



Salesforce CDP API 開発者ガイド

ド

バージョン 53.0, Winter '22



本書の英語版と翻訳版で相違がある場合は英語版を優先するものとします。

©Copyright 2000–2021 salesforce.com, inc. All rights reserved. Salesforce およびその他の名称や商標は、salesforce.com, inc. の登録商標です。本ドキュメントに記載されたその他の商標は、各社に所有権があります。

目次

第 1 章: クイックスタート	1
第 2 章: Salesforce CDP の追加リソース	3
Salesforce CDP アクセストークンの取得と交換	4
プロファイル、クエリ、計算済みインサイトに関する Salesforce CDP API の制限	5
Salesforce CDP API の有効期限	5
第 3 章: Salesforce Interactions SDK	6
イベント仕様	8
イベント構造	8
カートインタラクション	10
カタログインタラクション	13
注文インタラクション	15
品目データ	18
ユーザデータ	18
アカウントデータ	21
同意データ	22
API リファレンス	23
初期化	24
データ収集	26
ID	27
Consent	29
サイトマップ	30
インテグレーション	45
デバッグ	48
Salesforce Interactions SDK の Salesforce CDP モジュール	49
Web コネクタスキーマへの SDK イベントの変換	51
プロファイルイベントのルーティング	69
カスタムイベント	72
匿名および既知のプロファイルデータ	73
第 5 章: Salesforce CDP Mobile SDK	84
第 6 章: Salesforce CDP メタデータ API	95
第 7 章: Salesforce CDP 取り込み API	98
使用開始	100
ストリーミング挿入に関するウォークスルー	102
一括挿入のウォークスルー	103
ストリーミング取り込み	105

レコードの挿入	106
レコードの削除	107
同期レコード検証	107
一括取り込み	109
ジョブの作成	110
ジョブデータのアップロード	111
ジョブの終了または中止	112
ジョブの削除	114
ジョブ情報の取得	115
すべてのジョブの取得	116
第 8 章: Salesforce CDP MuleSoft コネクタの実装	118
第 9 章: Amazon S3 データストリームの作成	119
Post /services/data/v51.0/ui-api/records	121
第 10 章: Salesforce CDP Profile API ライブラリ	124
GET /api/v1/profile/metadata	126
GET /api/v1/profile/metadata/{dataModelName}	128
GET /api/v1/profile/{dataModelName}	130
GET /api/v1/profile/{dataModelName}/{id}	132
GET /api/v1/profile/{dataModelName}/{id}/{childDataModelName}	134
GET /api/v1/profile/{dataModelName}/{id}/calculated-insights/{ci-name}	136
第 11 章: Salesforce CDP Query API ライブラリ	139
POST /api/v1/query	141
第 12 章: Salesforce CDP 計算済みインサイト API ライブラリ	143
GET /api/v1/insight/metadata/{ci-name}	144
GET /api/v1/insight/calculated-insights/{ci-name}	145

第1章 クイックスタート

Salesforce CDP アプリケーションは、一部の Salesforce CDP アプリケーションタスクをプログラムで実行できるようにするカスタマーデータプラットフォームです。

Salesforce CDP アプリケーションに慣れていない場合は、これらのコールを使用する前に [Salesforce CDP オンラインヘルプ](#) を参照してください。

Salesforce CDP SDK と API ライブラリ

Salesforce CDP API コールは、組織に特有のエンドポイントを使用します。エンドポイントの生成については、API ライブラリリファレンスのリストの後で説明します。

各 SDK または API ライブラリの概要ドキュメントに記載されている設定手順も確認してください。

[Salesforce CDP Web SDK](#)

これらのコールは Salesforce CDP アプリケーションインターフェースから使用することで最大限に活用できますが、自動生成されたエンドポイントを使用してメソッドを外部からコールすることもできます。

[Salesforce CDP Mobile SDK](#)

これらのコールは Salesforce CDP Mobile アプリケーションインターフェースから使用することで最大限に活用できますが、自動生成されたエンドポイントを使用してメソッドを外部からコールすることもできます。

コールライブラリ

[Salesforce CDP Profile API ライブラリ](#)

これらの API コールを外部の Web アプリケーションまたはモバイルアプリケーションに含めることで、顧客のプロファイル情報を参照できます。

[Salesforce CDP Query API ライブラリ](#)

これらの API コールを外部の Web アプリケーションまたはモバイルアプリケーションに含めて、データの抽出、外部アプリケーションの統合、データレイクに対するインタラクティブな照会など、さまざまな使用事例をサポートできます。

[Salesforce CDP 計算済みインサイト API ライブラリ](#)

クリックスタート

これらの API コールを外部の Web アプリケーションまたはモバイルアプリケーションに含めて、選択した SQL ディメンションと基準を含む計算済みインサイトを照会できます。検索条件も使用できます（省略可能）。

Salesforce API コールの Salesforce CDP パラメータ

これらの Salesforce CDP パラメータは、Salesforce REST API の Consent または Query コールで使用します。

[Salesforce Consent API](#)

[Salesforce Query API](#)

Salesforce CDP の Salesforce メタデータ API コール

Salesforce CDP のメタデータオブジェクトの説明は、Salesforce 開発者向けドキュメントの「メタデータ API」セクションにアルファベット順に掲載されています。

[メタデータ API](#)

Salesforce CDP の設定

Salesforce CDP API を使用するには、Salesforce CDP アプリケーションを実装する必要があります。Salesforce CDP アプリケーションの設定手順は、『[Salesforce CDP 実装ガイド](#)』に掲載されています。

データソースの設定

Salesforce CDP インスタンスを設定したら、Salesforce CDP からデータの取り込みと有効化の範囲を制御できます。

データソースを設定するには、ユーザインターフェースで[アプリケーションコネクタを作成](#)して、データの対応付けとセグメント用のデータを取り込む場所を Salesforce CDP アプリケーションに指示します。

