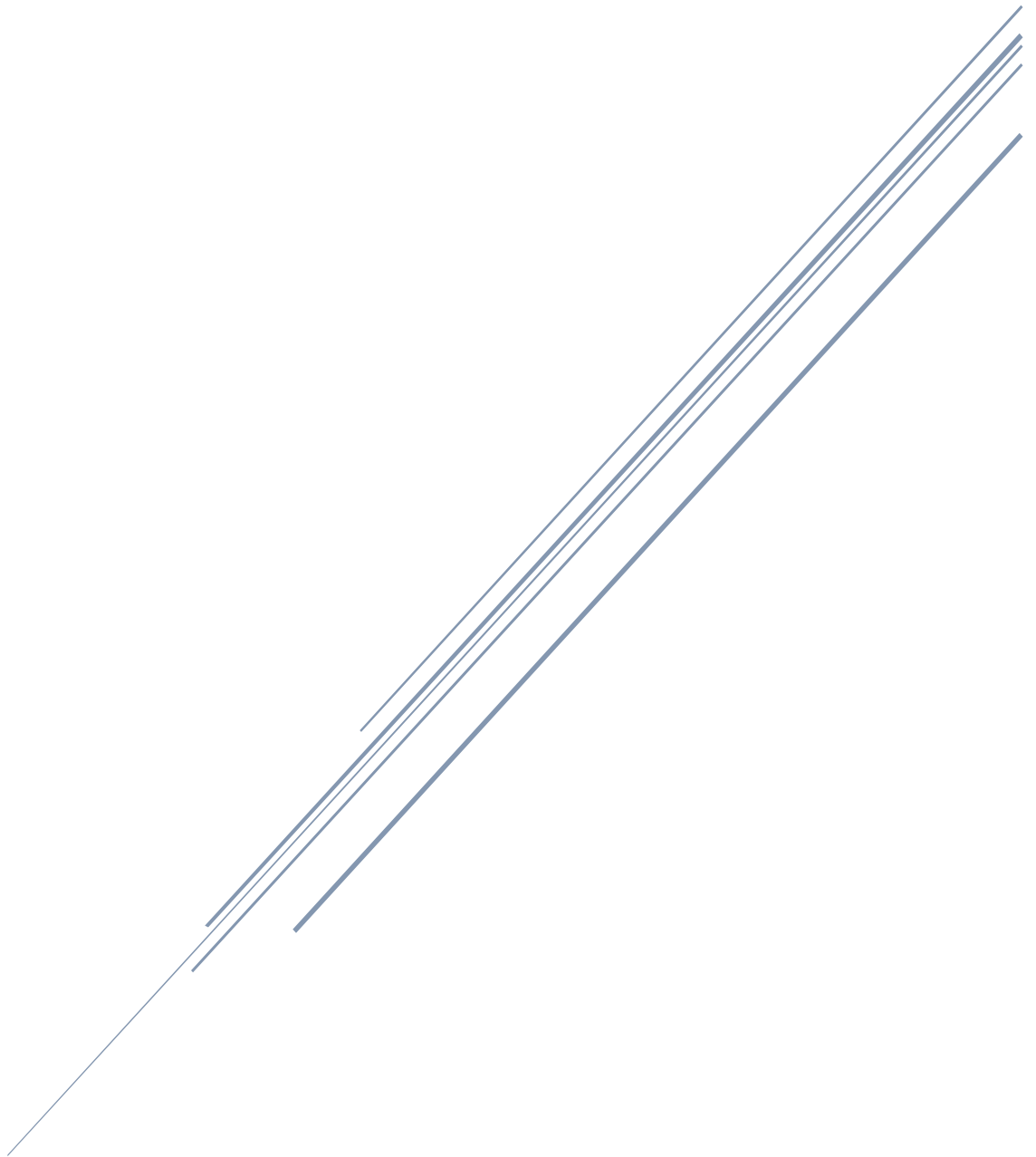


# CIOF システム API リファレン スマニュアル

2025 年 10 月 30 日 Ver2.3

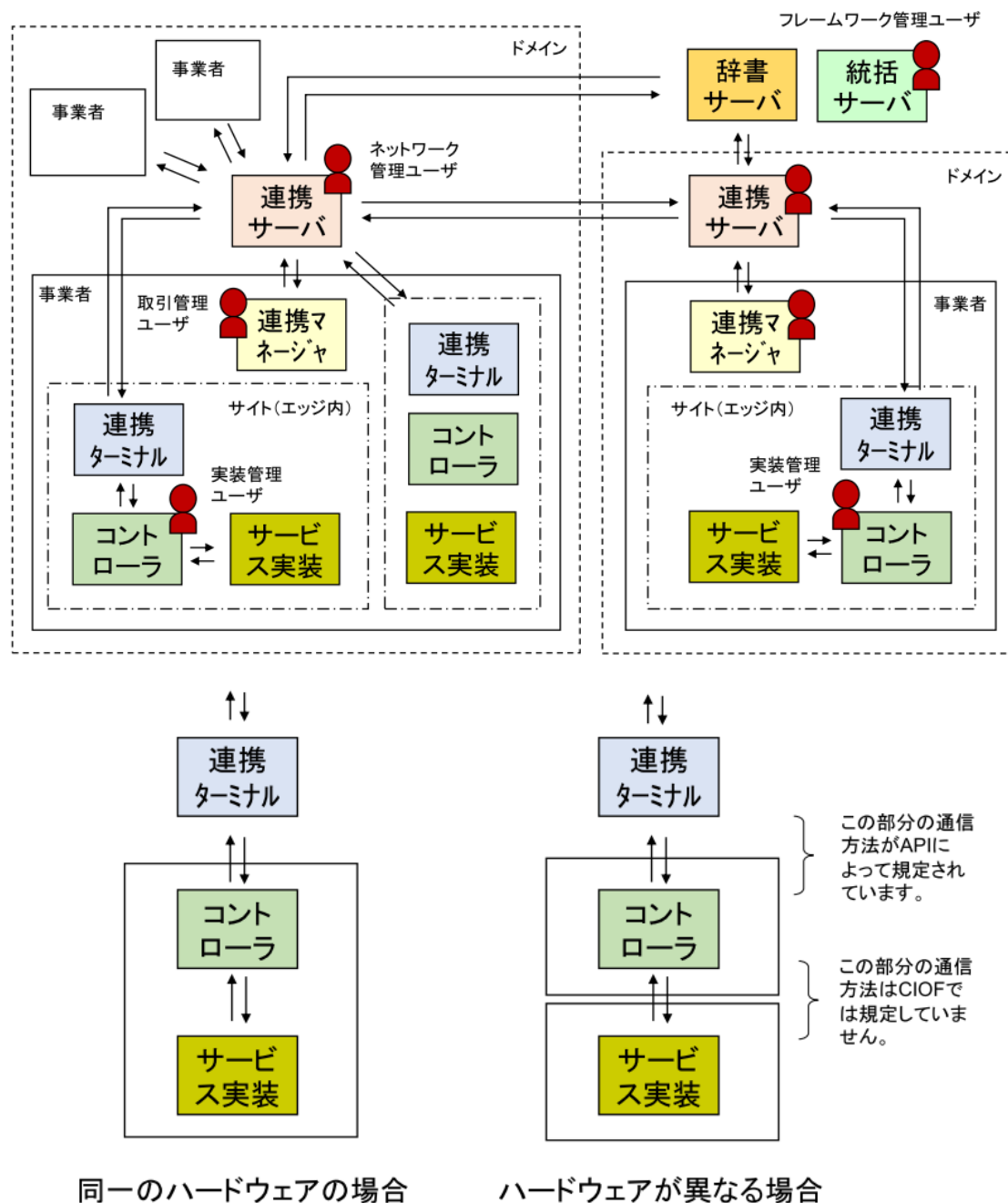


# 目次

1	システム構成.....	2
2	API リファレンスガイド.....	3
2.1	通信プロトコル .....	3
2.2	パラメータの形式.....	3
2.3	HTTP ステータスコード .....	3
2.4	CIOF エラーコード .....	4
2.5	ユーザ認証.....	4
2.6	コントローラ - 連携ターミナル間 API 規定 .....	4
2.7	各 API の仕様 .....	6
2.7.1	コントローラの状態通知.....	6
2.7.2	サービス実装の取得 .....	8
2.7.3	サービス実装の状態通知.....	12
2.7.4	データ実装の取得.....	15
2.7.5	データ実装の状態通知 .....	18
2.7.6	取引契約の取得 .....	21
2.7.7	カレンダーの取得 .....	25
2.7.8	取引データの取得.....	28
2.7.9	取引データの送信.....	31
2.7.10	リクエストパラメータの取得 .....	34
2.7.11	リクエストパラメータの送信 .....	36
2.7.12	データファイルの送信 .....	39
2.7.13	サービス記録の通知 .....	41
2.7.14	サービス記録の取得 .....	46
3	参照規定 .....	48
4	変更履歴 .....	48

# 1 システム構成

CIOF システム全体アーキテクチャおよび今回説明する対象であるコントローラの位置付けおよび API の規定範囲は、次の図で示す通りです。



## 2 API リファレンスガイド

---

### 2.1 通信プロトコル

HTTP(Hypertext Transfer Protocol)/1.1 [2] に準拠します。

### 2.2 パラメータの形式

通信に用いる文字コードは UTF-8(BOM なし)とします。

要求パラメータおよび応答パラメータは、JSON 形式とします。

### 2.3 HTTP ステータスコード

本書で説明する API は、次に示す標準的な HTTP のステータスコードを使用します。ただし、API によっては、本節に記載のないステータスコードを応答する場合があります、その場合は API 毎に別記載しています。

ステータスコード	説明
200	Success リクエストが適切に処理されたことを示します。
400	Bad request リクエストヘッダ、クエリパラメータ、またはリクエストボディが不正であることを示します。該当箇所の確認をお願いします。
401	Unauthorized 認証に失敗したことを示します。ヘッダ内の Bearer token を確認してください。
500	Internal Server Error サーバで内部エラーが発生したことを示します。別途、定められた手続きにより問い合わせください。
502	Bad Gateway 連携ターミナルランチャの設定が誤っており、連携ターミナルとうまく接続ができていないことを示します。連携ターミナルランチャのポート設定が誤っている場合がほとんどですが、ポート設定が正しいにもかかわらず、該当エラーが続く場合は、定められた手続きにより問い合わせください。

503	Service Unavailable 連携ターミナルが起動していないことを示します。しばらく待って再度 API を実行してください。
-----	--

## 2.4 CIOF エラーコード

本書で説明する API は、HTTP ステータスコードの他に独自のエラーコードを応答する予定です。エラーコードの内容については、今後のバージョンアップにて順次対応予定です。

## 2.5 ユーザ認証

コントローラによる連携ターミナルへの接続には、RFC6750[3] の規定に基づき、Bearer 認証のパラメータを HTTP の Authorization ヘッダに含めます。

