

Fujitsu Computing as a Service Data e-TRUST

APIリファレンスマニュアル: 別冊

まえがき

本書の目的

本書は、Fujitsu Computing as a Service Data e-TRUST(以後Data e-TRUST)のAPIリファレンスマニュアルの補足資料です。Data e-TRUSTを利用してアプリケーションおよびサービスを開発する際に活用してください。

本書の読者

本書は、Data e-TRUSTを利用してアプリケーションおよびサービスを企画、または開発される方を対象としています。本書を読むにあたって、以下の知識が必要です。

- ・ インターネットに関する基本的な知識
- ・ Web APIに関する基本的な知識
- ・ データベース(以下DB)に関する基本的な知識

また、Data e-TRUSTを利用する上であらかじめ理解する必要がある知識を、機能説明書に記載しています。機能説明書に記載の内容も併せてご確認ください。

マニュアル体系

目的・用途に合わせて、以下の関連マニュアルもお読みください。

マニュアル名称	目的・用途
機能説明書	本サービスの概要と主な機能、本サービスを利用する上で理解する必要がある情報を記載した資料です。
APIリファレンスマニュアル	Web APIを利用する際の詳細リファレンスを記載した資料です。HTML形式で記述されています。
APIリファレンスマニュアル:別冊(本書)	APIリファレンスマニュアルを補足する資料です。APIリファレンスマニュアルと併せてご確認ください。
メッセージ集	Web APIを利用する際のメッセージ内容および対処方法を記載した資料です。
注意事項・制限事項	Data e-TRUSTを利用する上での注意事項、および制限事項を記載した資料です。
リリース情報	Data e-TRUSTのリリース情報について記載した資料です。

本書の構成

本書は、以下の構成になっています。

章／付録	内容
第1章 Data e-TRUSTで利用できる機能を知る	Data e-TRUSTで提供する機能について説明します。
第2章 Data e-TRUSTにおける分散データ連携機能とは	分散データ連携機能と、当機能に含まれるAPIについて説明します。
第3章 Data e-TRUSTにおける同意管理機能とは	同意管理機能と、当機能に含まれるAPIについて説明します。
第4章 Data e-TRUSTにおける証跡・監査機能とは	証跡・監査機能と、当機能に含まれるAPIについて説明します。
第5章 Data e-TRUSTにおけるトラストシール機能とは	トラストシール機能と、当機能に含まれるAPIについて説明します。
第6章 Data e-TRUSTにおける管理機能とは	管理機能と、当機能に含まれるAPIについて説明します。

本書の表記について

本書では、Data e-TRUSTで提供するAPIを以下の形式で記載しています。
各APIの仕様詳細はAPIリファレンスマニュアルを参照してください。

記載例(テーブル定義APIの場合)

【API】

API名	HTTPメソッド	URI
テーブル定義(新規作成)	POST	/define_table
テーブル定義(更新)	PUT	/define_table
テーブル定義(削除)	POST	/delete_table

オープンソースソフトウェアまたは第三者が提供するソフトウェアの利用条件

本サービスで利用しているオープンソースソフトウェアまたは第三者が提供するソフトウェアに関する利用条件等については、ライセンス情報を参照してください。

輸出管理規制

本ドキュメントを輸出または第三者へ提供する場合は、お客様が居住する国および米国輸出管理関連法規等の規制をご確認のうえ、必要な手続をおとりください。

変更履歴

版数	日付	変更内容
第2版 (v1.1対応)	2024/4/19	以下のAPIを追加 2.11 分散データ連携機能における「クライアント通知取得」 6.2 管理機能における「エージェント削除」とは 6.3 管理機能における「エージェント一覧取得」とは 6.4 管理機能における「アプリケーションキー登録」とは 6.5 管理機能における「アプリケーションキー登録情報取得」とは 6.6 管理機能における「アプリケーションキー削除」とは 6.10 管理機能における「ユーザー作成」とは 6.11 管理機能における「ユーザー更新」とは 6.12 管理機能における「ユーザー削除」とは 6.13 管理機能における「ユーザー一覧取得」とは 6.14 管理機能における「ユーザー詳細取得」とは
初版	2023/3/24	初版公開

著作権表示

Copyright 2023-2024 FUJITSU LIMITED

目次

第1章 Data e-TRUSTで利用できる機能を知る	1
1.1 Data e-TRUSTとは	1
1.2 Data e-TRUSTで利用できる機能とは	1
1.3 Data e-TRUSTにおける各ロールと実行可能なAPI一覧	1
第2章 Data e-TRUSTにおける分散データ連携機能とは	4
2.1 分散データ連携機能における「テーブル定義」とは	4
2.2 分散データ連携機能における「テーブル定義情報取得」とは	5
2.3 分散データ連携機能における「個別データ登録・更新」とは	6
2.4 分散データ連携機能における「一括データ登録・更新」とは	6
2.5 分散データ連携機能における「データ削除」とは	6
2.6 分散データ連携機能における「データ取得」とは	6
2.7 分散データ連携機能における「データ送信」とは	6
2.8 分散データ連携機能における「データ送信キャンセル」とは	7
2.9 分散データ連携機能における「データ送信依頼」とは	7
2.10 分散データ連携機能における「データ送信依頼応答」とは	7
2.11 管理機能における「クライアント通知取得」とは	8
第3章 Data e-TRUSTにおける同意管理とは	9
3.1 同意管理における「同意回答」とは	9
3.2 同意管理における「同意情報取得」とは	9
第4章 Data e-TRUSTにおける証跡・監査機能とは	11
4.1 証跡・監査機能における「履歴登録」とは	11
4.2 証跡・監査機能における「リネージュ取得」とは	12
4.3 証跡・監査機能における「履歴検索」とは	12
4.4 証跡・監査機能における「ローカルデータ削除」とは	12
4.5 証跡・監査機能における「参照ポリシー設定」とは	12
4.6 証跡・監査機能における「改ざん検証」とは	13
第5章 Data e-TRUSTにおけるトラストシール機能とは	14
5.1 トラストシール機能における「証明書作成」とは	14
5.2 トラストシール機能における「証明書送付」とは	14
5.3 トラストシール機能における「証明書失効」とは	15
5.4 トラストシール機能における「作成済証明書取得」とは	15
5.5 トラストシール機能における「受領済証明書取得」とは	15
5.6 トラストシール機能における「トラストシール作成」とは	15
5.7 トラストシール機能における「作成済トラストシール取得」とは	16
5.8 トラストシール機能における「トラストシール検証」とは	16
第6章 Data e-TRUSTにおける管理機能とは	17
6.1 管理機能における「エージェント作成」とは	17
6.2 管理機能における「エージェント削除」とは	17
6.3 管理機能における「エージェント一覧取得」とは	18
6.4 管理機能における「アプリケーションキー登録」とは	18
6.5 管理機能における「アプリケーションキー登録情報取得」とは	18
6.6 管理機能における「アプリケーションキー削除」とは	19
6.7 管理機能における「利用履歴取得」とは	19
6.8 管理機能における「クライアント通知設定」とは	19
6.9 管理機能における「クライアント通知設定情報取得」とは	23
6.10 管理機能における「ユーザー作成」とは	23
6.11 管理機能における「ユーザー更新」とは	24
6.12 管理機能における「ユーザー削除」とは	24
6.13 管理機能における「ユーザー一覧取得」とは	24
6.14 管理機能における「ユーザー詳細取得」とは	24
付録A クライアント通知内容詳細(JSON形式)	25

第1章 Data e-TRUSTで利用できる機能を知る

1.1 Data e-TRUST とは

Data e-TRUSTは、セキュアでオンデマンドな分散データ流通の仕組みと、ブロックチェーンを拡張した柔軟でスケーラブルな台帳DBの機能を備えたサービスです。高度なデータセキュリティにより、異なるシステムやサービス間でのデータ流通の安全性と真正性を保証します。