自動生成されるエンドポイント

Salesforce CDP アプリケーションは、システムが生成した一意のエンドポイントを各組織に割り当てます。

Salesforce CDP でエンドポイントを取得したら、そのエンドポイントを自分の組織の API コールで使用します。他の Salesforce CDP ユーザがそのエンドポイントを使用して API 要求を実行することはありません。

ユーザは c360a.salesforce.com エンドポイント構造に基づいた自分のエンドポイントのみを使用できます。

第 2 章 Salesforce CDP の追加リソース

トピック:

- Salesforce CDP アクセストークンの取得と交換
- プロファイル、クエリ、計算済みインサイトに関する Salesforce CDP API の制限
- Salesforce CDP API の有効期限

Salesforce CDP 開発者向けの追加のドキュメントリソースです。

Salesforce CDP アクセストークンの取得と交換

以下の例では、Salesforce CDP アクセストークンを取得して交換する方法を説明しています。

アクセストークンの取得

次の例では、アクセストークンを要求しています。

```
POST https://login.salesforce.com/services/oauth2/token  
Content-Type: x-www-form-urlencoded  
grant_type=urn:ietf:params:oauth:grant-type:jwt-bearer  
-----  
client_id=00Dxx0000006JFr!AQEAQl7SvQ1t..2yYtyM0jfXtMNriivdMpTYE3mThqATnm5YqPTX.xnzUC45zd0HEIU3tc9GdHLqWOFVprsgszWdYMPRUoOy  
-----
```

応答形式:

```
{  
  "access_token":  
    "00Dxx0000006JFr!AQEAQl7SvQ1t..2yYtyM0jfXtMNriivdMpTYE3mThqATnm5YqPTX.xnzUC45zd0HEIU3tc9GdHLqWOFVprsgszWdYMPRUoOy",  
  "scope": "cdpquery api cdpprofile",  
  "instance_url": "http://orgmain2.localhost.internal.salesforce.com:6109",  
  "id":  
    "http://sukumar-ltm.internal.salesforce.com:6109/id/00Dxx0000006JFrEAM/005xx000001X9hBAAS",  
  "token_type": "Bearer"  
}
```

アサーション JWT を作成します。

アクセストークンと Salesforce CDP トークンとの交換

次の例では、Salesforce CDP トークンを要求しています。

```
POST <Instance URL>/services/a360/token  
Content-Type : x-www-form-urlencoded  
grant_type=urn:salesforce:grant-type:external:cdp  
&subject_token=<CORE ACCESS TOKEN>  
&subject_token_type=urn:ietf:params:oauth:token-type:access_token
```

応答形式:

```
{
  "access_token": "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "instance_url": "https://a360.cdp.cdp001.dev1-uswest2.aws.sfdc.cl",
  "token_type": "Bearer",
  "issued_token_type": "urn:ietf:params:oauth:token-type:jwt",
  "expires_in": 7191
}
```

応答に含まれている access_token を、Salesforce CDP URL である instance_url として使用します。

プロファイル、クエリ、計算済みインサイトに関する Salesforce CDP API の制限

API には、プロファイル、クエリ、計算済みインサイトの 3 つの異なるクラスがあります。現時点では、制限は API ごとではなく Salesforce エディションごとに決められています。したがって、以下の制限はすべての API で Salesforce CDP を使用する場合に適用されます。制限は、プロファイル、クエリ、計算済みインサイト API 全体で累積して計算されます。

Salesforce CDP エディション	1 日のクエリ数
Corporate	25000
Enterprise	75000

Salesforce CDP API の有効期限

Salesforce は、各 API バージョンを最初のリリース日から最低 3 年間サポートします。API の品質およびパフォーマンスを充実させ、改善するために、3 年を超えるバージョンのサポートは停止される場合があります。

API バージョンに廃止の予定がある場合、サポートが終了する最低 1 年前までに事前通知されます。Salesforce は、廃止予定の API バージョンを使用するお客様に直接通知します。

第3章

Salesforce Interactions SDK

トピック:

- イベント仕様
- API リファレンス
- Salesforce Interactions SDK の Salesforce CDP モジュール

Salesforce Interactions SDK は、拡張可能なデータキャプチャおよび収集フレームワークです。これを使用して、Web サイトでさまざまな種別の顧客インタラクションを追跡し、情報を Salesforce に送信します。その後、この情報を使用して、Web 訪問者の行動プロファイル、利用者のセグメンテーション、Salesforce インテグレーションを構築できます。

Salesforce Interactions SDK には、次の機能があります。

- 顧客インタラクションの追跡
- 匿名 ID および指定 ID を追跡するための ID および Cookie 管理
- 同意管理インテグレーションフック
- サイトマップを使用した設定に基づく計測
- カスタム機能を追加するためのインテグレーションフック

要件

Web サイトの顧客には、ES6 互換コードを実行できるブラウザが必要です。

ユーザインタラクションの追跡

`sendEvent` API コールは、Web サイトの顧客からインタラクションを取得し、関連付けられたデータを収集する方法です。インタラクションイベントには、`View Catalog Object` (カタログオブジェクトを参照) などの指定識別子、`catalogObject` や `order` などの必須値、追跡されたユーザ定義値が含まれます。たとえば、次のような `View Catalog Object` (カタログオブジェクトを参照) イベントがあります。

```
SalesforceInteractions.sendEvent({
  interaction : {
    name : "View Catalog Object",
    catalogObject: {
      type : "Product",
      id : "65e4e737",
      attributes: {
        description: "Shoes"
      }
    }
  }
})
```

