (>), "ge" (), "lt" (<), "le" (), "substring_of" (部分一致)のいずれか。(*5)	12文字
		value	string または 数値	M	比較対象の値。	文字列の場合、 1~128文字 整数の場合、 -2,147,483,648 ~2,147,483,647 実数の場合、 整数部分10桁、 小数部分5桁
		notification	(*6)	M	通知内容	
		http		0	HTTP通知設定	
		method	string	M	"GET", "POST", "PUT", "DELETE", "HEAD", "OPTIONS", "TRACE"の いずれか。	7文字
		uri	string	M	URI。 "http://~"または "https://~"	256文字
		basic_auth_id	string	0	Basic認証用のID	20文字
		basic_auth_pass	string	0	Basic認証用のパスワード	20文字
		header_fields	(配列)	0	HTTPヘッダ (配列の要素は最大10個)	
		field_name	string	M	ヘッダフィールド名。":" は含みません。	20文字
		field_value	string	M	上記ヘッダフィールドに 格納する値	512文字
		body	string	0	ボディに格納する値 省略時はイベントのトリ ガとなったリソースデー タ本体・イベントID・イベ ント発生日時・対象リソー スパス	1024文字

	smtp		0	SMTP 通知設定	
	send_to	string	M	通知先 E メールアドレス	256 文字
	subject	string	0	件名	256 文字
	body	string	M	本文	140 文字

(*1) M: 必須、0: オプション。各子要素の M/O は、親要素を設定した場合の要否を表します。

(*2) start_time と end_time の片方のみを設定することはできません。また、両設定値は日時または、時刻のどちらかに揃える必要があります。

(*3) 日時指定の場合は、ISO8601 に従ってください (20141225T103612Z など)。また、精度は秒まで指定可能です。時刻指定の場合は ISO8601 から年月日の情報を削ったものとなります (T103612Z など)。精度は同様に秒までです。タイムゾーン指定は「±hhmm」形式で省略時は「Z」を添えてください。

(*4) start_time と end_time の片方のみを指定することはできません。また、両設定値は日時または、時刻のどちらかに揃える必要があります。

(*5) value が数値の場合は、eq, ne, gt, ge, lt, le が指定できます。value が文字列の場合は eq, ne, substring_of が指定できます。文字列の比較においては、大文字小文字の区別を行います。

(*6) notification 配下には、http か smtp いずれかの設定が必要です。

- Response
 - 正常応答時

Parameters	値
Status-Code	201
Headers	Location
Body	なし

1. Location には作成したイベントの URI を格納します。

- エラー応答時については 2.3.2 節を参照ください。

9.2. イベント情報の参照

- Request

Parameters	値
Method	GET
URI	Event ID 指定による参照 QUERY による検索
	データリストの取得 HIT 数の取得
Headers	2.3.1 節 参照
Body	なし

- URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M
Tenant ID	対象テナントの識別子	M
Event ID	イベントの ID ・省略時 (「/events」) は対象テナントの全てのイベントを対象とします。 (「/events/」の書き方は許容しません。)	0
QUERY	検索条件 (後述)。省略時 (?含む) は、条件なし (全て) として扱う。	0

(*1) M: 必須、0: オプション

- QUERY 一覧

書式	説明
\$filter=<filter 条件>	<filter 条件> に一致するもののみを返すよう、結果を限定します。<filter 条件> は、「プロパティ名 演算子 条件値」とし、and で複数個を定義する

	ことができます。使用可能な演算子、プロパティ名は後述します。
\$top=n	検索結果として得られるデータを n 件に限定します。最大取得件数は 1000 件です。
\$skip=n	検索結果として得られたデータを n 件 skip します。検索結果として得られるデータの順番は EventID でソートされます。

1. /_count が指定された場合\$top、\$skip は無視されます (\$filter のみ併用できます)

- ・ filter 条件の演算子

演算子	説明	例
eq	等号	\$filter=_resource_path eq 'hoge'

- ・ filter 条件に利用可能なプロパティ名

プロパティ名	説明	備考
_resource_path	リソースパス	「/」も含めてパーセントエンコード不要

- ・ filter 条件のサポート関数

Function	説明	例
bool startswith(string p0, string p1)	前方一致	\$filter=startswith(_resource_path, 'hoge') eq true

1. false 指定は未対応です。
2. \$filter 条件直下で_resource_path eq hoge と指定されたときは完全指定となり、startswith()を利用して指定された際は前方一致指定となります。

- ・ Response

[Event ID 指定での参照 または データリストの取得]

- ・ 正常応答時

Parameters		値
Status-Code		200
Headers	Content-Type	application/json; charset=UTF-8
Body		一致したイベント情報

1. Body

- 次の通り、各イベントの情報を JSON オブジェクトとし、JSON 配列形式で複数格納します。(太字: イベント登録時との差分)
- 各イベント情報はイベント ID(event_id)で昇順にてソートされます。

Parameters	形式	M/O(*1)	説明	最大長
events		M	開始タグ	
event_id	string	M	イベント ID	12 文字
conditions		M	イベント条件	
targets		M	対象	
resource_path	string	M	対象のリソースパス	128 文字
operations	string (配列)	M	対象のリソースデータ操作 ・ JSON リソースの場合、 「 [“ create ” , “ update ”] 」 を指定してください。 ・ Binary リソースの場合、 「 “ create ” 」 を指定してください。	33 文字
read_access_code	string	M	対象リソースパスの read 権	48 文字