Data e-TRUSTを利用することで、デジタル取引における個人や企業の様々な情報の真正性と、安心安全で自由なデータ流通の両立を実現します。

1.2 Data e-TRUSTで利用できる機能とは

Data e-TRUSTでは、5つの機能を提供します。これらの機能を利用することで、トラストなデータ管理を実現できます。

Data e-TRUSTで利用できる機能

- 分散データ連携機能
エージェント間で、登録されたデータを共有・連携できます。
- 同意管理機能
エージェント間でのデータ送信・同期時に、データオーナーによる同意処理ができます。
- 証跡・監査機能
エージェント間でのデータの取引を記録し、検証できます。
- トラストシール機能
データの発行者やデータ本体の改ざんがされていないことを証明できる、トラストシールを利用できます。
- 管理機能
Data e-TRUSTの各機能を利用する上で、必要となる機能を提供します。



参照

各APIのリクエストパラメーター、レスポンスについて

本書はAPIリファレンスの補足資料です。

各APIのリクエストパラメーターの詳細やリクエスト例、レスポンス例についてはAPIリファレンスを参照ください。

1.3 Data e-TRUSTにおける各ロールと実行可能なAPI一覧

Data e-TRUSTでは、ユーザーが保有するロールごとに、実行可能なAPIが異なります。

以下にロールごとに実行可能なAPI一覧を記載します。

ロールについての考え方は機能説明書3章を、各APIの詳細やロールごとに実行可能な機能などについてはAPIリファレンスおよび本書の各APIについての記述を参照してください。

表1.1 分散データ連携機能、証跡・監査機能用ロールで実行可能なAPI一覧

API種別	ロール種別(分散データ連携機能、証跡・監査機能用ロール)			
	サービス運用者 ロール	企業管理者 ロール	一般ユーザー ロール	検証者 ロール
テーブル定義	○	○	-	-
テーブル定義情報取得	○	○	-	-
個別データ登録・更新	-	○	-	-

API種別	ロール種別(分散データ連携機能、証跡・監査機能用ロール)			
	サービス運用者 ロール	企業管理者 ロール	一般ユーザー ロール	検証者 ロール
一括データ登録・更新	-	○	-	-
データ削除	-	○	-	-
データ取得	-	○	○	-
データ送信	-	○	-	-
データ送信キャンセル	-	○	-	-
データ送信依頼	-	○	-	-
データ送信依頼応答	-	○	-	-
クライアント通知取得	○	-	-	-
同意回答	-	○	○	-
同意情報取得	○	○	○	-
履歴登録	-	○	-	-
リネージュ取得	-	○	-	-
履歴検索	-	○	○	-
ローカルデータ削除	-	○	-	-
参照ポリシー設定	-	○	-	-
改ざん検証	○	○	○	○
エージェント作成	○	○	-	-
エージェント削除	○	-	-	-
エージェント一覧取得	○	-	-	-
アプリケーションキー登録	○	○	-	-
アプリケーションキー登録情報取得	○	○	○	-
アプリケーションキー削除	○	○	-	-
利用履歴取得	○	○	○	-
クライアント通知設定	-	-	-	-
クライアント通知設定情報取得	-	-	-	-
ユーザー作成	○	-	-	-
ユーザー更新	○	-	-	-
ユーザー削除	○	-	-	-
ユーザー一覧取得	○	-	-	-
ユーザー詳細取得	○	-	-	-

表1.2 トラストシール機能用ロールで実行可能なAPI一覧

API種別	ロール種別(トラストシール機能用ロール)		
	トラストシール 管理ロール	エージェント用 トラストシール 利用ロール	ユーザー用 トラストシール 利用ロール
証明書作成	-	○	○
証明書送付	-	○	○

API種別	ロール種別(トラストシール機能用ロール)		
	トラストシール 管理ロール	エージェント用 トラストシール 利用ロール	ユーザー用 トラストシール 利用ロール
証明書失効	-	○	○
作成済証明書取得	○	○	○
受領済証明書取得	○	○	○
トラストシール作成	-	○	○
作成済トラストシール取得	○	○	○
トラストシール検証	○	○	○

第2章 Data e-TRUSTにおける分散データ連携機能とは

分散データ連携機能では、登録された企業エージェント(以下エージェント)間で、各エージェントに登録されているデータを共有・連携できます。

エージェントは、企業・組織ごとに作成されるデータベースを管理する単位で、APIを利用してデータ操作、企業・組織間のデータ送受信を仲介します。また、データベースのアクセス権限は、エージェント単位で制御されます。

分散データ連携機能では、以下のAPIを提供します。

【分散データ連携機能で提供するAPI】

- ・ テーブル定義
- ・ テーブル定義情報取得
- ・ 個別データ登録・更新
- ・ 一括データ登録・更新
- ・ データ削除
- ・ データ取得
- ・ データ送信
- ・ データ送信キャンセル
- ・ データ送信依頼
- ・ データ送信依頼応答
- ・ クライアント通知取得

2.1 分散データ連携機能における「テーブル定義」とは

分散データ連携におけるテーブル定義では、DBに登録するテーブルを定義できます。

テーブル定義には、新規作成・更新・削除のAPIエンドポイントがあります。

【API】

API名	HTTPメソッド	URI
テーブル定義(新規作成)	POST	/define_table
テーブル定義(更新)	PUT	/define_table
テーブル定義(削除)	POST	/delete_table

ポイント

テーブルの同期設定について

テーブルの同期設定(リクエストボディのtable_syncキー)で、「データ送信」による他エージェントへの送信・同期の可否を設定できます。

- ・ true
同期対象(データ送信可能なテーブル)
- ・ false
同期対象外(データ送信不可能なテーブル)

データ送信については、機能説明書3.3.4を参照してください。

ポイント

データオーナー型について

カラム定義の型指定では、Data e-TRUSTの独自のデータオーナー(dataowner)型を指定できます。
データオーナー型カラムでは、各レコードの所有者および所有者の所属エージェントを指定します。
データオーナー型をカラムに含むテーブルで定義したカラム上のデータは、他エージェントへのデータ送信・同期時にデータオーナーによる同意取得が必要となります。

データ送信と同意に関しては、機能説明書3.3.4を参照してください。

ポイント

データ送受信履歴指定

データ送受信履歴指定では、データの送受信時に送受信履歴情報を記録するかどうかをテーブル毎に指定できます。
指定可能な条件は以下の5通りです。

表2.1 送受信履歴指定で指定可能な条件と記録内容

データ送受信履歴指定 (リクエストボディーのsync_historyキー)	送信履歴 を記録する	受信履歴 を記録する	プライマリーキー を記録する*
on	○	○	×
on_receive	×	○	×
on_with_key_data	○	○	○
on_receive_wirh_key_data	×	○	○
off	×	×	×

*プライマリーキーが100文字を超える場合は、先頭から最大100文字分のみ記録されます。

2.2 分散データ連携機能における「テーブル定義情報取得」とは

テーブル定義情報取得では、テーブル定義で作成されたテーブルの構成情報を取得できます。取得できるテーブル定義情報は実行者のロールによって異なります。

企業管理者ロールでは、自身が所属するエージェント(以下自エージェント)のテーブルのテーブル定義情報と、自身が所属しない他のエージェント(以下他エージェント)のテーブルの中で自エージェントが同期対象として指定されているテーブルのテーブル定義情報を取得できます。

サービス運用者ロールは、すべてのエージェント上のテーブル定義情報を取得できます。

表2.2 ロールごとに取得できるテーブル定義情報

ロール種別	自エージェント	他エージェント (同期対象外)	他エージェント (同期対象)
企業管理者ロール	○	×	○
サービス運用者ロール	○ (全エージェントのテーブル定義情報を取得可能)		