Salesforce Interactions SDK には、Salesforce 全体ですぐに使用できるイベントデータモデルも付属しています。

ID と Cookie の管理

Salesforce Interactions SDK では、ファーストパーティ Cookie を使用して、または Web サイトから ID 情報が渡されることで、匿名 ID と指定 ID を追跡できます。Cookie の先頭には `_sfid` のプレフィックスが付けられ、末尾は、Cookie が属しているドメインに固有の文字列で終わります。Cookie と ID の動作をカスタマイズするには、Salesforce Interactions SDK で設定を使用するか、[ID](#)(ページ 27) API をコールすることができます。

同意管理

Web サイトのすべての訪問者と顧客が Cookie 追跡に同意するとは限りません。同意管理をサポートするために、Salesforce Interactions SDK は、顧客が追跡に同意した場合にのみイベントを送信するように設計されています。同意が取り消されると、Web SDK でその設定が取得され、イベントの転送が直ちに停止されます。たとえば、次のメソッドでは同意が Salesforce Interactions SDK に渡されます。

```
SalesforceInteractions.init({
  consents: [
    {
      provider: 'ExampleProvider',
      purpose: 'Tracking',
      status: 'Opt In'
    }
  ]
})
```

顧客の同意の取得は、Salesforce Interactions SDK で設定を使用するか、[同意](#)(ページ 29) API をコールして処理することができます。

サイトマップ[®]

Salesforce Interactions SDK サイトマップは、Web SDK 内で実行され、Web SDK と共にリリースされる、設定に基づくインテグレーションレイヤです。主要なビジネスオブジェクトをインタラクションイベントに対応付ける Web SDK から提供された API を使用して特定のサイトと統合するには、サイトマップをリリースします。

関連リソース

- 古い Web SDK バージョンを実装している場合は、[「Salesforce CDP Web SDK」](#) を参照してください。

イベント仕様

Salesforce Interactions SDK のイベント仕様。

イベント構造

Salesforce Interactions Web SDK で収集されるイベントには、システムで自動的に設定されるプロパティと、ユーザが設定する省略可能なプロパティが含まれます。

カートインタラクション

カートインタラクションでは、顧客がオンラインショッピングカートの内容を変更できます。

カタログインタラクション

カタログインタラクションを使用して、顧客がさまざまな追跡項目を操作したことを取得します。たとえば、商品やブログ投稿への操作です。

注文インタラクション

注文インタラクションは、新規注文、進行中の注文、または完了した注文を管理するためにユーザが実行したアクションを取得する E コマースイベントです。

品目データ

品目は、購入可能な品目を表すことを目的とし、カートおよび注文インタラクションで使用されます。

ユーザデータ

Salesforce Interactions Web SDK を使用して、インタラクションを関連付けてまたは関連付けずにユーザのプロファイルデータを追跡します。

アカウントデータ

アカウントデータはアカウントレベルのユーザ情報を表します。アカウント項目を使用すると、1つのアカウントで複数の顧客のデータを取得できます。

同意データ

同意データは、Salesforce Interactions SDK での同意の付与または取り消しに使用されます。顧客が追跡へのオプトインを選択した場合、この同意の詳細が、最初に追跡されたイベントのイベント構造に付加されます。

イベント構造

Salesforce Interactions Web SDK で収集されるイベントには、システムで自動的に設定されるプロパティと、ユーザが設定する省略可能なプロパティが含まれます。

項目	提供者	内容	説明
account	開発者	アカウント	アカウントのプロファイルイベントデータを取得します。
consents	SDK	同意	追跡への顧客の同意後または同意の取り消し時に取得された最初のイベン

項目	提供者	内容	説明
			トに付加された同意値のリストが含まれます。
interaction	開発者	インタラクション	顧客エンゲージメントイベントデータを取得します。
pageView	SDK	ページビュー	ページの読み込み中にイベントが取得されたかどうかを示します。
source	SDK	ソース	ページに関するイベントメタデータが含まれます。
time	SDK	時間	イベントのタイムスタンプ(UNIXエポックからのミリ秒単位の経過時間)を表示します。
user	開発者	ユーザ	顧客のプロファイルイベントデータを取得します。

開発者が提供するデータ

interaction 項目は、顧客エンゲージメントデータを取得するための項目です。少なくとも、*interaction* には有効な *name* 値が必要です。Salesforce Interactions Web SDK には、次のような特化されたインタラクション種別がいくつか付属します。

- カートインタラクション (ページ 10)
- カタログインタラクション (ページ 13)
- 注文インタラクション (ページ 15)

user 項目では、顧客のユーザ (ページ 18) プロファイルデータを取得します。

account 項目では、アカウント (ページ 21) レベルでプロファイルデータを取得します。アカウント項目を使用すると、1つのアカウントで複数の顧客のデータを取得できます。

 **例:** Salesforce Interactions Web SDK を使用して、サイトの計測環境を整えて、行動データとプロファイルデータの両方を含むイベントを取得します。

```
SalesforceInteractions.sendEvent({
  interaction: {
    name: "View Catalog Object",
    catalogObject: {
      type: "Product",
      id: "product-1"
    }
  }
})
```

```
        },
        user: {
            attributes: {
                email: "user@domain.com"
            }
        }
    })
}
```

このイベント構造で取得された結果は次のように表示されます。

```
{
    interaction: {
        name: "View Catalog Object",
        catalogObject: {
            type: "Product",
            id: "product-1"
        }
    },
    source: {
        pageType: "homepage",
        url: "https://my.domain.com/home",
        urlReferrer: "https://my.domain.com/search",
        channel: "Web",
        locale: "en_US"
    },
    user: {
        anonymousId: "aa19dc4fc636d8",
        attributes: {
            email: "user@domain.com"
        }
    },
    consents: [
        {
            purpose: 'Tracking',
            provider: 'Example Provider',
            status: 'Opt In'
        }
    ],
    pageView: false,
    time: 1628700769593,
}
```