## 2.6 コントローラ – 連携ターミナル間 API 規定

サイトごとに配置されている連携ターミナルは、コントローラに対して以下の 13 の API を提供します。それぞれのコントローラは、あらかじめ連携マネージャを用いて取得した認証コード<sup>1</sup>を用いてこれらの API にアクセスします。

No	名称	説明	メソッド
1	コントローラの状態通知	コントローラの起動および終了を通知します。	PUT
2	サービス実装の取得	サービス実装、プロセス実装、イベント実装の構造を取得します。	GET
3	サービス実装の状態通知	サービス実装、プロセス実装、イベント実装の状態を通知します。	PUT
4	データ実装の取得	データ実装、データ項目実装の構造を取得します。	GET
5	データ実装の状態通知	データ実装、データ項目実装の状態を通知します。	PUT
6	取引契約の取得	取引契約の内容を取得します。	GET
7	カレンダーの取得	カレンダーの定義を取得します。	GET
8	取引データの取得	取引データを取得します。	GET
9	取引データの送信	契約内容に基づき取引データを送信します。	POST
10	リクエストパラメータの取得	リクエストパラメータを取得します。	GET
11	リクエストパラメータの送信	リクエストパラメータを送信します。	POST
12	データファイルの送信	データファイルを送信します。	POST
13	サービス記録の通知	サービス記録を送信します。	POST

<sup>1</sup> 認証コードの取得方法は、別紙「CIOF システム導入手順書」を参照してください。

14	サービス記録の取得	サービス記録を取得します。	GET
----	-----------	---------------	-----

図 1 コントローラ用 API 一覧

次節以降で、上記 14 の API の仕様について記載します。

## 2.7 各 API の仕様

### 2.7.1 コントローラの状態通知

API 名称

概要

HTTP メソッド

URL パス

要求ヘッダ

要求パラメータ

コントローラの状態通知

本 API は、コントローラの状態を CIOF に伝えるためのものです。コントローラは、起動・終了した際に、その時の状態を CIOF システムに通知する必要があります。次に API の利用パターンを示します。

1.コントローラが電源 ON し、通信が可能となったタイミングで、コントローラ状態通知 API にて、status を ready とすると同時に、polling\_rate を設定します。

2.電源 OFF 時は、コントローラ状態通知 API にて、status を disconnected とします。

これら対応によって、CIOF システムにコントローラの状態が通知されますが、それによって通信に制限がかかることはありません。管理者は連携マネージャから、コントローラの状態を確認することができます。

PUT

/hct/api/v2/controller

Content-Type: application/json  
Authorization: Bearer token=<authorization\_key><sup>2</sup>

メッセージボディ部に、以下のパラメータを与えます。

```
{  
    "status": "ready",  
    "polling_rate": 60  
}
```

属性	型	説明
status	string	(任意)コントローラの状態

<sup>2</sup> <authorization\_key>には、連携マネージャから取得した認証ファイル (EdgeControllerAPIKey.yml)内の authorization key を用います

応答

応答ヘッダ

応答パラメータ

		ス。稼働中”ready”、未接続”disconnected”、停止中”stopped”の区分となります
polling_rate	number	(任意)コントローラから連携ターミナルへのポーリングレートを秒単位で示します。10 秒～1000 秒の範囲とします