○:取得可能、×:取得不可

【API】

API名	HTTPメソッド	URI
テーブル定義情報取得	POST	/get_table_definition

2.3 分散データ連携機能における「個別データ登録・更新」とは

個別データ登録・更新では、JSON形式で指定したレコードデータを、指定のテーブルに登録・更新できます。

【API】

API名	HTTPメソッド	URI
個別データ登録・更新	POST	/register_data

2.4 分散データ連携機能における「一括データ登録・更新」とは

一括データ登録・更新では、CSV形式のファイルを利用することで、指定したテーブルにレコードを一括で登録・更新できます。

【API】

API名	HTTPメソッド	URI
一括データ登録・更新	POST	/upload_data

2.5 分散データ連携機能における「データ削除」とは

データ削除では、テーブル内の任意のレコードを削除できます。削除するレコードの条件指定も可能です。

【API】

API名	HTTPメソッド	URI
データ削除	POST	/delete_data

2.6 分散データ連携機能における「データ取得」とは

データ取得では、指定された条件でテーブルを検索して、データを取得できます。

【API】

API名	HTTPメソッド	URI
データ取得	POST	/get_data

2.7 分散データ連携機能における「データ送信」とは

データ送信では、指定したエージェントに任意のデータを送信・同期できます。

【API】

API名	HTTPメソッド	URI
データ送信	POST	/send_data



参照

データ送信処理について

エージェント間でデータ送信をする場合、送信パターンにより以下のAPIを組み合わせて送信処理を実施します。詳細は機能説明書3.3.4を参照してください。

- ・ データ送信API
- ・ データ送信依頼API
- ・ データ送信依頼応答API
- ・ 同意回答API

2.8 分散データ連携機能における「データ送信キャンセル」とは

データ送信キャンセルでは、「データ送信」により開始した他エージェントへのデータの送信・同期を停止できます。

【API】

API名	HTTPメソッド	URI
データ送信キャンセル	POST	/stop_send_data



参照

データ同期の停止方法、再開方法

データ送信キャンセルAPIを利用した、データ送信・同期の停止方法については、機能説明書の3.3.6を参照してください。データ送信キャンセルAPI実行後、データ送信・同期を再開する場合は、機能説明書の3.3.4に記載のあるデータ送信処理を参照してください。

2.9 分散データ連携機能における「データ送信依頼」とは

データ送信依頼では、他のエージェントに対して「データ送信」を依頼できます。

【API】

API名	HTTPメソッド	URI
データ送信依頼	POST	/request_send_data

2.10 分散データ連携機能における「データ送信依頼応答」とは

データ送信依頼応答では、「データ送信依頼」によるデータ送信依頼通知を受信したエージェントが、データ送信の可否をデータ送信依頼元のエージェントに応答できます。

【API】

API名	HTTPメソッド	URI
データ送信依頼応答	POST	/reply_request_send_data

2.11 管理機能における「クライアント通知取得」とは

クライアント通知取得では、Pull型でクライアント通知を取得できます。

クライアント通知設定を(クライアント通知受信用サーバー)を設定済みの場合でも、本APIは実行可能です。

【API】

API名	HTTPメソッド	URI
クライアント通知取得	GET	/get_client_notice_history



.....

クライアント通知取得によって取得可能なクライアント通知の内容については以下を参照してください。

- ・ 6.8 管理機能における「クライアント通知設定」とは
 - ・ 付録A クライアント通知内容詳細(JSON形式)
-

第3章 Data e-TRUSTにおける同意管理とは

同意管理では、エージェント間でのデータ送信・同期時における同意処理に必要な機能を提供しています。

Data e-TRUSTにおける同意処理とは、他エージェントへのデータ送信・同期時にデータ所有者のデータオーナーによる同意に基づいた送信・同期をするための処理のことです。

データオーナーによる同意処理の対象とするデータは、テーブル定義時にData e-TRUST独自のデータオーナー型をカラムに含むテーブルで定義したカラムに登録する必要があります。

データオーナー型をカラムに含むテーブルに登録したデータは、他エージェントへのデータ送信・同期時にデータオーナーによる同意を取得した場合のみ、送信・同期が可能となります。

同意管理機能では、以下のAPIを提供します。

- ・ 同意回答
- ・ 同意情報取得

3.1 同意管理における「同意回答」とは

同意回答では、他エージェントから「同意依頼通知」によって通知された「データ送信」の依頼を承諾、または拒否できます。

すでに一度以上承諾、または拒否の同意回答をした同意依頼に対して再び同意回答をすることで回答内容を撤回できます。

同意回答結果は、クライアント通知によってデータ送信元の同意依頼元エージェントへ通知されます。しかし、同意回答済みの同意依頼に再び同意回答する際に、「承諾」→「承諾」や「拒否」→「拒否」のように回答内容が変化しない場合は、同意依頼元エージェントへの通知はされません。

【API】

API名	HTTPメソッド	URI
同意回答	POST	/agreement_answer



参照

同意回答処理について

同意回答処理は、同意回答APIや同意依頼通知、データ送信APIなどを組み合わせたデータ送信処理で実施します。

詳細は機能説明書3.3.4を参照してください。



参照

クライアント通知について

クライアント通知の詳細は、本書の6.3を参照してください。

3.2 同意管理における「同意情報取得」とは

同意情報取得では、指定した条件で同意情報を取得できます。

【API】

API名	HTTPメソッド	URI
同意情報取得	POST	/get_agreement_info

ポイント

ユーザーロールごとに取得可能な同意情報の範囲

同意情報取得APIでは、ユーザーロールごとに取得可能な同意情報の範囲が異なります。
各ユーザーロールで取得可能な同意情報の範囲は以下の通りです。

ユーザーロール	取得可能な同意情報の範囲
サービス運用者ロール	全エージェント、全ユーザーの同意情報
企業管理者ロール	同意依頼元エージェントID、または同意依頼先エージェントIDが自身のエージェントIDと一致する同意情報
一般ユーザーロール	同意依頼先エージェントIDが自身のエージェントIDと一致し、かつ、同意依頼先ユーザーIDが自身のユーザーIDと一致する同意情報

第4章 Data e-TRUSTにおける証跡・監査機能とは

証跡・監査機能では、データ取引を記録し検証可能な形で管理することで、データ流通の安全性と真正性を保証します。以下の証跡・監査機能を提供します。

- ・ 履歴登録
- ・ リネージュ取得
- ・ 履歴検索
- ・ ローカルデータ削除
- ・ 参照ポリシー設定
- ・ 改ざん検証



参照

証跡・監査機能について

証跡・監査機能を利用する上で必要となる前提知識や各操作の概要、利用方法については機能説明書の4章を参照してください。

4.1 証跡・監査機能における「履歴登録」とは

証跡・監査機能における履歴登録では、指定した履歴情報を登録します。履歴登録により、履歴情報をリネージュとして扱うことができます。

【API】

API名	HTTPメソッド	URI
履歴登録	POST	/trail_registration



ポイント

履歴IDの自動採番について

リクエストに履歴ID(cdl:EventId)が指定されていない場合は、システムが履歴IDを自動で採番します。



ポイント

前履歴ID群の自動設定について

リクエストにリネージュID(cdl:LineageId)が指定され、かつ前履歴ID群(cdl:PreviousEventIdList)のリストが指定されていない場合は、リネージュの自動連結機能により、前履歴IDが前履歴ID群のリストに追加されます。



注意

任意のユーザー定義キーの予約語について

"cdl:"で始まるキー名は予約語のため、グローバルデータ部およびローカルデータ部の任意に定義可能なキー名として指定できません。

4.2 証跡・監査機能における「リネージュ取得」とは

リネージュ取得では、「履歴登録」により登録したリネージュを取得できます。

【API】

API名	HTTPメソッド	URI
リネージュ取得	GET	/trail_acquisition/{cdleventid}

4.3 証跡・監査機能における「履歴検索」とは

履歴検索では「履歴登録」で登録した履歴情報を検索できます。

履歴検索には以下のAPIエンドポイントがあります。

【API】

API名	HTTPメソッド	URI
履歴検索(検索対象:ヘッダー部)	POST	/trail_search_headers
履歴検索(検索対象:グローバルデータ部)	POST	/trail_search_globaldata
履歴検索(検索対象:ローカルデータ部、組織横断検索)	POST	/trail_search_localdata
履歴検索(検索対象:ローカルデータ部、対象組織内検索)	POST	/trail_search_localdata/{organizationid}
履歴検索(検索対象:改ざん検証部)	POST	/trail_search_verifications