カートインタラクション

カートインタラクションでは、顧客がオンラインショッピングカートの内容を変更できます。

カートインタラクションには次の2つの形式があります。

- **単一品目:**顧客のカート内の個々の品目を変更します。
- **複数品目:**顧客のカートに対して一括操作を行います。

单一品目

次のインタラクション名は、個々の品目に影響するカートインタラクションで使用されます。

インタラクション名	値
SalesforceInteractions.CartInteractionName.AddToCart	Add To Cart (カートに追加)
SalesforceInteractions.CartInteractionName.RemoveFromCart	Remove From Cart (カートから削除)

```
{
  interaction: {
    name: "Add To Cart",
    lineItem: {
      catalogObjectType: "Product",
      catalogObjectId: "product-1",
      quantity: 1,
      price: 9.99,
      currency: "USD",
      attributes: {
        giftWrapping: true
      }
    }
  }
}
```

单一品目カートインタラクションの項目。

項目名	項目の型	説明
lineItem	品目データ (ページ 18)	必須。1つの品目データ値。
name	string	必須。イベント名。

複数品目

カート内のすべての品目に影響するカートインタラクションで使用されるインタラクション名。

インタラクション名	値
SalesforceInteractions.CartInteractionName.ReplaceCart	Replace Cart (カートを置き換える)

```
{
  interaction: {
    name: "Replace Cart",
```

```

        lineItems: [
            {
                catalogObjectType: "Product",
                catalogObjectId: "product-1",
                quantity: 1,
                price: 9.99,
                currency: "USD",
                attributes: {
                    giftWrapping: true
                }
            },
            {
                catalogObjectType: "Product",
                catalogObjectId: "product-2",
                quantity: 3,
                price: 20,
                currency: "USD",
                attributes: {
                    giftWrapping: false
                }
            }
        ]
    }
}

```

複数品目カートインタラクションの項目。

項目名	項目の型	説明
lineItems	array	必須。ゼロ対多の品目データ (ページ 18) 値の配列
name	string	必須。イベント名。

例: Web SDK を使用してカートインタラクションを取得する例を次に示します。

```

SalesforceInteractions.sendEvent({
    interaction: {
        name: "Add To Cart",
        lineItem: {
            catalogObjectType: "Product",
            catalogObjectId: "product-1",
            quantity: 1,
            price: 9.99,
            currency: "USD",
            attributes: {
                giftWrapping: true
            }
        }
    }
})

```

サイトマップを使用してカートインタラクションを取得する例を次に示します。

```

SalesforceInteractions.init().then(() => {
    const sitemapConfig = {

```

```

    global: {},
    pageTypes: [
        {
            name: 'product_detail_page',
            isMatch: () => true,
            listeners: [
                SalesforceInteractions.listener("click", ".add-to-cart-button", () =>
{
                SalesforceInteractions.sendEvent({
                    interaction: {
                        name: SalesforceInteractions.CartInteractionName.AddToCart,
                        lineItem: {
                            catalogObjectType: "Product",
                            catalogObjectId: "product-1",
                            quantity: 1,
                            price: 9.99,
                            currency: "USD",
                            attributes: {
                                giftWrapping: true
                            }
                        }
                    }
                })
            ]
        }
    ]
}
SalesforceInteractions.initSitemap(sitemapConfig)
})

```

カタログインタラクション

カタログインタラクションを使用して、顧客がさまざまな追跡項目を操作したことを取ります。たとえば、商品やブログ投稿への操作です。

次のインタラクション名はカタログインタラクションとして処理されます。

インタラクション名	値
SalesforceInteractions.CatalogObjectInteractionName.CommentCatalogObject	Comment Catalog Object(カタログオブジェクトにコメント)
SalesforceInteractions.CatalogObjectInteractionName.FavoriteCatalogObject	Favorite Catalog Object(カタログオブジェクトをお気に入りに登録)
SalesforceInteractions.CatalogObjectInteractionName.QuickViewCatalogObject	Quick View Catalog Object(カタログオブジェクトをクイック参照)
SalesforceInteractions.CatalogObjectInteractionName.ReviewCatalogObject	Review Catalog Object(カタログオブジェクトを確認)
SalesforceInteractions.CatalogObjectInteractionName.ShareCatalogObject	Share Catalog Object(カタログオブジェクトを共有)

インタラクション名	値
SalesforceInteractions.CatalogObjectInteractionName.ViewCatalogObject	View Catalog Object (カタログオブジェクトを参照)
SalesforceInteractions.CatalogObjectInteractionName.ViewCatalogObjectDetail	View Catalog Object Detail (カタログオブジェクトの詳細を参照)

```
{
  interaction: {
    name: 'View Catalog Object',
    catalogObject: {
      type: 'Product',
      id: 'product-xyz',
      attributes: {
        name: 'Product XYZ',
        category: 'Clothing',
        color: 'Red',
        sku: 'product-xyz-sku'
      },
      relatedCatalogObjects: [
        {
          Sizes: ["S", "M", "L"]
        }
      ]
    }
  }
}
```

カタログインタラクション項目:

項目名	項目の型	説明
catalogObject.attributes	object	ユーザ指定の値の辞書。
catalogObject.id	string	必須。カタログオブジェクトを表す一意の ID。
catalogObject.relatedCatalogObjects	object	関連するカタログオブジェクトの辞書。
catalogObject.type	string	必須。カタログオブジェクトを表す種別名。
name	string	必須。イベント名。

 **例:** Web SDK を使用してカタログインタラクションを取得する方法を次に示します。

```
SalesforceInteractions.sendEvent({
  interaction: {
    name: 'View Catalog Object',
    catalogObject: {
      type: 'Product',
      id: 'product-xyz',
    }
  }
})
```

```

        attributes: {
            name: 'Product XYZ',
            category: 'Clothing',
            color: 'Red',
            sku: 'product-xyz-sku'
        },
        relatedCatalogObjects: [
            Sizes: ["S", "M", "L"]
        ]
    }
}
})