					または hierarchy_get 権を持つアクセスコード	
		notification_condition		0	通知条件	
		start_time	日時または時刻	0	開始日時または、時刻	20 文字
		end_time	日時または時刻	0	終了日時または、時刻	20 文字
		body_conditions		0	データボディ条件	
		path_type	string	M	path の形式。 “JSONPath” を指定してください。	8 文字
		path	string	M	ボディ要素を指定するためのパス	1902 文字
		comparing_operator	string	M	比較演算子。 “eq” (=), “ne” (), “gt” (>), “ge” (), “lt” (<), “le” (), “substring_of” (部分一致)のいずれか。	12 文字
		value	string または 数値	M	比較対象の値。	文字列の場合、 1 ~ 128 文字 整数の場合、 -2, 147, 483, 648 ~ 2, 147, 483, 647 実数の場合、 整数部分 10 桁、 小数部分 5 桁
		awake_condition		0	イベント抑止解除条件 本設定を行うと、 notification_condition を 満たしイベント通知実施 後、イベント抑止状態 になります。イベント抑止状態 を解除するための条件をこ こで設定します。	
		body_conditions		0	データボディ条件	
		path_type	string	M	path の形式。 “JSONPath” を指定してく ださい。	8 文字
		path	string	M	ボディ要素を指定するためのパス	1902 文字
		comparing_operator	string	M	比較演算子。 “eq” (=), “ne” (), “gt” (>), “ge” (), “lt” (<), “le” (), “substring_of” (部分一致)のいずれか。	12 文字
		value	string	M	比較対象の値。	文字列の場合、

				または 数値			1 ~ 128 文字 整数の場合、 -2, 147, 483, 648 ~ 2, 147, 483, 647 実数の場合、 整数部分 10 桁、 小数部分 5 桁
				notification		M	通知内容
				http		0	HTTP 通知設定
				method	string	M	“ GET ”, “ POST ”, “ PUT ”, “ DELETE ”, “ HEAD ”, “ OPTIONS ”, “ TRACE ” のいずれか。 7 文字
				uri	string	M	URI。 “ http:// ~ ” または “ https:// ~ ” 256 文字
				basic_auth_id	string	0	Basic 認証用の ID 20 文字
				basic_auth_pass	string	0	Basic 認証用のパスワード 20 文字
				header_fields	(配列)	0	HTTP ヘッダ
				field_name	string	M	ヘッダフィールド名。“ : ” は含みません。 20 文字
				field_value	string	M	上記ヘッダフィールドに格納する値 512 文字
				body	string	0	ボディに格納する値。 省略時はイベントのトリガ となったリソースデータ本 体・イベント ID・イベント 発生日時・対象リソースパ ス。 1024 文字
				smtp		0	SMTP 通知設定
				send_to	string	M	通知先 E メールアドレス 256 文字
				subject	string	0	件名 256 文字
				body	string	M	本文 140 文字

(*1) M: 必須、0: オプション。各子要素の M/O は、親要素を設定した場合の有無を表します。

- パラメータに設定値がない場合、レスポンスにはパラメータそのものを格納しません。
- リクエストの Authorization ヘッダに格納したアクセスコードが、list 権を持たないリソースに対するイベント情報は、レスポンスに格納しません。

・ エラー応答時

1. 条件に一致するイベント情報が存在しなかった場合、404 を返します。
2. 条件に一致したイベント数が 1000 を超える場合、400 を返します。
 - この時、エラーレスポンスに受付可能な top 数として、acceptable_top=n を Body に格納します。
3. その他エラーについては、2.3.2 節を参照ください。

[HIT 数の取得]

- ・ 正常応答時

Parameters	値
Status-Code	200
Headers	Content-Type text/plain
Body	一致したリソースデータ数

1. Body

- 一致したイベント数をそのまま返却します。
- リクエストの Authorization ヘッダに格納したアクセスコードが、list 権を持たないリソースに対するイベントは、カウント対象としません。

- ・ エラー応答時については、2.3.2 節を参照ください。

9.3. イベント情報の更新

- ・ Request

Parameters	値
Method	PUT
URI	<Base URL>/v1/<Tenant ID>/_events/<Event ID>
Headers	2.3.1 節 および下記参照
Body	イベント情報

- ・ URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M
Tenant ID	対象テナントの識別子	M
Event ID	イベント ID	M

(*1) M: 必須、O: オプション

- ・ HTTP ヘッダ

Headers	値	補足	M/O(*1)
Content-Type	application/json; charset=UTF-8	固定	M

(*1) M: 必須、O: オプション

- ・ Body : イベント情報

- 9.1 節を参照。
- 対象のイベント情報を、更新データで全て上書きします。
 - 更新前に設定パラメータとして A,B が登録されており、更新で A だけを指定した場合、B は削除指示と判断して B を削除します。

- ・ Response

- ・ 正常応答時

Parameters	値
Status-Code	200
Headers	2.3.2 節 参照
Body	なし

- ・ エラー応答時には 2.3.2 節を参照ください。

9.4. イベントの削除

- ・ Request

Parameters	値
Method	DELETE
URI	<Base URL>/v1/<Tenant ID>/_events/<Event ID>
Headers	2.3.1 節 参照
Body	なし

- ・ URI パラメータ

パラメータ名	説明	M/O(*1)
Base URL	2.3.1 節 参照	M
Tenant ID	対象テナントの識別子	M
Event ID	イベント ID	M

(*1) M：必須、O：オプション

- ・ Response
 - ・ 正常応答時

Parameters	値
Status-Code	204
Headers	2.3.2 節 参照
Body	なし

- ・ エラー応答時には 2.3.2 節を参照ください。

第10章 外部システム呼出しリファレンス

本サービスの以下の機能では、外部システムの呼出しを行います。本章では、呼出しにおける動作仕様について説明します。

- ・ API 呼出し (イベント機能)
- ・ Email 送信 (イベント機能)
- ・ API 呼出し (リソース(転送)_JSON 機能)

10.1. API 呼び出し (イベント機能)

イベントにおけるアクション内容として「API 呼び出し」を選択した際に呼び出される API の内容を以下に示します。

- ・ Request

Parameters	値
Method	< イベントにおいて指定されたメソッド >
URI	< イベントにおいて指定された URI >
Headers	Host: < イベントにおいて指定された URI に含まれるホスト名 >
	< イベントにおいて指定されたヘッダ名 > : < イベントにおいて指定されたヘッダ値 > (複数)
	認証 ID、認証パスワードをいれた場合 : Authorization: Basic < 認証 ID: 認証パスワードを Base64 エンコードした値 >
Body	ボディに値を設定した場合 : ボディに設定した値
	ボディに値を設定しなかった場合 : JSON 形式で表 13 のデータを格納