4.4 証跡・監査機能における「ローカルデータ削除」とは

ローカルデータ削除では、「履歴登録」により登録した履歴情報のうち、指定した履歴情報のローカルデータ部に含まれる情報を削除できます。

【API】

API名	HTTPメソッド	URI
ローカルデータ削除	DELETE	/trail_localdata/{cdleventid}/{cdltagid}



注意

削除したローカルデータは復旧できません。

ただし、履歴登録時に改ざん検証部に追加されたローカルデータのハッシュ値は削除されないため、ローカルデータ削除後も改ざん検証APIによる改ざん検知ができます。

4.5 証跡・監査機能における「参照ポリシー設定」とは

参照ポリシー設定では、履歴情報ごとにローカルデータ部を参照するために必要な権限を管理できます。

参照ポリシー設定APIには以下のAPIエンドポイントがあります。

【API】

API名	HTTPメソッド	URI
参照ポリシー設定(設定)	POST	/trail_policies_localdata/{cdleventid}
参照ポリシー設定(削除)	PUT	/trail_policies_localdata/{cdleventid}/{cdltagid}
参照ポリシー設定(一覧取得)	GET	/trail_policies_localdata/{cdleventid}/{cdltagid}

参照

ユーザー情報非公開モードでの情報公開について

ユーザー情報非公開モードの場合でも、適切な参照ポリシーを設定することで他エージェントやロール、ユーザー単位でローカルデータ部に参照権限を付与できます。

設定例はAPIリファレンスを参照してください。

4.6 証跡・監査機能における「改ざん検証」とは

改ざん検証では、指定したリネージュ全体または個別の履歴情報に対して、改ざん有無を検証できます。

【API】

API名	HTTPメソッド	URI
改ざん検証	POST	/trail_verification

第5章 Data e-TRUSTにおけるトラストシール機能とは

Data e- TRUSTのトラストシール機能では、データの発行者やデータ自体が改ざんされていないことを証明できます。トラストシール機能を利用することで、証明機関が作成した証明書と証明対象のデータからトラストシールを作成します。トラストシールとセットでデータを送受信することで、データ自体とデータを発行した企業・組織の真正性を担保できます。トラストシール機能では以下のAPIを提供します。

【トラストシール機能で提供するAPI】

- ・ 証明書作成
- ・ 証明書送付
- ・ 証明書失効
- ・ 作成済証明書取得
- ・ 受領済証明書取得
- ・ トラストシール作成
- ・ 作成済トラストシール取得
- ・ トラストシール検証



参照

トラストシール機能について

トラストシール機能を利用する上で必要となる前提知識や利用の流れについては、機能説明書の第5章を参照してください。

5.1 トラストシール機能における「証明書作成」とは

証明書作成では、トラストシールを作成するために必要な証明書を作成します。

【API】

API名	HTTPメソッド	URI
証明書作成	POST	/create_credential



注意

作成した証明書のholderによる参照について

証明書は、作成時点ではholderによる参照ができません。holderが証明書を参照するためには、「証明書送付」の実行により参照権限を付与する必要があります。詳細は5.2を参照してください。

5.2 トラストシール機能における「証明書送付」とは

証明書送付では、holderが証明書を参照できるようにします。

【API】

API名	HTTPメソッド	URI
証明書送付	POST	/send_credential

ポイント

holderへの参照権限の付与

「証明書作成」で作成した証明書は、作成時点ではholderには参照権限がありません。そのため、holderに参照権限を設定することで「受領済証明書取得」で参照できるようにします。

5.3 トラストシール機能における「証明書失効」とは

証明書失効では、指定した発行済の証明書を失効させます。

【API】

API名	HTTPメソッド	URI
証明書失効	POST	/revoke_credential

5.4 トラストシール機能における「作成済証明書取得」とは

作成済証明書取得では、作成済の証明書を、指定した条件で取得します。

【API】

API名	HTTPメソッド	URI
作成済証明書取得	POST	/get_created_credential

5.5 トラストシール機能における「受領済証明書取得」とは

受領済証明書取得では、「証明書送付」により受領した証明書の一覧を、指定した条件で取得します。

【API】

API名	HTTPメソッド	URI
受領済み証明書取得	POST	/get_received_credential

5.6 トラストシール機能における「トラストシール作成」とは

トラストシール作成では、真正性を証明したいデータとセットで送付するための、トラストシールを作成します。

【API】

API名	HTTPメソッド	URI
トラストシール作成	POST	/create_trust_seal

ポイント

トラストシール作成に利用する証明書について

issuerによって発行された証明書を利用してトラストシールを作成する場合は、credential要素にリスト形式で証明書を指定できます。証明書は複数利用可能です。
証明書を利用せずトラストシールを作成する場合は、credential要素は不要です。

5.7 トラストシール機能における「作成済トラストシール取得」とは

作成済トラストシール取得では、作成済トラストシールの一覧を、指定した条件で取得します。

【API】

API名	HTTPメソッド	URI
作成済トラストシール取得	POST	/get_created_trust_seal

5.8 トラストシール機能における「トラストシール検証」とは

トラストシール検証では、データとセットで送付されたトラストシールに基づいて、トラストシール作成に利用された証明書がcreator本人のものか、トラストシール本体が改ざんされていないかの検証ができます。
また、creatorは「作成済トラストシール取得」を利用することで、データの真正性の検証に成功できることを確認できます。

【API】

API名	HTTPメソッド	URI
トラストシール検証	POST	/verify_trust_seal

第6章 Data e-TRUSTにおける管理機能とは

管理機能では、Data e-TRUSTの各機能を利用するための設定を管理できます。
管理機能では、以下のAPIを提供します。

- ・ エージェント作成
- ・ エージェント削除
- ・ エージェント一覧取得
- ・ アプリケーションキー登録
- ・ アプリケーションキー登録情報取得
- ・ アプリケーションキー削除
- ・ 利用履歴取得
- ・ クライアント通知設定
- ・ クライアント通知設定情報取得
- ・ ユーザー作成
- ・ ユーザー更新
- ・ ユーザー削除
- ・ ユーザー一覧取得
- ・ ユーザー詳細取得

6.1 管理機能における「エージェント作成」とは

エージェント作成では、指定したエージェントIDに対して、エージェント専用のデータベースを作成します。
1エージェントにつき、1つのデータベースを作成できます。
作成したデータベースは、エージェントIDに紐づいた権限を持つユーザーのみアクセス可能です。

【API】

API名	HTTPメソッド	URI
エージェント作成	POST	/create_agent

注意

- ・ エージェントIDは一意のものを指定する必要があります。
 - ー 一度作成に利用したエージェントIDは再利用ができません。
 - ー エージェント削除により削除済のエージェントIDも再利用はできません。
- ・ 作成したエージェントは、修正ができません。エージェント作成時には、ご注意ください。

6.2 管理機能における「エージェント削除」とは

エージェント削除では、指定したエージェントIDに対して、エージェント専用のデータベースを削除します。
エージェントIDに紐づいた権限を持つユーザーのみ削除可能です。

【API】

API名	HTTPメソッド	URI
エージェント削除	POST	/delete_agent

注意

削除対象のエージェントが「テーブル定義(作成)」したテーブルは、「テーブル定義(削除)」を利用して削除済みであることが、エージェント削除の前提条件となります。

6.3 管理機能における「エージェント一覧取得」とは

エージェント一覧取得では、エージェントの一覧情報を取得します。

【API】

API名	HTTPメソッド	URI
エージェント一覧取得	POST	/list_agent

注意

エージェントIDおよびエージェント名、状態について、完全一致、不一致、前方一致および部分一致が指定可能です。

6.4 管理機能における「アプリケーションキー登録」とは

アプリケーションキー登録では、API実行用のアプリケーションキーを、指定されたエージェントID、ユーザーID、エージェントロールを元に発行・登録します。

本APIで発行・登録したアプリケーションキーを利用することで、APIを実行できます。

IdP連携についての考え方は、機能説明書を参照してください。

【API】

API名	HTTPメソッド	URI
アプリケーションキー登録	POST	/create_approval_user

注意

本APIはアプリケーションキーを利用したAPI実行方法では実行できません。内部IdPとIDトークンを利用する実行方法で本APIを実行してください。

6.5 管理機能における「アプリケーションキー登録情報取得」とは

アプリケーションキー登録情報取得では、指定された条件に一致するアプリケーションキーの登録情報を取得します。

【API】

API名	HTTPメソッド	URI
アプリケーションキー登録情報取得	POST	/get_approval_user

6.6 管理機能における「アプリケーションキー削除」とは

アプリケーションキー削除では、指定されたAPI実行用のアプリケーションキーを削除します。

【API】

API名	HTTPメソッド	URI
アプリケーションキー削除	POST	/delete_approval_user



注意

本APIはアプリケーションキーを利用したAPI実行方法では実行できません。内部IdPとIDトークンを利用する実行方法で本APIを実行してください。

6.7 管理機能における「利用履歴取得」とは

利用履歴取得では、APIの利用履歴を取得します。

利用履歴を確認することで、なりすましや不正アクセスの検出が可能です。

【API】

API名	HTTPメソッド	URI
利用履歴取得	POST	/get_api_history



ポイント

ユーザーロールごとに取得可能な利用履歴

利用履歴取得APIでは、実行ユーザーが保有するユーザーロールにより、取得可能な利用履歴の対象が異なります。

ユーザーロール	履歴取得対象
サービス運用者ロール	全エージェント 全ユーザー
企業管理者ロール	自エージェント
一般ユーザーロール	自身(実行ユーザー)