```

サイトマップを使用してカタログインタラクションを取得する方法を次に示します。

```

SalesforceInteractions.init().then(() => {
    const sitemapConfig = {
        global: {},
        pageTypes: [
            {
                name: 'product_detail',
                isMatch: () => true,
                interaction: {
                    name: SalesforceInteractions.CatalogObjectName.ViewCatalogObject,
                    catalogObject: {
                        type: 'Product',
                        id: 'product-xyz',
                        attributes: {
                            name: 'Product XYZ',
                            category: 'Clothing',
                            color: 'Red',
                            sku: 'product-xyz-sku'
                        },
                        relatedCatalogObjects: [
                            Sizes: ["S", "M", "L"]
                        ]
                    }
                }
            }
        ]
    }
    SalesforceInteractions.initSitemap(sitemapConfig)
})

```

注文インタラクション

注文インタラクションは、新規注文、進行中の注文、または完了した注文を管理するためにユーザが実行したアクションを取得するエコマースイベントです。

次のインタラクション名は注文インタラクションとして処理されます。

名前	値
SalesforceInteractions.OrderInteractionName.Cancel	Cancel

名前	値
SalesforceInteractions.OrderInteractionName.Deliver	Deliver
SalesforceInteractions.OrderInteractionName.Exchange	Exchange
SalesforceInteractions.OrderInteractionName.Preorder	Preorder
SalesforceInteractions.OrderInteractionName.Purchase	Purchase
SalesforceInteractions.OrderInteractionName.Ship	Ship
SalesforceInteractions.OrderInteractionName.Return	Return

```
{
  interaction: {
    name: 'Purchase',
    order: {
      orderId: '9432',
      totalValue: 9.99,
      currency: "USD",
      lineItems: [
        {
          catalogObjectType: 'Product',
          catalogObjectId: 'product-xyz',
          quantity: 1,
          price: 9.99,
          attributes: {
            giftWrapping: true
          }
        }
      ],
      attributes: {
        promoCode: "SAVE10"
      }
    }
  }
}
```

注文インタラクション項目:

項目名	項目の型	説明
name	string	必須。 イベント名。
order.attributes	object	ユーザ指定の値。
order.currency	string	注文の通貨。
order.lineItems	品目データ (ページ 18)	品目データ値の配列。
order.orderId	string	必須。 注文を表す一意の識別子。
order.totalValue	number	必須。 注文の合計金額。

 **例:** Web SDKを使用して注文インタラクションを取得する方法を次に示します。

```
SalesforceInteractions.sendEvent({
  interaction: {
    name: 'Purchase',
    order: {
      orderId: '9432',
      totalValue: 9.99,
      currency: "USD",
      lineItems: [
        {
          catalogObjectType: 'Product',
          catalogObjectId: 'product-xyz',
          quantity: 1,
          price: 9.99,
          attributes: {
            giftWrapping: true
          }
        }
      ],
      attributes: {
        promoCode: "SAVE10"
      }
    }
  }
})
```

サイトマップを使用して注文インタラクションを取得する方法を次に示します。

```
SalesforceInteractions.init().then(() => {
  const sitemapConfig = {
    global: {},
    pageTypes: [
      {
        name: 'product_detail',
        isMatch: () => true,
        interaction: {
          name: SalesforceInteractions.OrderInteractionName.Purchase,
          order: {
            orderId: '9432',
            totalValue: 9.99,
            currency: "USD",
            lineItems: [
              {
                catalogObjectType: 'Product',
                catalogObjectId: 'product-xyz',
                quantity: 1,
                price: 9.99,
                attributes: {
                  giftWrapping: true
                }
              }
            ],
            attributes: {
              promoCode: "SAVE10"
            }
          }
        }
      }
    ]
  }
})
```

```

    SalesforceInteractions.initSitemap(sitemapConfig)
})

```

品目データ

品目は、購入可能な品目を表すことを目的とし、カートおよび注文インタラクションで使用されます。

品目項目:

項目名	項目の型	説明
attributes	object	ユーザ指定の値。
catalogObjectId	string	必須。品目内で参照されるカタログオブジェクトを表す一意の識別子。
catalogObjectType	string	必須。品目内で参照されるカタログオブジェクトを表す名前。
currency	string	価格項目の通貨。
価格	number	品目内で参照されるカタログオブジェクトの価格。
quantity	number	必須。この品目内のカタログオブジェクトの数。

⌚ **例:** サイトマップを使用するインタラクション内で使用される品目の基本構造を次に示します。

```

{
  catalogObjectType: "Product",
  catalogObjectId: "product-1",
  quantity: 1,
  price: 9.99,
  currency: "USD",
  attributes: {
    giftWrapping: true
  }
}

```

ユーザデータ

Salesforce Interactions Web SDKを使用して、インタラクションを関連付けてまたは関連付けずにユーザのプロファイルデータを追跡します。

```

{
  user: {
    anonymousId: '<Set by SDK>',

```

```

        identities: {
            loyaltyId: "885627312393"
        },
        attributes: {
            firstName: 'Joe',
            lastName: 'Smith',
            email: 'joe.smith@domain.com'
        }
    }
}

```

ユーザオブジェクト項目:

項目名	項目の型	説明
anonymousId	string	この項目はSDKによって管理され、すべてのイベントに自動的に付加されます。
user.attributes	object	ユーザ指定の値。
user.identities	object	匿名ユーザを既知のユーザにリンクするためのユーザ指定の値。