(注) 基本的には HTTP/1.1 仕様に沿った要求を発行しますが、お客様の設定内容やシステムの実装により HTTP 仕様を逸脱する場合があります。問題が発生した場合は弊社までご相談ください。

表 13 : イベント通知データとして格納される JSON データ

Name	value
event_id	イベント ID
date	イベント判定を行った日時 (*1)
resource_path	イベントに使われたリソースパス
operation	リソースパスに対して行った制御内容 create は登録、update は更新を意味する
body	リソースパスに対して登録または更新された際の Body 文

(*1) ISO8601(基本表記としてのミリ秒表現を使用)に従います (20141225T103612.001Z など)。精度はミリ秒(ミリ秒を省略した場合、0 ミリ秒とみなします)です。秒とミリ秒の区切りは「.」、タイムゾーン指定は「±hhmm」形式で省略時は「Z」を添えます。本サービスがレスポンスに格納する場合、UTC を用います。

イベント ID が「123456789abc」、イベント判定を行った日時が日本時間 2016 年 3 月 1 日 0 時 0 分 0 秒、リソースパスには「Printer/1F/Printer」が指定されており、Body 文に「{"temperature": 24}」を登録した場合、イベントの送信される Body 文の例を以下に示します。

```
{
  "event_id" : "123456789abc",
  "date" : "20160301T000000.000Z",
  "resource_path" : "Printer/1F/Printer",
  "operation" : "create",
  "body" : {
```

```

"temperature" : 24
}
}

```

・ Response

HTTP メッセージフォーマット形式を満たしてください。ステータスラインのみを必要とします。以下に、ステータスコードに対する処理仕様を示します。

ステータスコード	判断	リトライ
2xx	受付成功	しない
408, 503, 504, 509 応答未受信	受付失敗	する
上記以外	受付失敗	しない

10.2. Email 送信 (イベント機能)

イベントにおいてアクション内容として「Email」を選択した際に送信されるメールの内容を以下に示します。

Parameters	値	
To	イベントにおいて指定されたメールアドレス	
From	eventinfo@<zone>.fujitsu.com(*1)	
Return-Path	iot-system-info@ml.css.fujitsu.com	
Subject	イベントにおいて指定された件名	
Body	Content-Type	text/plain; charset=UTF-8
	本体	イベントにおいて指定された本文
添付	Content-Type	text/plain; charset=UTF-8; name=eventinfo.txt
	本体	JSON 形式で 表 13 のデータを格納

(*1) ドメイン名の<zone>はサービス提供システムに依存します。ご契約後の通知内容に従ってください。

10.3. API 呼出し (リソース(転送)_JSON 機能)

リソース(転送)_JSON を利用した際に呼び出される API の内容を以下に示します。

・ Request

Parameters	値
Method	<リソース(転送)_JSON において指定されたメソッド>
URI	<リソース(転送)_JSON において指定された URI >
Headers	Host: <リソース(転送)_JSON において指定された URI に含まれるホスト名>
	<リソース(転送)_JSON において指定されたヘッダ名> : <リソース(転送)_JSON において指定されたヘッダ値> (複数)
	認証 ID、認証パスワードをいれた場合 : Authorization: Basic <認証 ID:認証パスワードを Base64 エンコードした値>
Body	リソース(転送)_JSON に格納したデータ

(注) 基本的には HTTP/1.1 仕様に沿った要求を発行しますが、お客様の設定内容やシステムの実装により HTTP 仕様を逸脱する場合があります。問題が発生した場合は弊社までご相談ください。

・ Response

HTTP メッセージフォーマット形式を満たしてください。ステータスラインのみを必要とします。以下に、ステータスコードに対する処理仕様を示します。

ステータスコード	判断	リトライ
----------	----	------

2xx	受付成功	しない
408, 503, 504, 509 応答未受信	受付失敗	する
上記以外	受付失敗	しない

第11章 システム格納データリファレンス

本サービスでは、システムが自律的にリソースに格納するデータがあります。本章では、格納されるデータの形式や内容について説明します。

11.1. リコメンドリソース

ダイナミックリソースコントローラーにより、ゲートウェイ装置が参照するリコメンド値を JSON 形式で格納します。

リコメンドリソース名	サービスポータルから設定
------------	--------------

Parameters	形式	M/O(*1)	説明
recommend_value	string	M	リコメンド値 ("ON"または"OFF")
recommend_parameter	string	M	リコメンド引数 (サービスポータルにおける分散設定において設定した引数。 但し、本サービスのシステム負荷が極めて大きい場合、"ALM"が入ります)

エッジコンピューティング条件の引数が「recommend is on」である場合において、リコメンド値が「ON」に変化した際にリコメンドリソースに格納される JSON データ例を以下に示します。

```
{
  "recommend_value" : "ON "
  "recommend_parameter" : "recommend is on"
}
```

11.2. 負荷リソース

ダイナミックリソースコントローラーが、API アクセスの負荷計測結果を JSON 形式で格納します。

負荷リソース名	サービスポータルから設定
---------	--------------

Parameters	形式	M/O(*1)	説明
tps	string	M	お客様テナントに対する負荷データ (tps)[transaction/second]
bps	string	M	お客様テナントに対する負荷データ (bps)[bit/second]

(*1) M : 必須、O : オプション

10tps、10,000bps の場合において、負荷リソースに格納される JSON データの例を以下に記載します。

```
{
  "tps" : "10",
  "bps" : "10000"
}
```