2.3 節に準じます。ただし、以下の内容は本 API 独自の内容です。

ステータスコード	内容
422	polling_rate が規定の範囲外である場合に応答されます。

Content-Type: application/json

メッセージボディ部に、以下のパラメータを返します。

```
{
  "status": "ready",
  "polling_rate": 60
}
```

属性	型	説明
status	string	(任意)変更したコントローラのステータス。稼働中”ready”、未接続”disconnected”、停止中”stopped”の区分となります
polling_rate	number	(任意)変更したコントローラから連携ターミナルへのポーリングレートを秒単位で示します。10 秒～1000 秒の範囲とします



## 2.7.2 サービス実装の取得

API 名称	サービス実装の取得
概要	<p>サービス実装、プロセス実装、イベント実装の構造を取得します。</p> <p>ここで取得される情報は、あらかじめ連携マネージャを利用して、CIOF システムに登録されたものです。コントローラは、本 API によって得られた情報が、配下のサービス実装、プロセス実装、イベント実装に関する実際の構成（リアル世界における構成）と同じであることを確認しなければなりません。もし、実際の構成と異なる場合は、連携マネージャを利用して、CIOF システムに登録された内容を修正したうえで、コントローラによって再度修正された構成情報を取得する必要があります。</p>
HTTP メソッド	GET
URL パス	/hct/api/v2/service_implementations
要求ヘッダ	Content-Type: application/json Authorization: Bearer token=<authorization_key> <sup>2</sup>
要求パラメータ	なし
応答	2.3 節に準じます。
応答ヘッダ	Content-Type: application/json
応答パラメータ	<p>メッセージボディ部に、以下のパラメータを返します。</p> <pre>[   {     "id": "10501",     "local_id": "x5VrQsPfiqrzc2J",     "name": "環境情報取得サービス",     "description": "環境データを取得する",     "device_id": [       "device001"     ],   }, ]</pre>

```

“process_implementations”: [
  {
    “id”: “10601”,
    “local_id”: “ShN69VWpC9”,
    “name”: “温湿度の計測”,
    “description”: “温度を計測して、値を応答する”,
    “process_operation_implementations”: [
      {
        “id”: “10701”,
        “index”: 1,
        “data_implementation_id”: “10901”,
        “operation_type”: “create”
      }
    ],
    “event_implementations”: [
      {
        “id”: “10801”,
        “local_id”: “ThSf7UaGDUnd”,
        “event_type”: “trigger”,
        “name”: “計測開始”,
        “description”: “計測を開始するトリガとなるイベント”
      }
    ]
  }
]

```

属性	型	説明
id	string	サービス実装の ID
local_id	string	サービス実装の内部 ID
name	string	サービス実装の名称(実装の元となったモデルの名称)
description	string	サービス実装の簡単な説明(実装の元となったモデルの説明)
device_id	array [string]	サービス実装が実際に実装されているデバイス ID の配列

以下、process_implementations[]内について、		
id	string	プロセス実装の ID
local_id	string	プロセス実装の内部 ID
name	string	プロセス実装の名称(実装の元となったモデルの名称)
description	string	プロセス実装の簡単な説明(実装の元となったモデルの説明)
以下、process_operation_implementations[]内について、		
id	string	プロセス手順実装の ID
index	number	1 からの通し番号
data_implementation_id	string	プロセスによって生成、利用、改変、または削除されるデータ実装の ID
operation_type	string	操作種別。生成"create"、利用"read"、改変"update"、削除"delete"の区分となります
以下、event_implementations[]内について、		
id	string	<p>イベント実装またはカレンダーの ID。ここで取得されるイベント実装は、該当するサービス実装に紐づくイベント実装のうち、以下の条件を満たすものです。</p> <p>①イベントプロファイルが存在し、記録可能<sup>3</sup>であるもの (event_type = monitoring であるもの)</p> <p>②イベントプロファイルが存在し、トリガ実装が存在するトリガとなるイベント実装であるもの (event_type = trigger であるもの)</p> <p>③カレンダーが存在し、トリガと</p>

<sup>3</sup> 記録可能かどうかは、連携マネージャにてイベント実装を登録する際に「記録」にチェックしてあるものを指します。

		なっているもの(id がカレンダー ID で、event_type = trigger であるもの)
local_id	string	イベント実装の内部 ID
name	string	イベントの特徴を表す名称
event_type	string	トリガ "trigger"、モニタ "monitoring" の区分となります
description	string	イベント実装の説明

### 2.7.3 サービス実装の状態通知

API 名称	サービス実装の状態通知												
概要	<p>サービス実装、プロセス実装、イベント実装の状態、内部 ID を CIOF システムに通知します。</p> <p>コントローラは、本 API を用いて配下の各実装の状態を逐次報告する必要があります。各実装の状態として、正常 (ready)、停止 (halt)、故障 (failure)、例外 (exception) の 4 つを通知することができます。通知された状態は、連携マネージャを通して確認することができます。この状態通知によってシステムが何等か介入することはありません。また、取引先から状態を確認されることもありません。</p> <p>HTTP リクエスト時に、"id"を指定した各実装の"local_id"に指定された文字列が内部 ID として CIOF システムへ登録されます。以後、サービス実装を通知する場合には、"id"を指定せずに"local_id"のみで該当の各実装にアクセスできるようになります。内部 ID は、登録時に次のような範囲でのユニーク制約がありますので、注意してください。</p> <table><tr><th>名称</th><th>ユニーク制約範囲</th></tr><tr><td>サービス実装内部 ID</td><td>コントローラ (サイト)</td></tr><tr><td>プロセス実装内部 ID</td><td>サービス実装</td></tr><tr><td>イベント実装内部 ID</td><td>プロセス実装</td></tr><tr><td>データ実装内部 ID</td><td>コントローラ (サイト)</td></tr><tr><td>データ項目実装内部 ID</td><td>データ実装</td></tr></table>	名称	ユニーク制約範囲	サービス実装内部 ID	コントローラ (サイト)	プロセス実装内部 ID	サービス実装	イベント実装内部 ID	プロセス実装	データ実装内部 ID	コントローラ (サイト)	データ項目実装内部 ID	データ実装
名称	ユニーク制約範囲												
サービス実装内部 ID	コントローラ (サイト)												
プロセス実装内部 ID	サービス実装												
イベント実装内部 ID	プロセス実装												
データ実装内部 ID	コントローラ (サイト)												
データ項目実装内部 ID	データ実装												
HTTP メソッド	PUT												
リクエスト URL	/hct/api/v2/service_implementations												
要求ヘッダ	Content-Type: application/json Authorization: Bearer token=<authorization_key> <sup>2</sup>												
要求パラメータ	<p>メッセージボディ部に、以下のパラメータを与えます。</p> <pre>[   {     "id": "10501",</pre>												

```

“local_id”: “x5VrQsPfiqrzc2J”,
“status”: “ready”,
“remarks”: “通常状態”,
“process_implementations”: [
{
“id”: “10601”,
“local_id”: “ShN69VWpC9”,
“status”: “ready”,
“remarks”: “通常状態”,
“event_implementations”: [
{
“id”: “10801”,
“local_id”: “ThSf7UaGDUnd”,
“status”: “ready”,
“remarks”: “通常状態”
}
]
}
]
}
]