例: Web SDKを使用して、インタラクションを関連付けずにユーザプロファイルデータを取得する方法を次に示します。

```

SalesforceInteractions.sendEvent({
    user: {
        identities: {
            loyaltyId: "885627312393"
        },
        attributes: {
            firstName: 'Joe',
            lastName: 'Smith',
            email: 'joe.smith@domain.com'
        }
    }
})

```

Web SDKを使用して、インタラクションを関連付けてユーザプロファイルデータを取得する方法を次に示します。

```

SalesforceInteractions.sendEvent({
    interaction: {
        name: 'View Catalog Object',
        catalogObject: {
            type: 'Product',
            id: 'product-xyz',
            attributes: {
                name: 'Product XYZ',
                category: 'Clothing',

```

```
        color: 'Red'
    }
}
},
user: {
    identities: {
        loyaltyId: "885627312393"
    },
    attributes: {
        firstName: 'Joe',
        lastName: 'Smith',
        email: 'joe.smith@domain.com'
    }
}
})
```

サイトマップを使用してユーザプロファイルデータを取得する方法を次に示します。

```
SalesforceInteractions.init().then(() => {
    const sitemapConfig = {
        global: {
            onActionEvent: (event) => {
                const email = window.user_info && window.user_info.email;
                if (email) {
                    event.user = event.user || {};
                    event.user.attributes = event.user.attributes || {};
                    event.user.attributes.emailAddress = email;
                }
                return event;
            },
        },
        pageTypes: [
            {
                name: 'product_detail',
                isMatch: () => true,
                interaction: {
                    name: SalesforceInteractions.InteractionName.ViewCatalogObject,
                    catalogObject: {
                        type: 'Product',
                        id: 'product-xyz',
                        attributes: {
                            name: 'Product XYZ',
                            category: 'Clothing',
                            color: 'Red'
                        }
                    }
                }
            }
        ]
    }
    SalesforceInteractions.initSitemap(sitemapConfig)
})
```

アカウントデータ

アカウントデータはアカウントレベルのユーザ情報を表します。アカウント項目を使用すると、1つのアカウントで複数の顧客のデータを取得できます。

```
{
  account: {
    id: '43296241300',
    attributes: {
      role: 'Subscriber'
    }
  }
}
```

ユーザオブジェクト項目:

項目名	項目の型	説明
account.attributes	object	ユーザ指定の値。
account.id	string	必須。ユーザのアカウント識別子。

例: Web SDK を使用して、インタラクションを関連付けずにアカウントデータを取得する方法を次に示します。

```
SalesforceInteractions.sendEvent({
  account: {
    id: '43296241300',
    attributes: {
      role: 'Subscriber'
    }
  }
})
```

Web SDK を使用して、インタラクションを関連付けてアカウントデータを取得する方法を次に示します。

```
SalesforceInteractions.sendEvent({
  interaction: {
    name: 'View Catalog Object',
    catalogObject: {
      type: 'Product',
      id: 'product-xyz',
      attributes: {
        name: 'Product XYZ',
        category: 'Clothing',
        color: 'Red'
      }
    }
  },
  account: {
    id: '43296241300',
    attributes: {
      role: 'Subscriber'
    }
  }
})
```

}

サイトマップを使用してアカウントデータを取得する方法を次に示します。

```
SalesforceInteractions.init().then(() => {
  const sitemapConfig = {
    global: {
      onActionEvent: (event) => {
        const accountId = window.accountId;
        if (accountId) {
          event.account = event.user || {};
          event.account.id = accountId;
        }
        return event;
      },
    },
    pageTypes: [
      {
        name: 'product_detail',
        isMatch: () => true,
        interaction: {
          name: SalesforceInteractions.InteractionName.ViewCatalogObject,
          catalogObject: {
            type: 'Product',
            id: 'product-xyz',
            attributes: {
              name: 'Product XYZ',
              category: 'Clothing',
              color: 'Red'
            }
          }
        }
      }
    ]
  }
}

SalesforceInteractions.initSitemap(sitemapConfig)
})
```

同意データ

同意データは、Salesforce Interactions SDK での同意の付与または取り消しに使用されます。顧客が追跡へのオプトインを選択した場合、この同意の詳細が、最初に追跡されたイベントのイベント構造に付加されます。

同意(ページ22)オブジェクト項目:

項目	型	説明
provider	string	必須。SDKに同意を提供する同意管理システムを示す指定識別子。
purpose	string	必須。顧客が何に同意したか。例: Tracking (追跡)

項目	型	説明
status	string	必須。値: Opt In (オプトイン) または Opt Out (オプトアウト)

イベントに付加される同意データの基本構造を次に示します。

```
{
  provider: "Example Provider",
  purpose: "Tracking",
  status: "Opt In"
}
```

次の同意の目的名および状況名が提供されています。

名前	値
SalesforceInteractions.ConsentPurpose.Tracking	Tracking (追跡)
SalesforceInteractions.ConsentStatus.OptIn	Opt In (オプトイン)
SalesforceInteractions.ConsentStatus.OptOut	Opt Out (オプトアウト)

API リファレンス

このセクションでは、Salesforce Interactions Web SDK で提供されているリファレンスに関するガイダンスを提供します。

初期化

Salesforce Interactions SDK の推奨初期化フローは一般的なパターンに従います。

データ収集

Salesforce Interactions SDK でデータの取得を開始するには、Salesforce Interactions SDK を初期化し、追跡への同意が付与される必要があります。

ID

Salesforce Interactions SDK では、ファーストパーティ Cookie `_sfid_{domainHash}` を使用して、新規顧客とリピート顧客を識別することができます。また、ハッシュ関数を使用して Web サイトのドメインを実行することで、`domainHash` が生成されます。