11.3. エラー収集用リソース

エラーログ蓄積機能が、本サービス内および通信先の影響でリクエストが欠損した場合のエラー情報を格納します。該当時刻近辺の処理結果を確認し対応ください。

エラー収集用リソース名		_error		
Parameters	形式	M/O(*1)	説明	
collection_first_time	日時(*2)	M	エラー収集期間の開始時刻	
collection_last_time	日時	M	エラー収集期間の終了時刻	
errors	(配列)	M	エラー情報	
trigger	string	M	失敗要因が設定されます。	
resource_path	string	M	エラーが発生したリソースパス。エラー収集期間内の最も古い情報のリソースパスが設定されます。	
data_registration_time	日時	0	リソースデータ登録時刻。失敗要因がイベントの場合のみ設定されます。	
event_id	string	0	イベント ID。失敗要因がイベントの場合のみ設定されます。	
error_factor_message	string	M	失敗要因メッセージ。下記参照ください。	
number_of_times	数値	M	エラー収集期間内で失敗要因と失敗要因メッセージが同一であったものの回数。	
request_id	string の配列	0	エラー発生となった要求にて指定した x-iotpf-request-id の値が設定されます。	
message	string	0	エラー情報登録失敗の場合に下記が設定されます。 ・エラー発生回数過多により request_id 内容が未記録: 'request_id' was not able to be registered because of so many numbers.	

(*1) M: 必須、0: オプション

(*2) ISO8601(基本表記としてのミリ秒表現を使用)に従います(20141225T103612.001Z など)。精度はミリ秒(ミリ秒を省略した場合、0ミリ秒とみなします)です。秒とミリ秒の区切りは「.」、タイムゾーン指定は「±hhmm」形式で省略時は「Z」を添えます。本サービスがレスポンスに格納する場合、UTC を用います。

error_factor_message に記載される失敗要因メッセージの一覧を以下に記載します。

trigger	失敗要因メッセージ	説明
Transfer Resource	It failed in the transmission of forwarding data. (IoT-PF internal error)	転送失敗: システム内要因により失敗しました。
Transfer Resource	It failed in the transmission of forwarding data. (Forwarding parameter error)	転送失敗: 転送設定情報の不備により失敗しました。
Transfer Resource	It failed in the transmission of forwarding data. (Connection error(HTTP))	転送失敗: 通信エラー(HTTP)により失敗しました。
Transfer Resource	It failed in the transmission of forwarding data. (Connection error(TCP))	転送失敗: 通信エラー(TCP)により失敗しました。
Event	Event connection failed.	イベント送信失敗: イベント送信先への接続に失敗しました。
Event	Event transmission retry over. code=408	イベント送信失敗: イベント送信が、レスポンスコード 408 により再送しましたが、再送

		回数オーバーにより失敗しました。
Event	Event transmission failed. code=404	イベント送信失敗：イベント送信が、レスポンスコード 404 で失敗しました。
Event	Event transmission retry over. code=503	イベント送信失敗：イベント送信が、レスポンスコード 503 により再送しましたが、再送回数オーバーにより失敗しました。
Event	Event transmission failed. code=500	イベント送信失敗：イベント送信が、レスポンスコード 500 により失敗しました。
Event	Event Send Request was rejected due to inflow regulation.	イベント送信失敗：要求過負荷によりイベント送信に失敗しました。
Event	Fail to send SMTP mail.	イベント送信失敗：システム内要因により、イベント送信(メール送信)に失敗しました。
Event	Retry out to send SMTP mail.	イベント送信失敗：メールサーバとの通信異常が発生し、イベント送信(メール送信)に失敗しました。
MQTT Broker	Failed to execute publishing resource data, in the system internal processing.	MQTT データ格納失敗：内部処理においてデータ喪失が発生し、リソースデータの格納に失敗しました。
MQTT Parser	Failed to execute publishing resource data, in the system internal processing.	MQTT データ格納失敗：内部処理においてデータ喪失が発生し、リソースデータの格納に失敗しました。
MQTT Parser	Failed to execute publishing resource data, in invalid data format.	MQTT データ格納失敗：登録データ(ペイロードJSON/登録日時/リクエスト識別ID等)が仕様に準じていない形式のため、データの格納に失敗しました。
MQTT Parser	Failed to execute publishing resource data, in message payload length error.	MQTT データ格納失敗：登録データ(ペイロード長)が不正のため、データの格納に失敗しました。

付録1 Response エラー時のメッセージ一覧

ここでは、API 要求がエラーとなった場合に Body 文に通知される内容を説明します。エラーは以下の形式で通知されます。

フィールド名	値
errors.message	エラー文
errors.acceptable_top	レスポンスの Body サイズが 16MB 以内となる top 数 (エラーコードが"number of response-data is larger than 1000"または,"response size is larger than 16MB"の場合のみ)

Body 文の例を以下に記載します。

```
{
  "errors": [
    {
      "message": "response size is larger than 16MB",
      "acceptable_top": 100
    }
  ]
}
```

errors.message に記載されるエラーの Body 文内容の一覧を以下に記載します。

HTTP レスポンス	エラーの Body 文内容	対処方針
503 Service Unavailable	{"errors":[{"message":"[CREATE] iot-pf is temporarily unavailable.[11007]}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[CREATE] iot-pf is temporarily unavailable.[11009]}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[CREATE] iot-pf is temporarily unavailable.[11010]}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[CREATE] iot-pf internal error.[11011]}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[CREATE] iot-pf internal error.[11013]}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクへまでお願い致します。
500 Internal Server Error	{"errors":[{"message":"[CREATE] iot-pf internal error.[11016]}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[CREATE] iot-pf internal error.[11017]}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[CREATE] iot-pf is temporarily unavailable.[11022]}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。