6.8 管理機能における「クライアント通知設定」とは

クライアント通知設定では、Data e-TRUSTで発生する各種イベントをPush型で通知するための設定をします。

設定は全エージェント共通の設定として登録されます。

クライアント通知設定には以下のAPIエンドポイントがあります。

【API】

API名	HTTPメソッド	URI
クライアント通知設定(登録)	POST	/set_client_notice
クライアント通知設定(更新)	PUT	/set_client_notice

API名	HTTPメソッド	URI
クライアント通知設定(削除)	POST	/delete_client_notice

注意

クライアント通知設定の実行権限について

本機能は、Data e-TRUSTとそれを利用するサービス間を連携するための機能であり、実行にはData e-TRUSTのシステム管理権限が必要となります。

本機能を利用する場合は、Data e-TRUSTの運用管理者へご相談ください。

注意

エンドユーザーへの通知方法について

クライアント通知は、Webhookを利用して設定した通知先ペイロードURLに対して通知されます。

そのため、エンドユーザー宛に通知が必要な場合は、クライアント通知を受信したサーバー側で別途エンドユーザー宛に通知する必要があります。

ポイント

通知情報の暗号化について

クライアント通知設定の情報のうち、通知メッセージ認証コード、HTTPS接続情報は暗号化して保持されます。

〈クライアント通知設定でできること〉

- ・ 通知種別単位で、通知先となるペイロードURLが指定できます
- ・ 受信クライアント側で通知認証をするための、認証コードを設定できます。
- ・ クライアント証明書認証を利用した接続認証ができます
- ・ 複数の通知設定を一括で設定できます

〈クライアント通知の設定項目〉

- ・ 通知先ペイロードURL
Webhookの通知を受信するサーバーのURLを設定してください
- ・ ポート番号
Webhookの通知を受信するサーバーのポート番号を指定してください
- ・ Webhook認証コード設定
以下のWebhookの認証コード設定をセットで指定してください。セットで指定しない場合、エラーになります。
 - 認証コードキー
 - 認証コード
- ・ クライアント証明書設定
ペイロードURLがHTTP形式の場合は省略してください。クライアント証明書とクライアントキーをセットで指定しない場合、エラーになります。
どちらも証明書形式はBASE64 Encoded X.509です。
 - クライアント証明書

- － クライアントキー
- － CA証明書

〈クライアント通知種別〉

クライアント通知種別	クライアント通知種別(英名)	通知先エージェント	通知内容
同期情報:テーブル更新	sync_info_update_table	テーブル定義元エージェントのテーブル定義更新APIの実行により、同期している自エージェントが保持するテーブル定義が更新されたエージェント	テーブル定義元のエージェントが実行したテーブル定義(更新)APIにより、自エージェントのテーブル定義が更新されたことを示す通知です。
テーブル更新結果	result_update_table	テーブル定義更新APIを実行したエージェント	自身が実行したテーブル定義(更新)APIによる、テーブル定義の更新処理の実行結果が通知されます。
同期情報:テーブル削除	sync_info_delete_table	テーブル定義元エージェントのテーブル定義削除実行により、自エージェントのテーブル定義が削除されたエージェント	テーブル定義元のエージェントが実行したテーブル定義(削除)APIにより、自エージェントのテーブルが削除されたことを示す通知です。
テーブル削除結果	result_delete_table	テーブル定義削除APIを実行したエージェント	自身が実行したテーブル定義(削除)APIによる、テーブル定義の削除処理の実行結果が通知されます。
同期情報:個別データ登録・更新	sync_info_register_data	データ送信元エージェントの個別データ登録・更新API実行により、自エージェントのデータが更新されたエージェント	データ送信元エージェントが実行した個別データ登録・更新APIにより、自エージェントのデータが更新されたことを示す通知です。
個別データ登録・更新結果	result_register_data	個別データ登録・更新APIを実行したエージェント	自身が実行した個別データ登録・更新APIによる、同期先のエージェント上のデータも含めた登録・更新処理の実行結果が通知されます。
受付結果:一括データ登録・更新	accept_upload_data	一括データ登録・更新APIを実行したエージェント	自身が実行した一括データ登録・更新APIの、データの登録・更新の受付処理の結果が通知されます。
同期情報:一括データ登録・更新	sync_info_upload_data	データ送信元エージェントのテーブル一括データ登録・更新API実行により、自エージェントの登録データが更新されたエージェント	データ送信元エージェントが実行した一括データ登録・更新API実行により、自エージェント上のデータが更新されたことを示す通知です。
一括データ登録・更新結果	result_upload_data	一括データ登録・更新APIを実行したエージェント	自身が実行した一括データ登録・更新APIの実行による、同期先のエージェント上のデータも含めた更新処理の実行結果が通知されます。
同期情報:データ削除	sync_info_delete_data	データ送信元エージェントのデータ削除API実行により、自エージェントの登録データが更新されたエージェント	データ送信元エージェントが実行したデータ削除APIにより、自エージェントのデータが削除されたことを示す通知です。
データ削除結果	result_delete_data	データ削除APIを実行したエージェント	自身が実行したデータ削除APIの実行による、同期先のエージェント上のデータも含めた削除処理の実行結果が通知されます。
同期情報:データ送信	sync_info_send_data	他エージェントのデータ送信API実行により、自エージェントのデータが登録・更新されたエージェント	他エージェントが実行したデータ送信APIにより、自エージェントへデータが送信・同

クライアント 通知種別	クライアント 通知種別(英名)	通知先 エージェント	通知内容
			期され、参照できるようになったことを示す通知です。
データ送信結果	result_send_data	データ送信APIを実行したエージェント	自身が実行したデータ送信APIの実行による、送信先のエージェントへのデータの送信・同期処理の実行結果が通知されます。
同期情報:データ送信 キャンセル	sync_info_stop_send_data	データ送信元エージェントのデータ送信キャンセルAPI実行により、自エージェントのデータが更新・削除されたエージェント	データ送信元エージェントが実行したデータ送信キャンセルAPI実行により、自エージェントへのデータ同期が停止され、参照できなくなったことを示す通知です。
データ送信キャンセル 結果	result_stop_send_data	データ送信キャンセルAPIを実行したエージェント	自身が実行したデータ送信キャンセルAPIの実行による、同期先のエージェントへのデータの同期が停止し、データ送信のキャンセルが完了したことを示す通知です。
同意依頼	request_agreement	同意の依頼先(データオーナー)エージェント	データオーナーに対し、他エージェントへのデータの送信・同期の可否を応答するよう同意依頼をするための通知です。
同意回答	response_agreement	同意の依頼元(データ送信APIの送信元)エージェント	データ送信元エージェントに、データオーナーによって回答された、データ送信・同期への同意の回答を伝えるための通知です。
同意済みデータ送信結果	result_agreement_data	同意API発行元(データオーナー)エージェント	データオーナーに対し、同意回答APIで送信・同期の許可をしたデータの送信処理がすべて完了したことを伝えるための通知です。 一度データ送信の同意を拒否したのち、同意するパターンでの、再同意の場合も通知されます。
同意拒否結果	result_disagreement_data	同意API発行元(データオーナー)エージェント	データオーナーに対し、同意回答APIで送信・同期を拒否したデータが、回答内容通りに送信拒否され、同期されずに正常に処理が完了したことを伝えるための通知です。
同意撤回	response_withdrawal_agreement	同意の依頼元(データ送信APIの送信元)エージェント	データ送信元エージェントに対し、データオーナーによって同意が撤回されたこと伝えるための通知です。
同意撤回結果	result_cancel_agreement_data	同意API発行元(データオーナー)エージェント	データオーナーに対し、同意撤回をしたデータの送信キャンセル処理が完了したことを伝えるための通知です。
再同意	response_reagreement	同意の依頼元(データ送信APIの送信元)エージェント	データ送信元エージェントに対し、データオーナーによって同意拒否または撤回されていたデータへの、再同意許可が行われたことを伝えるための通知です。
同意回答無効化	result_revoke_agreement_answer	同意API発行元(データオーナー)エージェント	データオーナーに対し、すでに回答済の同意回答内容が無効化されたことを伝えるための通知です。 同意回答無効化通知後は、無効化された同意依頼への同意回答はできません。