Consent

Salesforce Interactions SDK はユーザのプライバシーを尊重するように設計されているため、同意が付与されるまで、収集したデータは保存または転送されません。サイトに既存の同意管理機能がある場合は、その機能を SDK に統合し、ユーザが追跡に同意していることを通知する必要があります。

サイトマップ

Salesforce Interactions SDK サイトマップでは、ページナビゲーション中にデータを抽出することができます。また、複数のページでデータキャプチャロジックを共有したり、データキャプチャロジックを Web ページ プレゼンテーションロジックから切り離したりできます。

インテグレーション

Salesforce Interaction SDK には、Web SDK に固有のイベントにリスナーをバインドするために使用できる組み込み CustomEvents があります。このイベントにより渡されたデータは、リスナーのコールバック関数に渡されるイベントオブジェクトの詳細プロパティ内にあります。

デバッグ

設定すると、Salesforce Interaction SDK でデバッグ目的でメッセージをブラウザコンソールに出力できます。デフォルトでは、ロガーは `none` のログレベルで設定されます。Web SDK で使用するロガーでは、標準ブラウザの `console.log` 関数がラップされます。このロガーには `SalesforceInteractions.log` でアクセスできます。

初期化

Salesforce Interactions SDK の推奨初期化フローは一般的なパターンに従います。

```
SalesforceInteractions.init({
  consents: [...],
  cookieDomain: '...'
}).then(() => {
  SalesforceInteractions.initSitemap({
    global: { ... },
    pageTypeDefault: { ... },
    pageTypes: [...]
  })
})
```

このメソッドでは、同意、ID、およびサイトマップ機能を有効にした Web SDK を初期化します。

初期化メソッド

```
init(sdkConfig: SdkConfig): Promise<void>
```

引数:

- `sdkConfig` - consent および `cookieDomain` 項目を持つオブジェクト。

項目名	項目の型	説明
<code>consents</code>	<code>Consent[]</code> <code>Promise<Consent[]></code>	必須。同意データ値のリスト、または同意データ値のリストを解決する <code>Promise</code> 。
<code>cookieDomain</code>	<code>string</code>	IDデータを保存するファーストパーティ Cookie で使用するドメイン。デ

項目名	項目の型	説明
		フォルトでは現在のサイトのドメインになります。

```
// initialize with consent provided by a user interaction
SalesforceInteractions.init({
  cookieDomain: 'domain.com',
  consents: new Promise(resolve => {
    // user clicks button that grants consent
    document.getElementById('opt-in')
      .addEventListener('click', () => {
        resolve([
          provider: 'Test Provider',
          purpose: 'Tracking',
          status: SalesforceInteractions.ConsentStatus.OptIn
        ])
      }, { once: true })
  })
})

// user clicks button that revokes consent
document.getElementById('opt-out')
  .addEventListener('click', () => {
    resolve([
      provider: 'Test Provider',
      purpose: 'Tracking',
      status: SalesforceInteractions.ConsentStatus.OptOut
    ])
  }, { once: true })
})

// initialize with consent preconfigured
SalesforceInteractions.init({
  cookieDomain: 'domain.com',
  consents: [
    provider: 'Test Provider',
    purpose: 'Tracking',
    status: SalesforceInteractions.ConsentStatus.OptIn
  ]
})

// initialize with no consent
SalesforceInteractions.init({
  cookieDomain: 'domain.com',
  consents: []
})

reinit(): void
```

Web SDK を再初期化し、設定済みのサイトマップから適用するルールを再評価します。

```
SalesforceInteractions.init({...}).then(() => {
  let href = window.location.href
```

```
// monitor the url for changes and reinitialize when detected
setInterval(() => {
  if (href !== window.location.href) {
    href = window.location.href
    SalesforceInteractions.reinit()
  }
}, 200)

SalesforceInteractions.initSitemap({...})
})
```

データ収集

Salesforce Interactions SDK でデータの取得を開始するには、Salesforce Interactions SDK を初期化し、追跡への同意が付与される必要があります。

[初期化](#) (ページ 24) と [同意](#) (ページ 29) は次の方法で実行できます。

```
SalesforceInteractions.init({
  consents: [
    {
      purpose: "Tracking",
      provider: "OneTrust",
      status: "Opt In"
    }
  ]
})
```

サイトに既存の同意管理機能がある場合は、その機能を Salesforce Interactions SDK に統合し、ユーザが追跡に同意していることを通知します。

ユーザエンゲージメントデータの取得

顧客インタラクションを取得するには、`sendEvent` 関数をコールします。

```
SalesforceInteractions.sendEvent({
  interaction : {
    name : "View Catalog Object",
    catalogObject: {
      type : "Product",
      id : "65e4e737",
      attributes: {
        description: "Shoes"
      }
    }
  }
})
```

プロファイルデータの取得

プロファイルデータも同じ `sendEvent` 関数コールで取得されます。

```
SalesforceInteractions.sendEvent({
  user: {
    attributes: {
      email: 'user@domain.com'
    }
  }
})
```

プロファイルデータとエンゲージメントデータの取得

プロファイルデータを取得する場合、ユーザはその最小限のインタラクション形式に制限されません。いずれかのインタラクションデータモデルをプロファイル情報と組み合わせて、イベントのコンテキストを可能な限り収集することができます。

```
SalesforceInteractions.sendEvent({
  interaction : {
    name : "View Catalog Object",
    catalogObject: {
      type : "Product",
      id : "65e4e737",
      attributes: {
        description: "Shoes"
      }
    }
  },
  user: {
    attributes: {
      email: 'user@domain.com'
    }
  }
})
```

ID

Salesforce Interactions SDK では、ファーストパーティ Cookie `_sfid_${domainHash}` を使用して、新規顧客とリピート顧客を識別することができます。また、ハッシュ関数を使用して Web サイトのドメインを実行することで、`domainHash` が生成されます。