503 Service Unavailable	{"errors":[{"message":"[CREATE] No Connection.[11027]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[SEARCH] iot-pf is temporarily unavailable.[11028]"}]}	システム高負荷の場合に発生することがあります。検索範囲を絞って再実施することを推奨します。
503 Service Unavailable	{"errors":[{"message":"[SEARCH] iot-pf is temporarily unavailable.[11030]"}]}	システム高負荷の場合に発生することがあります。検索範囲を絞って再実施することを推奨します。
503 Service Unavailable	{"errors":[{"message":"[SEARCH] iot-pf is temporarily unavailable.[11031]"}]}	システム高負荷の場合に発生する可能性があります。検索範囲を絞って再実施することを推奨します。
500 Internal Server Error	{"errors":[{"message":"[SEARCH] iot-pf internal error.[11032]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[SEARCH] iot-pf internal error.[11034]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[SEARCH] iot-pf internal error.[11036]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[SEARCH] iot-pf internal error.[11037]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[SEARCH] iot-pf is temporarily unavailable.[11039]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[SEARCH] No Connection.[11040]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[UPDATE] iot-pf is temporarily unavailable.[11041]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[UPDATE] iot-pf is temporarily unavailable.[11043]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[UPDATE] iot-pf is temporarily unavailable.[11044]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[UPDATE] iot-pf internal error.[11045]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[UPDATE] iot-pf internal error.[11047]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[UPDATE] iot-pf internal error.[11049]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[UPDATE] iot-pf internal error.[11050]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスク

		までご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[UPDATE] iot-pf is temporally unavailable.[11051]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[UPDATE] No Connection.[11052]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[REMOVE] iot-pf is temporally unavailable.[11053]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[REMOVE] iot-pf is temporally unavailable.[11055]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[REMOVE] iot-pf is temporally unavailable.[11056]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[REMOVE] iot-pf internal error.[11057]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[REMOVE] iot-pf internal error.[11059]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[REMOVE] iot-pf internal error.[11061]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[REMOVE] iot-pf internal error.[11062]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[REMOVE] iot-pf is temporally unavailable.[11063]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[REMOVE] No Connection.[11064]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[CREATE] iot-pf is temporally unavailable.[12003]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
400 Bad Request	{"errors":[{"message":"[CREATE] main data is required."}]}	Body文に記載がありません。Body文が正しく記載されているか確認してください。
400 Bad Request	{"errors":[{"message":"[CREATE] main data is too large."}]}	Body文が大きすぎます。データを分割するなどBody文を小さくしてください。
503 Service Unavailable	{"errors":[{"message":"[CREATE] iot-pf is temporally unavailable.[12006]"}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
400 Bad Request	{"errors":[{"message":"[CREATE] url format error."}]}	URLの指定が誤っています。URLを見なおしてください。
400 Bad Request	{"errors":[{"message":"[CREATE] query num invalid."}]}	URLに"?"が複数存在します。クエリを見なおしてください。
400 Bad Request	{"errors":[{"message":"[CREATE] query"}]}	クエリが長すぎます。クエリを見な

	<code>too large."}]}</code>	おしてください。
400 Bad Request	<code>{"errors":[{"message":"[CREATE] url unescape error."}]}</code>	URLの指定が誤っています。URLを見なおしてください。
400 Bad Request	<code>{"errors":[{"message":"[CREATE] access code is wrong."}]}</code>	アクセスコードが誤っています。正しいアクセスコードが記載されているか確認してください。
400 Bad Request	<code>{"errors":[{"message":"[CREATE] x-iotpf-request-id format error."}]}</code>	x-iotpf-request-idの内容が不正です。
400 Bad Request	<code>{"errors":[{"message":"[SEARCH] url format error."}]}</code>	URLの指定が誤っています。URLを見なおしてください。
400 Bad Request	<code>{"errors":[{"message":"[SEARCH] query num invalid."}]}</code>	URLに"?"が複数存在します。クエリを見なおしてください。
400 Bad Request	<code>{"errors":[{"message":"[SEARCH] query too large."}]}</code>	クエリが長すぎます。クエリを見なおしてください。
400 Bad Request	<code>{"errors":[{"message":"[SEARCH] url unescape error."}]}</code>	URLの指定が誤っています。URLを見なおしてください。
400 Bad Request	<code>{"errors":[{"message":"[SEARCH] access code is wrong."}]}</code>	アクセスコードが誤っています。正しいアクセスコードが記載されているか確認してください。
503 Service Unavailable	<code>{"errors":[{"message":"iot-pf is temporarily unavailable.[12026]"}]}</code>	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
400 Bad Request	<code>{"errors":[{"message":"[UPDATE] main data is required."}]}</code>	Body文に記載がありません。Body文が正しく記載されているか確認してください。
400 Bad Request	<code>{"errors":[{"message":"[UPDATE] main data is too large."}]}</code>	Body文が大きすぎます。データを分割するなどBody文を小さくしてください。
503 Service Unavailable	<code>{"errors":[{"message":"iot-pf is temporarily unavailable.[12029]"}]}</code>	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
400 Bad Request	<code>{"errors":[{"message":"[UPDATE] url format error."}]}</code>	URLの指定が誤っています。URLを見なおしてください。
400 Bad Request	<code>{"errors":[{"message":"[UPDATE] query num invalid."}]}</code>	URLに"?"が複数存在します。クエリを見なおしてください。
400 Bad Request	<code>{"errors":[{"message":"[UPDATE] query too large."}]}</code>	クエリが長すぎます。クエリを見なおしてください。
400 Bad Request	<code>{"errors":[{"message":"[UPDATE] url unescape error."}]}</code>	URLの指定が誤っています。URLを見なおしてください。
400 Bad Request	<code>{"errors":[{"message":"[UPDATE] access code is wrong."}]}</code>	アクセスコードが誤っています。正しいアクセスコードが記載されているか確認してください。
400 Bad Request	<code>{"errors":[{"message":"[UPDATE] x-iotpf-request-id format error."}]}</code>	x-iotpf-request-idの内容が不正です。
400 Bad Request	<code>{"errors":[{"message":"[REMOVE] url format error."}]}</code>	URLの指定が誤っています。URLを見なおしてください。
400 Bad Request	<code>{"errors":[{"message":"[REMOVE] query num invalid."}]}</code>	URLに"?"が複数存在します。クエリを見なおしてください。
400 Bad Request	<code>{"errors":[{"message":"[REMOVE] query too large."}]}</code>	クエリが長すぎます。クエリを見なおしてください。
400 Bad Request	<code>{"errors":[{"message":"[REMOVE] url unescape error."}]}</code>	URLの指定が誤っています。URLを見なおしてください。