```

属性	型	説明
id	string	(任意)サービス実装の ID
local_id	string	(必須)サービス実装の内部 ID
status	string	(必須)サービス実装の状態を示します。正常”ready”、停止”halt”、故障”failure”、例外”exception”の区分となります
remarks	string	(任意)通知内容
以下、(必須)process_implementations[]内について、		
id	string	(任意)プロセス実装の ID
local_id	string	(必須)プロセス実装の内部 ID
status	string	(必須) プロセス実装の状態を示します。正常”ready”、停止”halt”、故障”failure”、例外”exception”の区分となります
remarks	string	(任意)通知内容

	以下、(必須)event_implementations[]内について、		
	id	string	(任意)イベント実装の ID
	local_id	string	(必須)イベント実装の内部 ID
	status	string	(必須) イベント実装の状態を示します。正常”ready”、停止“halt”、故障“failure”、例外“exception”の区分となります
	remarks	string	(任意)通知内容

応答

2.3 節に準じます。ただし、以下の内容は本 API 独自の内容です。

ステータスコード	内容
422	状態通知対象となるサービス実装、プロセス実装、イベント実装のいずれかが存在しません。指定している local_id に誤りがあります。

応答ヘッダ

Content-Type: application/json

応答パラメータ

なし

## 2.7.4 データ実装の取得

API 名称	データ実装の取得
概要	<p>データ実装の構造を取得します。ここで取得される情報は、あらかじめ連携マネージャを利用して、CIOF システムに登録されたものです。コントローラは、本 API によって得られた情報が、配下のデータ実装に関する実際の構成（リアル世界における構成）と同じであることを確認しなければなりません。もし、実際の構成と異なる場合は、連携マネージャを利用して、CIOF システムに登録された内容を修正したうえで、コントローラによって再度修正された構成情報を取得する必要があります。</p> <p>データ実装は、CIOF システムで利用されるすべてのデータ構成モデルの実体（実装された形）です。プロセス実装によって生成、利用、改変、削除されます。</p>
HTTP メソッド	GET
URL パス	/hct/api/v2/data_implementations
要求ヘッダ	Content-Type: application/json Authorization: Bearer token=<authorization_key> <sup>2</sup>
要求パラメータ	なし
応答	2.3 節に準じます。
応答ヘッダ	Content-Type: application/json
応答パラメータ	メッセージボディ部に、以下のパラメータを返します。 <sup>4</sup> [ { "id": "10901", "local_id": "BNA67LcB", "name": "計測値",

---

<sup>4</sup> 本例では、データ項目実装が2つ存在していますが、定義により項目数が変わります。



```

“description”: “計測された値を示します”,
“data_property_implementations”: [
  {
    “id”: “11001”,
    “index”: 1,
    “local_id”: “bRxJwA4FQj”,
    “name”: “温度”,
    “description”: “温度の値を摂氏で示します”,
    “data_type”: “float”,
    “is_primary_key”: false,
    “is_required”: true,
    “default_value”: null
  },
  {
    “id”: “11002”,
    “index”: 2,
    “local_id”: “cRxJwA4FQj”,
    “name”: “湿度”,
    “description”: “湿度の値を相対湿度で示します”,
    “data_type”: “float”,
    “is_primary_key”: false,
    “is_required”: true,
    “default_value”: null
  }
]
}
]

```

属性	型	説明
id	string	データ実装の ID
local_id	string	データ実装の内部 ID
name	string	データ実装の名称(実装の元となったモデルの名称)
description	string	データ実装の簡単な説明(実装の元となったモデルの説明)
以下、data_property_implementations[]内について、		
id	string	データ項目実装の ID

index	number	1 からの通し番号
local_id	string	データ項目実装の内部 ID
name	string	データ項目実装の名称 (実装の元となったモデルの名称)
description	string	データ項目実装の簡単な説明 (実装の元となったモデルの説明)
data_type	string	データ型。文字列“string”、数値 (浮動小数) “float”、番号 (整数) “integer”、日付時刻 “datetime”、真偽値“boolean”、バイナリ “binary”、JSON“JSON”の区分となります
is_primary_key	boolean	主キーである場合に true とします
is_required	boolean	必須である場合に true とします
default_value	string	省略値

## 2.7.5 データ実装の状態通知

API 名称	データ実装の状態通知
概要	<p>データ実装の状態を CIOF システムに通知します。コントローラは、本 API を用いて配下のデータ実装の状態を逐次報告する必要があります。データ実装の状態としては、正常 (valid)、異常 (invalid)、例外 (exception) の 3 つの状態を通知することができます。通知された状態は、連携マネージャを通して確認することができます。この状態通知によってシステムが何等か介入することはありません。また、取引先から状態を確認されることもありません。</p>
HTTP メソッド	PUT
リクエスト URL	/hct/api/v2/data_implementations
要求ヘッダ	Content-Type: application/json Authorization: Bearer token=<authorization_key> <sup>2</sup>
要求パラメータ	<p>メッセージボディ部に、以下のパラメータを与えます。</p> <pre>[   {     "id": "10901",     "local_id": "BNA67LcB",     "status": "valid",     "remarks": "通常状態",     "data_property_implementations": [       {         "id": "11001",         "index": 1,         "local_id": "bRxJwA4FQj",         "status": "valid",         "remarks": "通常状態"       },       {         "id": "11002",         "index": 2,</pre>

```

    "local_id": "cRxJwA4FQj",
    "status": "valid",
    "remarks": "通常状態"
  }
]
}
]

```

属性	型	説明
id	string	(任意)データ実装の ID
local_id	string	(必須)データ実装の内部 ID
status	string	(必須)データ実装の状態を示します。正常 "valid", 異常 "invalid", 例外 "exception" の区分となります
remarks	string	(任意)通知内容
以下、(必須) data_property_implementations [] について、		
id	string	(任意) データ項目実装の ID
index	number	(必須)1 からの通し番号
local_id	string	(必須) データ項目実装の内部 ID
status	string	(必須) データ項目実装の状態を示します。正常 "valid", 異常 "invalid", 例外 "exception" の区分となります
remarks	string	(任意)通知内容

応答

2.3 節に準じます。ただし、以下の内容は本 API 独自の内容です。

ステータスコード	内容
422	状態通知対象となるデータ実装、データ項目実装のいずれかが存在しません。指定している local_id に誤りがあります。

応答ヘッダ

Content-Type: application/json

応答パラメータ	なし
---------	----

## 2.7.6 取引契約の取得

API 名称	取引契約の取得
概要	現在取引中となっている取引契約の内容を取得します。コントローラは、本 API によって、現在取引中となっている取引契約情報のリストを取得し、その情報をもとにデータ送受信を行います。取引契約は、連携マネージャ経由で締結されますが、契約締結時にコントローラへの通知がないため、コントローラ側から、本 API を用いて再読み込みを行う必要があります。
HTTP メソッド	GET
URL パス	/hct/api/v2/trade_contracts
要求ヘッダ	Content-Type: application/json Authorization: Bearer token=<authorization_key> <sup>2</sup>
要求パラメータ	なし
応答	2.3 節に準じます。
応答ヘッダ	Content-Type: application/json
応答パラメータ	メッセージボディ部に、以下のパラメータを返します。 [ { "id": "10001", "contract_type": "produce", "companion_terminal_id": "7P271SMX5M", "send_type": "push", "data_implementation_id": "10901", "data_implementation_local_id": "BNA67LcB", "service_implementation_id": "10501", "service_implementation_local_id": "x5VrQsPfqrzc2J", "process_implementation_id": "10601", "process_implementation_local_id": "ShN69VWpC9",

```

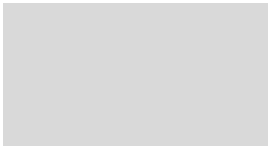
    "event_implementations_id": [
      "ThSf7UaGDUnd"
    ],
    "event_implementations_local_id": [
      "ThSf7UaGDUnd"
    ],
    "start_datetime": "2019-06-12T09:10:06.922Z",
    "end_datetime": "2021-06-12T09:10:06.922Z"
  "contract_parameters": [
    {
      "title": "price",
      "content": "1"
    }
  ]
}
]