クライアント通知種別	クライアント通知種別(英名)	通知先エージェント	通知内容
同意依頼無効化	result_revoke_agreement_request	同意の依頼元(データ送信APIの送信元)エージェント	データ送信元エージェントに対し、同意依頼が無効化されたことを伝えるための通知です。 同意依頼無効化通知後に、同様の同意を取得する必要がある場合は、再度データ送信をする必要があります。
データ送信依頼	request_send_data	データ送信依頼APIの宛先(送信先)エージェント	データ送信依頼APIで指定したエージェントに対し、データ送信依頼の内容を通知します。
データ送信依頼応答	response_send_data	データ送信依頼応答APIの宛先(送信先)エージェント (データ送信APIを発行するエージェント)	応答するデータ送信依頼の依頼元エージェントに対し、データ送信依頼応答の内容を通知します。



参照

クライアント通知の形式

クライアント通知ごとに、通知されるJSON形式の通知内容は付録Aを参照ください。

6.9 管理機能における「クライアント通知設定情報取得」とは

クライアント通知設定情報取得では、「クライアント通知設定」によって設定された、クライアント通知設定情報を取得できます。

【API】

API名	HTTPメソッド	URI
クライアント通知設定情報取得	GET	/get_client_notice



注意

クライアント通知設定情報取得の実行権限について

本機能は、Data e-TRUSTとそれを利用するサービス間を連携するための機能であり、実行にはData e-TRUSTのシステム管理権限が必要となります。

本機能を利用する場合は、Data e-TRUSTの運用管理者へご相談ください。

6.10 管理機能における「ユーザー作成」とは

ユーザー作成では、内部IdP(Data e-TRUSTが保有するAzure AD B2C)上にユーザーを作成します。

【API】

API名	HTTPメソッド	URI
ユーザー作成	POST	/create_user

6.11 管理機能における「ユーザー更新」とは

ユーザー更新では、内部IdP(Data e-TRUSTが保有するAzure AD B2C)上に登録済みのユーザーの情報を更新します。

【API】

API名	HTTPメソッド	URI
ユーザー更新	PUT	/update_user

6.12 管理機能における「ユーザー削除」とは

ユーザー削除では、内部IdP(Data e-TRUSTが保有するAzure AD B2C)上に登録済みのユーザーを削除します。

【API】

API名	HTTPメソッド	URI
ユーザー削除	POST	/delete_user

6.13 管理機能における「ユーザー一覧取得」とは

ユーザー一覧取得では、内部IdP(Data e-TRUSTが保有するAzure AD B2C)上に登録済みのユーザー情報の一覧を取得します。

【API】

API名	HTTPメソッド	URI
ユーザー一覧取得	POST	/list_user

6.14 管理機能における「ユーザー詳細取得」とは

ユーザー詳細取得では、内部IdP(Data e-TRUSTが保有するAzure AD B2C)上に登録済みの特定のユーザーの詳細情報を取得します。

【API】

API名	HTTPメソッド	URI
ユーザー詳細取得	POST	/get_user

付録A クライアント通知内容詳細(JSON形式)

クライアント通知機能により、通知先ペイロードURL宛に送信されるJSON形式の各クライアント通知内容を記載します。
各通知の概要に関しては、6.3を参照ください。

同期情報:テーブル更新 (sync_info_update_table)

```
{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": テーブル定義(更新) APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": テーブル定義(更新) で同期された(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "table_info": [{
        "table_name": 更新対象のテーブル名,
        "column": 定義更新カラムのオブジェクト配列(object[]),
      }]
    },
    "sync_time": 同期処理日時
  }
}
```

テーブル更新結果 (result_update_table)

```
{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": テーブル定義(更新) APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": テーブル定義(更新) で同期された(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "process_id": プロセスID,
      "table_info": [{
        "table_name": 更新対象のテーブル名,
        "column": 定義更新カラムのオブジェクト配列(object[]),
      }]
    },
  },
  "execution_result": テーブル定義更新APIの結果("OK" / "NG"),
  ※result_detailが全エージェントOKの場合のみOK "result_detail": {
    ※同期対象の各エージェントの処理結果 "エージェントID": { ※同期先エージェントID(target_agent_idに含まれる各エージェントID) をキーとする
      "result": "OK" / "NG",
      "error_info": {
        ※
        NG の場合のみ設定 "error_code": エラーコード,
        "message": メッセージ
      }
    },
    "エージェントID": {
      "result": OK / NG
    }
  }
}
```

同期情報:テーブル削除 (sync_info_delete_table)

```
{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
```

```

"detail": {
  "request_agent_id": テーブル定義(削除)APIの実行元(処理依頼元) エージェントID,
  "target_agent_id": テーブル定義(削除)で同期された(処理依頼先) エージェントIDのリスト(string[]),
  "execution_detail": {
    "table_name": 削除対象のテーブル名のリスト(string[]),
    "sync_time": 同期処理日時,
  }
}
}

```

テーブル削除結果 (result_delete_table)

```

{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": テーブル定義(削除)APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": テーブル定義(削除)で同期された(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "process_id": プロセスID,
      "table_name": 削除対象テーブルのリスト(string[])
    },
    "execution_result": テーブル定義(削除)APIの結果("OK" / "NG") ※result_detailが全エージェントOKの場合のみOK,
    "result_detail": {
      ※同期先エージェントID(target_agent_idに含まれる各エージェントID) をキーとする
      "エージェントID": {
        "result": "OK" / "NG",
        "error_info": {
          ※NG の場合のみ設定
          "error_code": エラーコード,
          "message": メッセージ
        }
      },
      "エージェントID": {
        "result": "OK" / "NG"
      }
    }
  }
}

```

同期情報：個別データ登録・更新 (sync_info_register_data)

```

{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": 個別データ登録・更新APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": データ登録・更新で同期された(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "register_type": 登録種別("insert" / "update" / "upsert")
      "table_info": [[ ※テーブル名に重複がある可能性がある(個別データ登録・更新APIでのリクエストパラメーター
"register_table"での順序通りとする)
        "table_name": 登録・更新対象のテーブル名,
        "record_count": 登録・更新対象レコード数
      ]],
      "sync_time": 同期処理日時,
    }
  }
}

```

個別データ登録・更新結果 (result_register_data)

```
{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": 個別データ登録・更新APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": API実行により影響を受けた(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "process_id": プロセスID,
      "register_type": 登録種別("insert" / "update" / "upsert"),
      "table_info": [{ ※テーブル名に重複がある可能性がある(個別データ登録・更新APIでのリクエストパラメーター
"register_table"での順序通りとする)
        "table_name": 登録・更新対象のテーブル名,
        "record_count": 登録・更新対象レコード数
      }],
    },
    "execution_result": 個別データ登録・更新APIの全エージェントを通じた実行結果("OK" / "NG") ※result_detailが全エー
ジェントOKの場合のみOK,
    "result_detail": {
      ※同期先エージェントID(target_agent_idに含まれる各エージェントID) をキーとする
      "エージェントID": {
        "result": "OK" / "NG",
        "error_info": {
          ※NG の場合のみ設定
          "error_code": エラーコード,
          "message": メッセージ
        }
      },
      "エージェントID": {
        "result": "OK" / "NG"
      }
    }
  }
}
```