400 Bad Request	{"errors":[{"message":"[REMOVE] query must not be exists. for present"}]}	最新削除(_present)にも関わらず、クエリが指定されています。クエリを除いて実行してください。
400 Bad Request	{"errors":[{"message":"[REMOVE] query is required. for past."}]}	過去のデータの削除(_past)にも関わらず、クエリが指定されていません。クエリを指定してください。
400 Bad Request	{"errors":[{"message":"[REMOVE] access code is wrong."}]}	アクセスコードが誤っています。正しいアクセスコードが記載されているか確認してください。
400 Bad Request	{"errors":[{"message":"[REMOVE] x-iotpf-request-id format error."}]}	x-iotpf-request-idの内容が不正です。
429 Too Many Requests	{"errors":[{"message":"Number of request per second has exceeded maximum usage limit of service contract."}]}	アクセス頻度がシステム許容値を超えています。アクセス頻度を見直してください。
400 Bad Request	{"errors":[{"message":"fail to get decompressed data size."}]}	指定された圧縮形式の情報が異常です。
400 Bad Request	{"errors":[{"message":" decompressed data is too large."}]}	伸長後のデータ長が大きすぎます。
423 Locked	{"errors":[{"message":" request access code has event. "}]}	対象アクセスコードに紐づいたイベントが存在しています。イベントの紐付けを解除してください。
423 Locked	{"errors":[{"message":" request access code can't read event."}]}	対象アクセスコードに紐付いたイベントが存在しているため、read権を削除できません。
423 Locked	{"errors":[{"message":" request access code has dispersion policy."}]}	対象アクセスコードに紐付いた分散ポリシーが存在しています。分散ポリシーの紐付けを解除してください。
423 Locked	{"errors":[{"message":" request access code has recommend resource."}]}	対象アクセスコードに紐付いたリコメンドリソースが存在しています。リコメンドリソースの紐付けを解除してください。
423 Locked	{"errors":[{"message":" request access code can't update dispersion policy. "}]}	対象アクセスコードに紐付いた分散ポリシーが存在しているため、update件を削除できません。
423 Locked	{"errors":[{"message":" request access code can't update recommend resource."}]}	対象アクセスコードに紐付いたリコメンドリソースが存在しているため、update件を削除できません。
423 Locked	{"errors":[{"message":" request access code and resource have event."}]}	対象アクセスコードとリソースに紐付いたイベントが存在しています。イベントの紐付けを解除してください。
503 Service Unavailable	{"errors":[{"message":" iot-pf Service Unavailable."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":" iot-pf internal error."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。

404 Not Found	{"errors":[{"message":" Reload configuration to fail to read config."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
400 Bad Request	{"errors":[{"message":" tenant id is required. "}]}	テナント ID が指定されていません。
400 Bad Request	{"errors":[{"message":" access code is required. "}]}	アクセスコードが指定されていません。
400 Bad Request	{"errors":[{"message":" input parameter error. : resource_path is duplicated. (resourcepath=[リソースパス])"}]}	リソースパスの重複があります。
400 Bad Request	{"errors":[{"message":" parameter duplication. "}]}	パラメータに重複があります。
400 Bad Request	{"errors":[{"message":" request access code already exists. "}]}	指定されたアクセスコードがすでに存在しています。
400 Bad Request	{"errors":[{"message":" request resource path does not exist. : ResourcePath= [リソースパス]"}]}	指定されたリソースパスは存在しません。
404 Not Found	{"errors":[{"message":" resource path not found. "}]}	対象のリソース ID が登録されていません。
404 Not Found	{"errors":[{"message":" access code not found. "}]}	検索条件に一致するアクセスコードが見つかりませんでした。
404 Not Found	{"errors":[{"message":" event not found. "}]}	検索条件に一致するイベントが見つかりませんでした。
404 Not Found	{"errors":[{"message":" target resource not found. "}]}	対象リソースが見つかりません。
404 Not Found	{"errors":[{"message":" tenant ID not found. "}]}	該当するテナント ID が見つかりません。
400 Bad Request	{"errors":[{"message":" event id is required. "}]}	イベント ID が指定されていません。
400 Bad Request	{"errors":[{"message":" URL FORMAT ERROR"}]}	正しいリクエスト URL ではありません。
400 Bad Request	{"errors":[{"message":" Incorrect access code search conditions. "}]}	アクセスコード検索条件に誤りがあります。
400 Bad Request	{"errors":[{"message":" Incorrect event code search conditions. "}]}	イベント検索条件に誤りがあります。
400 Bad Request	{"errors":[{"message":" Incorrect filter condition. "}]}	filter 条件に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter is error. : incorrect top condition "}]}	top 条件に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter is error. : incorrect skip condition "}]}	skip 条件に誤りがあります。
400 Bad Request	{"errors":[{"message":" Request data format error. "}]}	リクエストデータの内容に誤りがあります。
400 Bad Request	{"errors":[{"message":" Number of resource path is over for one request. (resourcePathSize=[リソースパス回数])"}]}	1 リクエストで登録可能なリソースパスの制限を超えています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : permissions"}]}	アクセスコード権限情報が不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : resource_path and operations in resource_operations"}]}	リソース権限の情報が不足しています。