```

属性	型	説明
id	string	取引契約の ID
contract_type	string	提供 "produce"、利用 "consume" の区分となります
companion_terminal_id	string	取引相手となるサイト ID
send_type	string	PUSH 型(push)か PULL 型(pull)かを示します
data_implementation_id	string	対応するデータ実装の ID
data_implementation_local_id	string	対応するデータ実装の内部 ID
service_implementation_id	string	対象データを提供または利用するサービス実装の ID
service_implementation_local_id	string	対象データを提供または利用するサービス実装の内部 ID
process_implementation_id	string	対象データを提供または利用するプロセス実装の

		ID。サービス実装に属します
process_implementation_local_id	string	対象データを提供または利用するプロセス実装の内部 ID。サービス実装に属します
event_implementations_id	array [string]	イベント実装の ID リスト。取引区分が提供の場合は「トリガとなるイベント実装の ID」、利用の場合は「モニタとなるイベント実装の ID」が設定されます。取引区分が提供の場合は、トリガとなるイベント実装の ID がカレンダー ID となる場合があります。
event_implementations_local_id	array [string]	イベント実装の内部 ID リスト。取引区分が提供の場合は「トリガとなるイベント実装の内部 ID」、利用の場合は「モニタとなるイベント実装の内部 ID」が設定されます。取引区分が提供の場合は、トリガとなるイベント実装の内部 ID がカレンダー ID となる場合があります。
start_datetime	string	契約が有効となる日時を Local Time で示します。有効前の場合は null
end_datetime	string	契約が無効となった日時を Local Time で示します。無効前の場合は null
以下、contract_parameters[]内について		
title	string	契約事項の契約項目名





content	string	契約事項の契約項目値
---------	--------	------------

### 2.7.7 カレンダの取得

API 名称	カレンダの取得
概要	カレンダを取得します。CIOF システムでは、コントローラに所属するカレンダ機能（日付時刻に連動する機能）をもつサービス実装が存在する場合には、連携マネージャを通してカレンダ定義を設定することができます。設定されたカレンダ定義は、CIOF システムによって自動的に実装され、該当するサービスに所属するトリガとなるイベント実装として機能します。
HTTP メソッド	GET
URL パス	/hct/api/v2/calendars
要求ヘッダ	Content-Type: application/json Authorization: Bearer token=<authorization_key> <sup>2</sup>
要求パラメータ	なし
応答	2.3 節に準じます。
応答ヘッダ	Content-Type: application/json
応答パラメータ	メッセージボディ部に、以下のパラメータを返します。 [ { "id": "W2WRGCYVNY", "name": "平日朝 8 時", "start_date": "2022-01-02T23:00:00.000Z", "days_of_week": [ "monday", "tuesday", "wednesday", "thursday", "friday" ], }, ]

```

    "weeks_of_month": null,
    "description": "平日朝 8 時を示すカレンダー",
    "interval": 1,
    "interval_type": "week",
    "end_date": null,
    "number_of_occurrences": null,
    "recurrence_time_zone": "Asia/Tokyo"
  }
]

```

属性	型	説明
id	string	定義したカレンダーを識別する ID
name	string	カレンダーの実装名称
start_time	string	カレンダーイベントを実行するにあたっての基準となる日時を Local Time で示します
days_of_week	array [string]	該当する曜日の配列
weeks_of_month	array [string]	該当する週(1,2,3,4,5)の配列
description	string	カレンダーの説明
interval	number	間隔に対応する数値
interval_type	string	間隔を示します。分“minute”、時間“hour”、日“day”、週“week”、月“month”、年“year”の区分となります
end_date	string	イベントの監視を終了する日時
number_of_occurrences	number	イベントを実行する回数。この回数おこなったら終了します
recurrence_time_zone	string	該当するタイムゾーンを示す。フォーマットは、コマンド

			<sup>5</sup> で取得したタイムゾーン ID の列挙型
--	--	--	-------------------------------------

---

<sup>5</sup> `curl -s -o - 'https://nodatime.org/TimeZones?version=2021a&format=json' | jq -r '.Zones[].Id' -`

### 2.7.8 取引データの取得

API 名称	取引データの取得												
概要	本 API により、自身のコントローラ配下のサービス実装に対して送信された取引データを取得します。URL パラメータの指定によって、サービス単位/契約単位での選択的な取引データ受信をすることができます。指定がない場合は、一括取得します。取得された取引データは連携ターミナル上から削除され、同じ取引データを 2 度受信することはできません。												
HTTP メソッド	GET												
URL パス	/hct/api/v2/messages?&service_implementation_id=<id> &service_implementation_local_id=<id> &trade_contract_id=<id>												
要求ヘッダ	Content-Type: application/json Authorization: Bearer token=<authorization_key> <sup>2</sup>												
要求パラメータ	URL パスにパラメータとして、以下を指定します。 <table><tr><th>属性</th><th>型</th><th>説明</th></tr><tr><td>service_implementation_id</td><td>string</td><td>(任意)取引データの宛先となるサービス実装の ID<sup>6</sup></td></tr><tr><td>service_implementation_local_id</td><td>string</td><td>(任意)取引データの宛先となるサービス実装の内部 ID<sup>6</sup></td></tr><tr><td>trade_contract_id</td><td>string</td><td>(任意)取引データに該当する取引契約の ID</td></tr></table>	属性	型	説明	service_implementation_id	string	(任意)取引データの宛先となるサービス実装の ID <sup>6</sup>	service_implementation_local_id	string	(任意)取引データの宛先となるサービス実装の内部 ID <sup>6</sup>	trade_contract_id	string	(任意)取引データに該当する取引契約の ID
属性	型	説明											
service_implementation_id	string	(任意)取引データの宛先となるサービス実装の ID <sup>6</sup>											
service_implementation_local_id	string	(任意)取引データの宛先となるサービス実装の内部 ID <sup>6</sup>											
trade_contract_id	string	(任意)取引データに該当する取引契約の ID											
応答	2.3 節に準じます。												
応答ヘッダ	Content-Type: application/json												

<sup>6</sup> 複数の ID を指定する場合には、例えば次のように記載します。

/hct/api/v2/messages?service\_implementation\_id=a&service\_implementation\_id=b

## 応答パラメータ

メッセージボディ部に、以下のパラメータを返します。

```
[
  {
    "id": "12345678",
    "domain_id": "81",
    "trade_contract_id": "10001",
    "request_parameter_id": "20001",
    "headers": [
      "温度",
      "湿度"
    ],
    "contents": [
      [
        "25",
        "85"
      ],
      [
        "27",
        "80"
      ],
      [
        "21",
        "76"
      ]
    ]
  }
]
```