受付結果：一括データ登録・更新 (accept_upload_data)

```
{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "process_id": プロセスID,
    "result": "OK" / "NG",
    "detail": {
      "message": "Success" / "Partial Success" / "Failed to insert/update/upsert",
      "err_items": [
        ※messageが "Partial Success"のときのみ存在する
        {
          "table_name": テーブル名,
          "data": [
            [{
              "colname": カラム名,
              "value": 対象外となったカラム値
            }]]
        }
      ]
    }
  }
}
```

同期情報：一括データ登録・更新 (sync_info_upload_data)

```
{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": 一括データ登録・更新APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": 一括データ登録更新で同期された(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "register_type": 登録種別("insert" / "update" / "upsert")
      "table_info": [{
        ※テーブル名に重複がある可能性がある(一括データ登録・更新APIでのリクエストパラメーター "table_name"での順序通りとする)
        "table_name": 登録・更新対象のテーブル名,
        "record_count": 登録・更新対象レコード数
      }],
      "sync_time": 同期処理日時,
    }
  }
}
```

一括データ登録・更新結果 (result_upload_data)

```
{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": 一括データ登録・更新APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": 一括データ登録更新で同期された(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "process_id": プロセスID,
      "register_type": 登録種別("insert" / "update" / "upsert"),
      "table_info": [{
        ※テーブル名に重複がある可能性がある(一括データ登録・更新APIでのリクエストパラメーター "table_name"での順序通りとする)
        "table_name": 登録・更新対象のテーブル,
        "record_count": 登録・更新対象レコード数
      }],
    },
    "execution_result": 一括データ登録・更新APIの全エージェントを通じた実行結果("OK" / "NG"),
    ※result_detailが全エージェントOKの場合のみOK
    "result_detail": {
      ※同期先エージェントID(target_agent_idに含まれる各エージェントID) をキーとする
      "エージェントID": {
        "result": "OK" / "NG",
        "error_info": {
          ※NG の場合のみ設定
          "error_code": エラーコード,
          "message": メッセージ
        }
      },
      "エージェントID": {
        "result": "OK" / "NG"
      }
    }
  }
}
```

同期情報：データ削除 (sync_info_delete_data)

```
{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
```

```

"detail": {
  "request_agent_id": データ削除APIの実行元(処理依頼元) エージェントID,
  "target_agent_id": データ削除で同期された(処理依頼先) エージェントIDのリスト(string[]),
  "execution_detail": {
    "delete_type": 削除種別("each" / "all"),
    "table_info": [{
      "table_name": 削除対象のテーブル名,
      "record_count": 登録・更新対象レコード数
    }]
    "sync_time": 同期処理日時,
  }
}
}

```

データ削除結果 (result_delete_data)

```

{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": データ削除APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": データ削除で同期された(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "process_id": プロセスID,
      "delete_type": 削除種別("each" / "all"),
      "table_info": [{
        "table_name": 削除対象のテーブル,
        "record_count": 削除対象レコード数
      }],
    },
    "execution_result": データ削除APIの全エージェントを通じた実行結果("OK" / "NG") ※result_detailが全エージェントOKの場合のみOK,
    "result_detail": {
      ※同期先エージェントID(target_agent_idに含まれる各エージェントID) をキーとする
      "エージェントID": {
        "result": "OK" / "NG",
        "error_info": {
          ※NG の場合のみ設定
          "error_code": エラーコード,
          "message": メッセージ
        }
      },
      "エージェントID": {
        "result": "OK" / "NG"
      }
    }
  }
}

```

同期情報 : データ送信 (sync_info_send_data)

```

{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": データ送信APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": データ送信で同期された(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "table_info": [{
        "table_name": 送信対象のテーブル,
        "record_count": 送信対象レコード数
      }],
    }
  }
}

```

```

    "sync_time": 同期処理日時,
  }
}

```

データ送信結果 (result_send_data)

```

{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": データ送信APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": データ送信で同期された(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "process_id": プロセスID,
      "send_info": [{
        "table_name": 送信対象のテーブル名,
        "record_count": 送信対象レコード数
      }]
    },
    "execution_result": データ送信APIの全エージェントを通じた実行結果("OK" / "NG"),
    ※ result_detailが全エージェントOKの場合のみOK
    "result_detail": {
      ※ 同期先エージェントID(target_agent_idに含まれる各エージェントID)をキーとする
      "エージェントID": {
        "result": "OK" / "NG",
        "error_info": {
          ※ NGの場合のみ設定
          "error_code": エラーコード,
          "message": メッセージ
        }
      },
      "エージェントID": {
        "result": "OK" / "NG",
      }
    }
  }
}

```

同期情報：データ送信キャンセル (sync_info_stop_send_data)

```

{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": データ送信キャンセルAPIの実行元(処理依頼元 = データオーナー) エージェントID,
    "target_agent_id": データ送信キャンセルで同期された(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "table_info": [{
        "table_name": 送信キャンセル対象のテーブル,
        "record_count": 送信キャンセル対象レコード数
      }],
      "sync_time": 同期処理日時
    }
  }
}

```

データ送信キャンセル結果 (result_stop_send_data)

```

{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),

```

```

"detail": {
  "request_agent_id": データ送信キャンセルAPIの実行元(処理依頼元) エージェントID,
  "target_agent_id": API実行により影響を受けた(処理依頼先) エージェントIDのリスト(string[]),
  "execution_detail": {
    "process_id": プロセスID,
    "send_info": [{
      "table_name": 送信キャンセル対象のテーブル名,
      "record_count": 送信キャンセル対象レコード数
    }]
  },
  "execution_result": データ送信キャンセルAPIの全エージェントを通じた実行結果("OK" / "NG"),
  ※result_detailが全エージェントOKの場合のみOK
  "result_detail": {
    ※同期先エージェントID(target_agent_idに含まれる各エージェントID) をキーとする "エージェントID": {
      "result": "OK" / "NG",
      "error_info": {
        ※NGの場合のみ設定
        "error_code": エラーコード,
        "message": メッセージ
      }
    },
    "エージェントID": {
      "result": "OK" / "NG",
    }
  }
}
}

```

同意依頼 (request_agreement)

```

{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": データ送信APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": データ送信で送信先となっている(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "request_agreement": [※同意依頼レコード情報(object[])]
      {
        "agreement_id": 同意依頼ID,
        "agr_req_src_agent_id": 同意依頼元エージェントID,
        "agr_req_dest_agent_id": 同意依頼先エージェントID,
        "agr_req_dest_user_id": 同意依頼先ユーザーID,
        "send_dest_agent_id": 送信先エージェントID(データ送信の送信先)
        "agr_target_table_name": 同意対象テーブル名,
        "req_column_info": [ ※ 同意依頼対象のカラム情報,
          {
            "name": カラム名,
            "owner_select": オーナー選択(true: 任意送信カラム, false: 必須送信カラム)
          },
        ],
        "agreement_sentence": 同意依頼文
      }
    ]
  }
}
}

```

同意回答 (response_agreement)

```
{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": データ送信APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": データ送信で送信先となっている(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "agreement_answer": [{
        "agreement_id": 同意依頼ID,
        "agr_answer_dest_agent_id": 同意回答先エージェントID(同意依頼元エージェントID),
        "agr_answer_src_agent_id": 同意回答元エージェントID(同意依頼先エージェントID),
        "agr_answer_src_user_id": 同意回答元ユーザーID(同意依頼先ユーザーID),
        "send_dest_agent_id": 送信先エージェントID(データ送信の送信先),
        "agr_target_table_name": 同意対象テーブル名,
        "agr_answer": 同意回答("accept": 承諾, "reject": 拒否),
        "answer_column_info": [{
          ※同意カラム情報(どのカラムの送信に同意したか). 同意回答が拒否のときはnull
          "name": カラム名,
          "answer": 承諾 / 拒否("accept": 承諾, "reject": 拒否),
        }]
      }]
    }
  }
}
```