400 Bad Request	{"errors":[{"message":" input parameter error is required. : resource_path of resource_operations"}]}	リソース権限のリソースパスが不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : operations of resource_operations"}]}	リソース権限のアクセス権が不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : conditions"}]}	イベント条件が不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : targets"}]}	イベント対象が不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : resource_path of targets"}]}	イベント対象のリソースパスが不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : operations of targets"}]}	対象のリソースデータ操作が不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : read_access_code of targets"}]}	対象リソースパスの read 権をもつアクセスコードが不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : path_type of notification_condition"}]}	通知条件の path 形式が不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : path of notification_condition"}]}	通知条件のボディ要素を指定するリソースパスが不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : comparing_operator of notification_condition"}]}	通知条件の比較演算子が不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : value of notification_condition"}]}	通知条件の比較対象の値が不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : path_type of awake_condition"}]}	抑止条件の path 形式が不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : path of awake_condition"}]}	抑止条件のボディ要素を指定するリソースパスが不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : comparing_operator of awake_condition"}]}	抑止条件の比較演算子が不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : value of awake_condition"}]}	抑止条件の比較対象の値が不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : notification smtp or http"}]}	通知内容の指定が不足します。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : send_to of smtp notification"}]}	通知内容の通知先 e メールアドレスが不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : body of smtp notification"}]}	通知内容の通知 e メール本文が不足しています。

400 Bad Request	{"errors":[{"message":" input parameter error is required. : method of http notification"}]}	通知内容のHTTP Methodが不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : uri of http notification"}]}	通知内容のURIが不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : field_name and field_value of http notification"}]}	通知内容のヘッダフィールドが不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : filed_name of http notification"}]}	通知内容のヘッダフィールド名が不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : filed_value of http notification"}]}	通知内容のヘッダフィールドの値が不足しています。
400 Bad Request	{"errors":[{"message":" input parameter error is required. : notification"}]}	通知内容が不足しています。
403 Forbidden	{"errors":[{"message":" Authorization accesscode is required."}]}	アクセスコードが指定されていません。
401 Unauthorized	{"errors":[{"message":" Authorization error. (AccessCode=[アクセスコード], NG_ResoucePath=[リソースパス])"}]}	指定されたアクセスコードで権限のないリソースが含まれています。
401 Unauthorized	{"errors":[{"message":" Client authorization error. (AccessCode=[アクセスコード])"}]}	アクセスコードで認可されていないクライアントからのアクセスです。
401 Unauthorized	{"errors":[{"message":" Protocol authorization error. (AccessCode=[アクセスコード])"}]}	アクセスコードで認可されていないプロトコルでのアクセスです。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect access code operations"}]}	アクセスコードのアクセス権の組合せが不正です。
400 Bad Request	{"errors":[{"message":" request ipfilter does not exist. (tenantId=[テナント ID], startIpAddr=[IP アドレス], endIpAddr=[IP アドレス])"}]}	指定されたアクセス制限設定がありません。
400 Bad Request	{"errors":[{"message":" hit data is larger than limit. (acceptable_top=1000)"}]}	検索されたデータが1000件を超えています。
403 Forbidden	{"errors":[{"message":" Authorization accesscode format error."}]}	アクセスコードの形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" Fail to url decode(utf-8)."}]}	不正な文字コードのデータが含まれています。
400 Bad Request	{"errors":[{"message":" URL format error. : Url Path is required."}]}	URLパスが必要です。
400 Bad Request	{"errors":[{"message":" URL format error. : Incorrect protocol"}]}	不正なプロトコルが指定されています。
400 Bad Request	{"errors":[{"message":" URL format error. : incorrect url path"}]}	URLパスが不正です。
400 Bad Request	{"errors":[{"message":" URL format error. : incorrect url path start"}]}	URLパスが不正な文字で始まっています。
400 Bad Request	{"errors":[{"message":" URL format error. : tenant id format error."}]}	テナントIDの形式が正しくありません。

400 Bad Request	{"errors":[{"message":" URL format error. : input _access_codes or _events."}]}	URL の指定が正しくありません。
400 Bad Request	{"errors":[{"message":" URL format error. : unnecessary query is input"}]}	不要なクエリが指定されています。
400 Bad Request	{"errors":[{"message":" URL format error. : access code format error."}]}	アクセスコードの形式が正しくありません。
400 Bad Request	{"errors":[{"message":" URL format error. : input _counts."}]}	HIT数の取得のURL指定が正しくありません。
400 Bad Request	{"errors":[{"message":" URL format error. : event id format error."}]}	イベントIDの形式が正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect access code name's string length"}]}	アクセスコード名の長さが正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect access code discription's string length"}]}	アクセスコードの説明の長さが正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : number of ipfilter is larger than 5."}]}	5個以上のアクセス制限が指定されています。
400 Bad Request	{"errors":[{"message":" input parameter error. : start ipaddress format error"}]}	アクセス制限のIPアドレスの形式が正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : end ipaddress format error"}]}	アクセス制限のIPアドレスの形式が正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : ipfilter format error."}]}	アクセス制限の指定形式が正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect ipfilter range"}]}	アクセス制限の指定形式が正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : operations format error. (NG Operation kind=[アクセス権])"}]}	アクセス権の指定形式が正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : operation is duplicated."}]}	同じアクセス権が重複して指定されています。
400 Bad Request	{"errors":[{"message":" input parameter error. : resource path format error."}]}	リソースパスの指定形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect event name's string length"}]}	イベント名の文字列長が正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. :event description's string length"}]}	イベントの説明文の文字列長が正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : event target operations format error."}]}	リソース操作の指定形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : inccorect notification_condition's start_time or end_time (start_time=[日時], end_time=[日時])"}]}	イベント判定日時の指定形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : cannot be set to body conditions"}]}	リソース_Binaryに対してイベント条件が指定されています。