属性	型	説明
id	string	取引データの ID。グローバルにはドメイン ID との組み合わせでユニークとなります
domain_id	string	取引データを生成したドメインの ID（連携サーバの ID）
trade_contract_id	string	取引契約の ID。グローバルにはドメイン ID との組み合わせでユニークとなります
request_	string	リクエストパラメータを識別する

	parameter_id		ための ID(PULL の場合のみ利用)
	headers	array [string]	取引データのレコードに対応するヘッダ。該当するデータ実装におけるデータ項目実装のリスト
	contents	array [array [string]]	取引データの内容 送信側で、string の値を大容量データの送信 API で取得したファイル id とした場合は、該当する大容量データファイルの取得 URL が設定されます。

## 2.7.9 取引データの送信

API 名称	取引データの送信
概要	<p>コントローラは、あらかじめ連携マネージャを用いて結んだ契約 ID を付与することで、本 API を用いて取引データを契約相手へ送信することができます。</p> <p>取引契約と異なるデータ項目を送ることはできません。contents については、契約に基づいたデータのバリデーションチェックが行われ、取引契約にて取り決めをした data_property_implementations [] 内の項目数および index によって示される項目順序が一致していない場合は、正しく通信ができません。</p> <p>リクエストパラメータによる要求があった場合は、要求のあった request_parameter_id を付与して取引データを送信します。</p> <p>取引データを送信すると、取引データ ID の応答があり、他の API を用いることで送信した取引データに紐づく履歴を確認することができます。</p>
HTTP メソッド	POST
リクエスト URL	/hct/api/v2/messages
要求ヘッダ	Content-Type: application/json Authorization: Bearer token=<authorization_key> <sup>2</sup>
要求パラメータ	<p>メッセージボディ部に、以下のパラメータを与えます。</p> <pre>{   "id": "12345678",   "request_parameter_id": "20001",   "domain_id": "81",   "trade_contract_id": "10001",   "contents": [     [       "25",       "85"     ],     [</pre>



```

        "27",
        "80"
    ],
    [
        "21",
        "76"
    ]
]
}

```

属性	型	説明
id	string	(任意)取引データの ID。グローバルにはドメイン ID との組み合わせでユニークとなります
request_parameter_id	string	(任意)リクエストパラメータの ID (PULL 型の場合のみ、相手方から送られてきた request_parameter_id に対応するデータであることを示すために付与する)
domain_id	string	(任意)取引データを生成したドメインの ID (連携サーバの ID)
trade_contract_id	string	(必須)取引契約の ID。グローバルにはドメイン ID との組み合わせでユニークとなります
contents	array [array [string]]	(任意)取引データの内容。最大 65536 レコード、最大サイズは約 10MB です string の値を大容量データの送信 API で取得したファイル id とした場合は、受信側で該当する大容量データファイルの取得 URL に変換されます。

応答

2.3 節に準じます。ただし、以下の内容は本 API 独自の内容です。

	ステータスコード	内容
	400	リクエストされた取引データ項目が、指定された取引契約による定義と一致していません。
	422	レコード数が多すぎる、またはメッセージサイズが多すぎて処理できません。

応答ヘッダ

Content-Type: application/json

応答パラメータ

メッセージボディ部に、以下のパラメータを返します。

```
{  
  "id": "12345678"  
}
```

属性	型	説明
id	string	取引データの ID。グローバルにはドメイン ID との組み合わせでユニークとなります

## 2.7.10 リクエストパラメータの取得

API 名称	リクエストパラメータの取得
概要	<p>本 API により、リクエストパラメータを取得します。リクエストパラメータには、生成 (create) と削除 (delete) の 2 種類があります。生成の場合は、相手方からデータ要求が来たことを示します。予め取引契約にて定めたルールに従い、フィルタ条件 (condition) に従ったデータを送信する必要があります。データを送信する場合は、相手方に対して、どの要求に対する回答であるかを示すに、request_parameter_id を通知します。削除の場合は、相手方から、データ削除要求が来たことを示します。data_id に該当するデータを削除する (複数のレコードに対応する場合は、すべて削除する) 必要があります。削除した結果は、サービス記録の通知によって連携ターミナルに通知する必要があります。</p>
HTTP メソッド	GET
URL パス	/hct/api/v2/requests
要求ヘッダ	Content-Type: application/json Authorization: Bearer token=<authorization_key> <sup>2</sup>
要求パラメータ	なし
応答	2.3 節に準じます。
応答ヘッダ	Content-Type: application/json
応答パラメータ	<p>メッセージボディ部に、以下のパラメータを返します。</p> <pre>[   {     "trade_contract_id": "10001",     "data_id": "12345678",     "domain_id": "81",     "request_type": "create",     "created_at": "2016-06-12T09:10:06.922Z",</pre>

```

“response_limit”: “2019-06-12T09:10:06.922Z”,
“condition”: “最新の 3 レコードをください”,
“request_parameter_id”: “20001”
}

```

```

]

```

属性	型	説明
trade_contract_id	string	リクエストパラメータの対象となる取引契約の ID。この ID とドメイン ID でグローバルにユニークとなります
data_id	string	削除リクエストの場合に削除対象となる取引データ ID
domain_id	string	取引データを生成したドメインの ID (連携サーバの ID)
request_type	string	生成"create"、削除"delete"の区分となります
created_at	string	サービス実装が本リクエストパラメータを生成した日時
response_limit	string	本リクエストに対する応答期限。なお、期限に対してシステムは介入しません
condition	string	データに対するクエリ文字列や自然言語による指定など、予め取引事業者間で定めたルールに従ったデータ取得方法の記述
request_parameter_id	string	リクエストパラメータの ID (request_type が生成の場合は、本リクエストパラメータ ID を付して応答する必要があります)

### 2.7.11 リクエストパラメータの送信

API 名称

概要

HTTP メソッド

リクエスト URL

要求ヘッダ

要求パラメータ

リクエストパラメータの送信

本 API により、リクエストパラメータを送信します。リクエストパラメータには、生成（create）と削除（delete）の 2 種類があります。生成（create）は、あらかじめ連携マネージャを用いて締結された取引契約のうち、PULL 型のみを使用することができ、相手方に取引データを要求するために使用します。取引契約にて定めたルールに従い、フィルタ条件（condition）を定めます。データ送信後に、request\_parameter\_id の応答があるので、保存しておき、相手方から取引データが送信された際に付された request\_parameter\_id と照らし合わせることで、どのリクエストに対する応答であるかを確認します。

削除（delete）は、相手方へ取引データ削除要求をする場合に使用します。data\_id に該当する取引データが削除要求の対象となります。相手方が要求した取引データを削除したかどうかは、連携マネージャから確認することができます。

POST

/hct/api/v2/requests

Content-Type: application/json  
Authorization: Bearer token=<authorization\_key><sup>2</sup>