同意済みデータ送信結果 (result_agreement_data)

```
{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": データ送信APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": データ送信で送信先となっている(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "agreement_answer": 同意回答内容(※省略。 同意撤回のagreement_answerと同じ),
    },
    "agreement_process_type": 同意種別。※同意済みデータ送信結果通知では、"agree",
    "execution_result": 同意済みデータ送信処理の全エージェントを通じた実行結果("OK" / "NG"),
    "result_detail": {
      "エージェントID": {
        "result": "OK" / "NG",
        "agreement_id": 同意依頼IDの配列(string[]),
        "error_info": {
          ※NGの場合のみ設定
          "error_code": エラーコード,
          "message": メッセージ
        }
      },
      "エージェントID": {
        "result": "OK" / "NG",
        "agreement_id": 同意依頼IDの配列(string[])
      },
    }
  }
}
```

同意拒否結果 (result_disagreement_data)

```
{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
```

```

"detail": {
  "request_agent_id": データ送信APIの実行元(処理依頼元) エージェントID,
  "target_agent_id": データ送信で送信先となっている(処理依頼先) エージェントIDのリスト(string[]),
  "execution_detail": {
    "agreement_answer": 同意回答内容(※省略。 同意撤回のagreement_answerと同じ),
  },
  "agreement_process_type": 同意種別,
  ※同意拒否結果通知では、 "reject"
  "execution_result": 同意拒否処理の全エージェントを通じた実行結果("OK" / "NG"),
  "result_detail": {
    "エージェントID": {
      "result": "OK" / "NG",
      "agreement_id": 同意依頼IDの配列(string[]),
      "error_info": {
        ※
        NGの場合のみ設定 "error_code": エラーコード,
        "message": メッセージ
      }
    },
    "エージェントID": {
      "result": "OK" / "NG",
      "agreement_id": 同意依頼IDの配列(string[])
    },
  }
}
}

```

同意撤回 (response_withdrawal_agreement)

```

{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": データ送信APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": データ送信で送信先となっている(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "agreement_answer": [{
        "agreement_id": 同意依頼ID,
        "agr_answer_dest_agent_id": 同意回答先エージェントID(同意依頼元エージェントID),
        "agr_answer_src_agent_id": 同意回答元エージェントID(同意依頼先エージェントID),
        "agr_answer_src_user_id": 同意回答元ユーザーID(同意依頼先ユーザーID),
        "send_dest_agent_id": 送信先エージェントID(データ送信の送信先),
        "agr_target_table_name": 同意対象テーブル名,
        "agr_answer": 同意回答("accept": 承諾, "reject": 拒否). 同意撤回のため必ず "reject"となる,
        "answer_column_info": ※同意カラム情報(どのカラムの送信に同意したか). 同意撤回のため必ずnullとなる.
      }]
    }
  }
}

```

同意撤回結果 (result_cancel_agreement_data)

```

{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": データ送信APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": データ送信で送信先となっている(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "agreement_answer": 同意回答内容(※省略。 同意撤回のagreement_answerと同じ),
    },
    "agreement_process_type": 同意種別,
    ※同意撤回結果通知では、 "revoke"
  }
}

```

```

"execution_result": 同意撤回処理の全エージェントを通じた実行結果("OK" / "NG"),
"result_detail": {
  "エージェントID": {
    "result": "OK" / "NG",
    "agreement_id": 同意依頼IDの配列(string[]),
    "error_info": {
      ※NGの場合のみ設定
      "error_code": エラーコード,
      "message": メッセージ
    }
  },
  "エージェントID": {
    "result": "OK" / "NG",
    "agreement_id": 同意依頼IDの配列(string[])
  },
}
}
}

```

再同意 (response_reagreement)

```

{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": データ送信APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": データ送信で送信先となっている(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "agreement_answer": [{
        "agreement_id": 同意依頼ID,
        "agr_answer_dest_agent_id": 同意回答先エージェントID(同意依頼元エージェントID),
        "agr_answer_src_agent_id": 同意回答元エージェントID(同意依頼先エージェントID),
        "agr_answer_src_user_id": 同意回答元ユーザーID(同意依頼先ユーザーID),
        "send_dest_agent_id": 送信先エージェントID(データ送信の送信先),
        "agr_target_table_name": 同意対象テーブル名,
        "agr_answer": 同意回答("accept": 承諾, "reject": 拒否).再同意のため必ず "accept",
        "answer_column_info": ※同意カラム情報(どのカラムの送信に再同意したか) [{
          "name": カラム名,
          "answer": 承諾 / 拒否("accept": 承諾, "reject": 拒否),
        }]
      }]
    }
  }
}

```

同意回答無効化 (result_revoke_agreement_answer)

```

{
  "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
  "detail": {
    "request_agent_id": データ送信APIの実行元(処理依頼元) エージェントID,
    "target_agent_id": データ送信で送信先となっている(処理依頼先) エージェントIDのリスト(string[]),
    "execution_detail": {
      "agreement_answer": [{
        "agreement_id": 同意依頼ID,
        "agr_answer_dest_agent_id": 同意回答先エージェントID(同意依頼元エージェントID),
        "agr_answer_src_agent_id": 同意回答元エージェントID(同意依頼先エージェントID),
        "agr_answer_src_user_id": 同意回答元ユーザーID(同意依頼先ユーザーID),
        "send_dest_agent_id": 送信先エージェントID(データ送信の送信先),
        "agr_target_table_name": 同意対象テーブル名,
        "agr_answer": 同意回答("accept": 承諾, "reject": 拒否),
      }]
    }
  }
}

```

```

        "answer_column_info": ※同意カラム情報(どのカラムの送信に同意したか). 同意回答が拒否のときはnull [{
            "name": カラム名,
            "answer": 承諾 / 拒否("accept": 承諾, "reject": 拒否),
        }]
    }
}

```

同意依頼無効化 (result_revoke_agreement_request)

```

{
    "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
    "detail": {
        "request_agent_id": データ送信APIの実行元(処理依頼元) エージェントID,
        "target_agent_id": データ送信で送信先となっている(処理依頼先) エージェントIDのリスト(string[]),
        "execution_detail": {
            "request_agreement": ※同意依頼レコード情報(object[]) [{
                "agreement_id": 同意依頼ID,
                "agr_req_src_agent_id": 同意依頼元エージェントID,
                "agr_req_dest_agent_id": 同意依頼先エージェントID,
                "agr_req_dest_user_id": 同意依頼先ユーザーID(同意依頼先ユーザーID),
                "send_dest_agent_id": 送信先エージェントID(データ送信の送信先)
                "agr_target_table_name": 同意対象テーブル名,
                "req_column_info": [※同意依頼対象のカラム情報,
                    {
                        "name": カラム名,
                        "owner_select": オーナー選択(true: 任意送信カラム, false: 必須送信カラム)
                    },
                ],
                "agreement_sentence": 同意依頼文
            }]
        }
    }
}

```

データ送信依頼 (request_send_data)

```

{
    "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
    "detail": {
        "request_agent_id": データ送信依頼APIの実行元(処理依頼元) エージェントID,
        "target_agent_id": データ送信依頼メッセージの送信先(処理依頼先) エージェントIDのリスト(string[]),
        "execution_detail": {
            "notification_id": データ送信依頼通知ID,
            "request_send_data": データ送信依頼情報オブジェクト(APIのリクエストで指定),
            "purpose_text": 利用目的
        }
    }
}

```

データ送信依頼応答 (response_send_data)

```

{
    "notification_target_agent_id": 通知先エージェントIDのリスト(string[]),
    "detail": {
        "request_agent_id": データ送信依頼応答APIの実行元(処理依頼元) エージェントID,
        "target_agent_id": データ送信依頼応答メッセージの送信先(処理依頼先) エージェントIDのリスト(string[]),
        "execution_detail": {

```

```
    "notification_id": データ送信依頼通知ID,  
    "reply_request_send_data": 依頼応答情報データオブジェクト(APIのリクエストで指定),  
    "additional_text": 追加テキスト  
  }  
}
```