	to binary resource."}}}	
400 Bad Request	{"errors":[{"message":" input parameter error. : access code format error."}]}}	アクセスコードの指定形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect path_type of notification condition"}]}}	通知条件の path_type の指定形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect path_type's string length of notification condition"}]}}	通知条件の path_type の文字列長が正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : comparing_operator of notification condition (comparing_operator=[比較条件])"}]}}	通知条件の比較条件の指定形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : comparing operator format error (comparing_operator=[比較演算子])"}]}}	通知条件の比較条件の比較演算子に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect path_type of awake condition"}]}}	抑止条件の指定形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect path's string length of awake condition"}]}}	抑止条件の path の指定文字列長が正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : comparing_operator of awake condition (comparing_operator=[比較条件])"}]}}	抑止条件の比較条件の指定形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect comparing operator of awake condition (comparing_operator=[比較演算子])"}]}}	抑止条件の比較条件の比較演算子に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorret send_to's length of smtp notification"}]}}	通知内容の連絡先 E メールアドレスの形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorret body's length of smtp notification"}]}}	通知内容の E メール本文の形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorret subject's length of smtp notification"}]}}	通知内容の E メール件名の形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect method of http notification"}]}}	通知内容の HTTP メソッドの指定形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect uri's length of http notification"}]}}	通知内容の URL 指定形式の文字列長が正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : number of header fields is larger than 10"}]}}	通知内容のヘッダフィールドが 11 個以上指定されています。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect field_name's length of http notification"}]}}	通知内容のヘッダフィールドの指定形式の文字列長が正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect field_name's length of http notification"}]}}	通知内容のヘッダフィールドの指

	error. : incorrect field_value's length of http notification"]}]}	定形式の文字列長が正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect basic_auth_id's length of http notification"}]}	通知内容の HTTP ベーシック認証 ID の文字列長が正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect basic_auth_pass's length of http notification"}]}	通知内容の HTTP ベーシック認証パスワードの文字列長が正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : certification info format error."}]}	証明書の指定形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : protocols format error."}]}	プロトコルの指定形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : certification file is too large. Size=[サイズ]}]}	証明書のサイズがシステム許容値を越えています。
400 Bad Request	{"errors":[{"message":" input parameter error. : certification format error. ([証明書形式文字列] is required)}]}	証明書の形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : protocols is duplicated."}]}	プロトコルの指定に重複があります。
500 Internal Server Error	{"errors":[{"message":" certification file parse error. ([エラー番号]}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":" [12401] transfer resource is temporarily unavailable"}]}	転送リソースが一時的に利用できません。しばらく待ってから再度ご利用ください。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect body's length of http notification"}]}	通知内容のボディ長が正しくありません。
403 Forbidden	{"errors":[{"message":" Ip filter NG. (tenantId=[テナント ID], accessCode=[アクセスコード], srcIp=[要求元 IP アドレス]}]}	アクセス制限設定により、アクセスを拒否しています。
400 Bad Request	{"errors":[{"message":" request http header error. (Header:[ヘッダ内容]}]}	HTTP ヘッダ内容に誤りがあります。
401 Unauthorized	{"errors":[{"message":" Authorization error. (AccessCode=[アクセスコード]}]}	アクセスコードに誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : uri of http notification format error."}]}	通知内容の HTTP Method の指定形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" input parameter error. : field_name length of http notification format error."}]}	通知内容の HTTP Method の指定文字列長が正しくありません。
400 Bad Request	{"errors":[{"message":" input parameter error. : incorrect body_conditions of awake condition"}]}	抑止条件の body_conditions の指定形式に誤りがあります。
400 Bad Request	{"errors":[{"message":" URL format error. : access code is necessary."}]}	URL にアクセスコードが指定されていません。
400 Bad Request	{"errors":[{"message":" [12990] Illegal input data."}]}	パラメータ指定に誤りがあります。

500 Internal Server Error	{"errors":[{"message":"[11901] Fail to loadCollection."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[12992] Abnormality occurred by PUT processing."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[11903] Fail to create CommonData."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[11904] Fail to get socket(PUT)."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[11905] Fail to send PreProcess(PUT)."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
400 Bad Request	{"errors":[{"message":"[12980] Illegal delete data."}]}	パラメータ指定に誤りがあります。
500 Internal Server Error	{"errors":[{"message":"[11911] Fail to loadCollection."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[12912] Abnormality occurred by DELETE processing."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[11913] Fail to create CommonData."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[11914] Fail to get socket(DELETE)."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[11915] Fail to send PreProcess(DELETE)."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
400 Bad Request	{"errors":[{"message":"[12920] Illegal get data."}]}	パラメータ指定に誤りがあります。
500 Internal Server Error	{"errors":[{"message":"[11921] Fail to loadCollection."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[12922] Abnormality occurred by GET processing."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[11923] Fail to create CommonData."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[11924] Fail to get socket(GET)."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[11925] Fail to send PreProcess(GET)."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[11935] Fail to send PreProcess."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。

408 Request Timeout	{"errors":[{"message":"[11936] REST-Connection Error."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[11937] Service Unavailable.."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
500 Internal Server Error	{"errors":[{"message":"[11938] Internal Server Error."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
503 Service Unavailable	{"errors":[{"message":"[12900] Before start service(for initialization)."}]}	システム初期化処理中です。しばらく待ってから再度ご利用ください。
429 Too Many Requests	{"errors":[{"message":"[12906] Number of request per second has exceeded maximum usage limit of service contract."}]}	アクセス頻度がシステム許容値を越えています。アクセス頻度を見直してください。
503 Service Unavailable	{"errors":[{"message":"[11937] Service Unavailable.."}]}	予期せぬ内部異常を検出しました。エラーメッセージをヘルプデスクまでご連絡願います。
400 Bad Request	{"errors":[{"message":"number of response-data is larger than 1000","acceptable_top": [top 値指定可能数]}]}	取得データ件数が、1,001以上となる(top オプションなし)参照リクエストです。acceptable_top の値に従い top 数を指定し直してください。
400 Bad Request	{"errors":[{"message":"response size is larger than 16MB","acceptable_top": [top 値指定可能数]}]}	取得データ量が、16MBより大となる参照リクエストです。acceptable_top の値に従い top 数を指定し直してください。
503 Service Unavailable	{"errors":[{"message":"number of concurrent request for heavy queries exceeded the maximum usage limit."}]}	処理負荷の高いクエリが多数同時に要求されています。同時要求数を減らしてください。