メッセージボディ部に、以下のパラメータを与えます。  
{  
 "trade\_contract\_id": "10001",  
 "data\_id": "12345678",  
 "domain\_id": "81",  
 "request\_type": "create",  
 "created\_at": "2016-06-12T09:10:06.922Z",  
 "response\_limit": "2019-06-12T09:10:06.922Z",  
 "condition": "最新の 3 レコードをください"  
}

属性	型	説明
----	---	----

trade_contract_id	string	(必須) リクエストパラメータの対象となる取引契約の ID。 この ID とドメイン ID でグローバルにユニークとなります
data_id	string	(任意) 削除リクエストの場合に削除対象となる取引データ ID
domain_id	string	(任意) 取引データを生成したドメインの ID (連携サーバの ID)
request_type	string	(必須) 生成 "create"、削除 "delete" の区分となります
created_at	string	(必須) サービス実装が本リクエストパラメータを生成した日時
response_limit	string	(任意) 本リクエストに対する応答期限。なお、期限に対してシステムは介入しません
condition	string	(任意) データに対するクエリ文字列や自然言語による指定など、予め取引事業者間で定めたルールに従ったデータ取得方法の記述

応答

2.3 節に準じます。ただし、以下の内容は本 API 独自の内容です。

ステータスコード	内容
401	該当する取引契約において、リクエストパラメータを送信する権限がありません。生成 (create) の場合は、自身が利用者であること、削除 (delete) の場合は、自身が提供者であることが必要です。
422	必須パラメータが正しく入力されていません。ID などが間違っている可能性があります。

応答ヘッダ

Content-Type: application/json

## 応答パラメータ

メッセージボディ部に、以下のパラメータを返します。

```
{
  "domain_id": "81",
  "request_type": "create",
  "created_at": "2016-06-12T09:10:06.922Z",
  "response_limit": "2019-06-12T09:10:06.922Z",
  "trade_contract_id": "10001",
  "condition": "最新の 3 レコードをください",
  "request_parameter_id": "20001",
  "data_id": "12345678"
}
```

属性	型	説明
domain_id	string	(任意)取引データを生成したドメインの ID (連携サーバの ID)
request_type	string	(必須)生成 "create"、削除 "delete"の区分となります
created_at	string	(必須)サービス実装がこのパラメータを生成した日時のタイムスタンプ
response_limit	string	(任意)本メッセージに対する応答期限
trade_contract_id	string	(必須)前提となる取引契約の ID。この ID とドメイン ID でグローバルにユニークとなります
condition	string	(任意)データに対するクエリ文字列や自然言語による指定など、予め取引事業者間で定めたルールに従ったデータ取得方法の記述
request_parameter_id	string	(任意)リクエストパラメータを識別するための ID (PULL の場合のみ利用)
data_id	string	(任意)削除リクエストの場合の該当する取引データ ID

2.7.12 データファイルの送信

API 名称

概要

HTTP メソッド

リクエスト URL

要求ヘッダ

要求パラメータ

応答

データファイルの送信

本 API を用いてファイル形式のデータを契約相手へ送信することができます。通常取引データの送信 API では、送信する message の内容が 10MB を超えると送信できません。また送信中のエラーのリカバリ等がありません。本 API を利用することで、大容量のファイルの内容を 1 つの属性に対する値として送信します。

ファイル形式のデータを送信するには、まず本 API を用いて対象ファイルをサーバ上にアップロードし、その id を受け取ります。

取引相手には、対象とするファイルの内容ではなく、ここで受け取った id を通常取引データの送信 API を用いて送信します。

データファイルを受け取る相手側では、通常取引データの取得 API のフローに従って取引データの受信を行います。HCS により、ファイル取得用 id がファイル取得用 URL に変換しますので、相手側では、そこで受け取った URL を用いて実際のデータファイルを取得（GET）してください。

POST

/hct/api/v2/blobs

Content-Type: multipart/form-data  
Authorization: Bearer token=<authorization\_key><sup>2</sup>

属性	型	説明
blob	file	(必須)任意のファイルを設定します。

2.3 節に準じます。ただし、以下の内容は本 API 独自の内容です。

ステータスコード	内容
422	サービス記録通知対象となる取引データ、サービス実装、イベント実装のいずれかが存在しません。指定している各種 id に誤り



	があります。エラーコードの詳細は末尾に記載しています。
--	-----------------------------

応答ヘッダ

Content-Type: application/json

応答パラメータ

メッセージボディ部に、以下のパラメータを返します。

```
{  
  "id": "blob:uuid-1"  
}
```

属性	型	説明
id	string	(必須)ファイルの id。この id を取引データの送信 API において、message 送信のパラメータに設定します。

### 2.7.13 サービス記録の通知

API 名称	サービス記録の通知
概要	<p>あらかじめ連携マネージャを用いて締結された取引契約において、取引データの操作に記録が必要な場合、本 API を用いてサービス記録を通知します。</p> <p>記録対象となる取引データ ID に対して、どのイベント実装がきっかけとなって、サービス実装が操作を加えたのかを、発生時刻とともに記録します。</p> <p>event_type が use の場合には、イベント実装(event_implementation_id または event_implementation_local_id) を通知する必要があります。また、event_type が error の場合には、result と note を利用して、その内容を通知する必要があります。</p>
HTTP メソッド	POST
リクエスト URL	/hct/api/v2/service_record
要求ヘッダ	Content-Type: application/json Authorization: Bearer token=<authorization_key> <sup>2</sup>
要求パラメータ	<p>メッセージボディ部に、以下のパラメータを与えます。</p> <pre>{   "service_implementation_id": "10521",   "service_implementation_local_id": "88664423",   "process_implementation_id": "10571",   "process_implementation_local_id": "09898685ghjk",   "event_implementation_id": "10621",   "event_implementation_local_id": "38z7uw73qwaj",   "data_id": "12345678",   "event_type": "use",   "timestamp": "2020-09-17T09:10:06.922Z",   "result": "0",   "note": "値を利用し、端末に表示しました。" }</pre>

属性	型	説明
service_implementation_id	string	(任意)事実を記録したサービス実装 ID
service_implementation_local_id	string	(任意)事実を記録したサービス実装の内部 ID
process_implementation_id	string	(任意)事実を記録したプロセス実装 ID
process_implementation_local_id	string	(任意)事実を記録したプロセス実装の内部 ID
event_implementation_id	string	(任意)事実を記録したプロセスに属するイベントの ID。事実区分が利用 (use) の場合に設定
event_implementation_local_id	string	(任意) (任意)事実を記録したプロセスに属するイベントの内部 ID。事実区分が利用 (use) の場合に設定
data_id	string	(必須)対象となった取引データ ID。サービス実装のドメインとは異なる場合があります
event_type	string	(必須)記録した事実の種類。読取 "read"、保存 "store"、複製 "duplicate"、改変 "revise"、削除 "delete"、利用 "use"、エラー "error"
timestamp	string	(必須)事実を記録した時点のタイムスタンプ
result	string	(任意)事実に関する値
note	string	(任意)サービス実装が示す事実に関する記述

応答

2.3 節に準じます。ただし、以下の内容は本 API 独自の内容です。

	ステータスコード	内容
	422	サービス記録通知対象となる取引データ、サービス実装、イベント実装のいずれかが存在しません。指定している各種 id に誤りがあります。エラーコードの詳細は末尾に記載しています。

応答ヘッダ

Content-Type: application/json

応答パラメータ

メッセージボディ部に、以下のパラメータを返します。

```
{  "id": "980001",  "domain_id": "81",  "owner_id": "2",  "service_implementation_id": "10521",  "service_implementation_local_id": "88664423",  "process_implementation_id": "10571",  "process_implementation_local_id": "09898685ghjk",  "event_implementation_id": "10621",  "event_implementation_local_id": "38z7uw73qwaj",  "data_id": "12345678",  "event_type": "use",  "timestamp": "2020-09-17T09:10:06.922Z",  "result": "0",  "note": "値を利用し、端末に表示しました。"}

```

属性	型	説明
id	string	(必須)記録を識別するための ID。事実を記録したドメイン内でユニーク
domain_id	string	(必須)事実を記録したドメインの ID(連携サーバの ID)
owner_id	string	(必須)事実を記録した事業者の ID
service_implementation_id	string	(任意)事実を記録したサービス実装 ID

service_implementation_local_id	string	(必須)事実を記録したサービス実装の内部 ID
process_implementation_id	string	(任意)事実を記録したプロセス実装 ID
process_implementation_local_id	string	(任意)事実を記録したプロセス実装の内部 ID
event_implementation_id	string	(任意)事実を記録したトリガとなったイベントの ID 事実区分が利用 (use) の場合に設定
event_implementation_local_id	string	(必須)事実を記録したイベントの内部 ID。事実区分が利用 (use) の場合に設定
data_id	string	(必須)対象となった取引データ ID。サービス実装のドメインとは異なる場合がある
event_type	string	(必須)記録した事実の種類。読取"read"、保存"store"、複製"duplicate"、改変"revise"、削除"delete"、利用"use"、エラー"error"
timestamp	string	(必須)事実を記録した時点のタイムスタンプ
result	string	(任意)事実に関する値
note	string	(任意)サービス実装が示す事実に関する記述

エラーコード 422 のパラメータ仕様は、次の通りです。

パラメータ	エラーコード	説明
data	is_invalid	指定した data_id で適切な送受信データが見つからない
event_implementation_id	is_required	event_type="use"かつ event_implementation_local_id が指定されていない場合、

		event_implementation_id の値が必要だが、存在しない
process_implementation_local_id	is_required	event_type="use" かつ process_implementation_id が指定されていない かつ event_implementation_local_id が指定されている場合に、 process_implementation_local_id の値が必要だが、存在しない
service_implementation_local_id	is_required	event_type="use" かつ service_implementation_id が指定されていない かつ process_implementation_local_id と event_implementation_local_id と が 指定されている場合に、 service_implementation_local_id の値が必要だが存在しない
event_implementation	is_invalid	event_type="use" または event_implementation_id か event_implementation_local_id が指定された時に、該当なイベント実装が見つからない
process_implementation	is_invalid	process_implementation_id か process_implementation_local_id が指定された時に、該当なプロセス実装が見つからない
service_implementation	is_invalid	service_implementation_id か service_implementation_local_id が指定された時に、該当なサービス実装が見つからない
result	is_invalid	event_type が"error"の場合、result の値を指定しないといけない
note	is_invalid	event_type が"error"の場合、note の値を指定しないといけない

#### 2.7.14 サービス記録の取得

API 名称

サービス記録の取得

概要

連携マネージャの「取引実績管理」画面から確認できる、取引データ毎の取引実績を取得します。

HTTP メソッド

GET

URL パス

/hct/api/v2/service\_record

要求ヘッダ

Content-Type: application/json  
Authorization: Bearer token=<authorization\_key><sup>2</sup>

要求パラメータ

URL パスにパラメータとして、以下を指定します。

属性	型	説明
data_id	string	(任意)取引データ ID を指定します
start_datetime	string	(任意)取得したい履歴の開始日時を Local Time で指定します。指定した時刻を含みます
end_datetime	string	(任意)取得したい履歴の終了日時を Local Time で指定します。指定時刻を含みません
tail	string	(任意)最新の N 件を取得します。指定しない場合は、100 となります。
service_implementation_id	string	(任意)限定したいサービス実装の ID を指定します
service_implementation_local_id	string	(任意)限定したいサービス実装の内部 ID を指定します
event_type	string	(任意) 限定したい事実の種類を指定します。読取"read"、保存"store"、複製"duplicate"、改変"revise"、削除"delete"、利用"use"、エラー"error"のうちのいずれかとなります

応答

2.3 節に準じます。

応答ヘッダ

Content-Type: application/json

応答パラメータ

メッセージボディ部に、以下のパラメータを返します。  
(フォーマットは実際の API 応答を確認してください)

属性	型	説明
id	string	サービス記録 ID
event_type	string	(必須)記録された事実の種類。 読取"read"、保存"store"、複製 "duplicate"、改変"revise"、削除 "delete"、利用"use"、エラー "error"のうちのいずれかとなります
recorded_at	string	(必須)取引履歴が記録された 日時を Local Time で示します
service_ implementation_id	string	(必須)取引データを扱ったサ ービス実装の ID
service_ implementation_local_id	string	(必須)事実を記録したサービ ス実装の内部 ID
owner_id	string	(必須)実績を記録した事業者 ID
terminal_id	string	(必須)実績を記録したターミ ナル ID
result	string	(任意)事実に関する値
note	string	(任意)サービス実装が示す事 実に関する記述



### 3 参照規定

- [1] 一般社団法人インダストリアル・バリューチェーン・イニシアティブ。「システムの機能と実装 ―基本要件仕様―Ver. 2.1 Draft 1.0」、2021。
- [2] R.Fielding and J.Reschke. Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing, 2014. RFC 7230, <http://tools.ietf.org/html/rfc7230>.
- [3] M. Jones and D. Hardt. The OAuth 2.0 Authorization Framework: Bearer Token Usage, 2012. RFC 6750, <https://tools.ietf.org/html/rfc6750>.

### 4 変更履歴

版数	作成日	作成者	改版内容
2.1	2021.7.15	IVI 坂	昨年度資料を引継ぎ作成
2.11	2021.7.19	IVI 坂	説明を補足
2.12	2021.8.19	IVI 坂	誤植を修正、各 API の説明を拡充、CIOF エラーコードの追加予定を記載
2.13	2021.9.16	IVI 坂	誤植を修正、各 API の説明を拡充
2.14	2022.1.20	IVI 坂	誤植を修正、一部 API 仕様変更 カレンダー実装という名称を削除し、カレンダーに統一
2.15	2022.2.17	IVI 坂	連携ターミナルランチャーが出力するエラーを記載
2.16	2022.5.20	IVI 坂	取引データの送信 API について、説明を修正
2.17	2022.7.19	IVI 坂	2.7.12 サービス記録の通知の誤植修正
2.2	2023.10.12	IVI 坂	2.7.13 取引履歴の取得 API の追加 補足ストーリーの削除 システムで扱う Datetime を GMT 指定ではなく Local Time 指定とするよう変更
2.3	2025.10.30	IVI 西岡	2.7.12 データファイルの送信 API を追加 取引データの取得と取引データの送信の説明を一部追加