- 新規顧客は、ランダムに生成された `anonymousId` を受け取ります。これはファーストパーティ Cookie に保持されます。
- リピート顧客は、ファーストパーティ Cookie にすでに保存されている `anonymousId` を使用します。

複数ドメインでの Cookie の共有

複数のドメインで Cookie を共有する必要がある場合、共有ドメイン名を使用して Salesforce Interactions Web SDK を設定します。これは、`SalesforceInteractions.init` に渡す設定を介して設定するか、`SalesforceInteractions.setCookieDomain` への関数コールで設定できます。

たとえば、次の 2 つのドメインで Cookie を共有するとします。

- `abc.domain.com`
- `xyz.domain.com`

Cookie ドメインは次のように設定できます。

```
SalesforceInteractions.init({
  cookieDomain: 'domain.com'
})
```

Cookie ドメインは、`setCookieDomain` 関数を使用して設定することもできます。

ID メソッド

```
getCookieDomain(): string
```

ファーストパーティ Cookie の生成に使用する Cookie ドメインの値を返します。ファーストパーティ Cookie には匿名顧客 ID が保存されます。

```
SalesforceInteractions.getCookieDomain()
  // => "domain.com"
```

```
setCookieDomain(newDomain:string): void
```

ファーストパーティ Cookie の生成に使用する Cookie ドメインの値を設定します。ファーストパーティ Cookie には匿名顧客 ID が保存されます。

```
SalesforceInteractions.setCookieDomain('domain.com')
```

 **メモ:** Salesforce Web Interactions SDK がすでに初期化されている場合、`setCookieDomain` 関数を介した Cookie の設定は有効になりません。この関数コールを初期化の前に実行することをお勧めしますが、`SalesforceInteractions.reinit()` をコールして Cookie の変更を強制的に有効にすることもできます。

```
getAnonymousId():string
```

ファーストパーティ Cookie に保持される匿名顧客 ID の値を返します。

```
SalesforceInteractions.getAnonymousId()
  // => "efc9953d6515dc7f"
```

```
setAnonymousId(newAnonymousId:string): void
```

ファーストパーティ Cookie に保持される匿名顧客 ID の値を設定します。

```
SalesforceInteractions.setAnonymousId(`id_${Date.now()}`)
```

```
resetAnonymousId():void
```

ファーストパーティ Cookie を削除し、新しく生成された匿名顧客 ID を使用して新しい Cookie を生成します。

```
SalesforceInteractions.resetAnonymousId()
```

Consent

Salesforce Interactions SDK はユーザのプライバシーを尊重するように設計されているため、同意が付与されるまで、収集したデータは保存または転送されません。サイトに既存の同意管理機能がある場合は、その機能をSDKに統合し、ユーザが追跡に同意していることを通知する必要があります。

ユーザの同意は、次の場合に Salesforce Interactions SDK に提供される可能性があります。

- [初期化](#) (ページ 24) 中
- updateConsents = (consents: Consent | Consent[]): void のコール中

次の例のように、初期化中に同意への Promise を提供し、同意データ値の配列で解決することをお勧めします。

```
SalesforceInteractions.init({
  consents: new Promise(resolve => {
    // user clicks button that grants consent
    document.getElementById('opt-in')
      .addEventListener('click', () => {
        resolve([
          {
            provider: 'Test Provider',
            purpose: SalesforceInteractions.ConsentPurpose.Tracking,
            status: SalesforceInteractions.ConsentStatus.OptIn
          }
        ])
      }, { once: true })
  }

  // user clicks button that revokes consent
  document.getElementById('opt-out')
    .addEventListener('click', () => {
      resolve([
        {
          provider: 'Test Provider',
          purpose: SalesforceInteractions.ConsentPurpose.Tracking,
          status: SalesforceInteractions.ConsentStatus.OptOut
        }
      ])
    }, { once: true })
  })
})
```

同意メソッド

```
getConsents: ConsentWithMetadata[]
```

最終更新時間および最終送信時間を含む [同意データ](#) (ページ 22) を返します。

```
updateConsents = (consents: Consent | Consent[]): void
```

初期化後に同意を更新する方法、たとえば、OneTrust コードから直接コールする方法を次に示します。

```
// when the user opts in
SalesforceInteractions.updateConsents({
  purpose: SalesforceInteractions.ConsentPurpose.Tracking,
```

```

    provider: "OneTrust",
    status: SalesforceInteractions.ConsentStatus.OptIn
  })

// when the user opts out
SalesforceInteractions.updateConsents({
  purpose: SalesforceInteractions.ConsentPurpose.Tracking,
  provider: "OneTrust", status:
  SalesforceInteractions.ConsentStatus.OptOut
})

```

サイトマップ

Salesforce Interactions SDK サイトマップでは、ページナビゲーション中にデータを抽出することができます。また、複数のページでデータキャプチャロジックを共有したり、データキャプチャロジックを Web ページプレゼンテーションロジックから切り離したりできます。

サイトマップは、Web SDK の初期化直後に設定することができます。一般的なパターンを次に示します。

```

SalesforceInteractions.init().then(() => {
  SalesforceInteractions.initSitemap({
    global: { ... },
    pageTypeDefault: { ... },
    pageTypes: [...]
  })
})

```

サイトマップの各セクションで次を定義できます。

- 複数の `listener`。ページで実行された特定の顧客イベントを取得します。
- `onActionEvent` フック。これを使用してデータを傍受し、取得したデータを変更できます。
- `name`。ページ設定を識別します。
- ページで取得する `locale`。
- `interaction`。ページの読み込み後に取得します。

GlobalPageConfig

`global` 設定は、対象の Web ページに適用される `PageConfig` または `DefaultPageConfig` にマージされます。対象の Web ページに一致する設定がない場合、`global` 設定は適用されません。

項目名	項目の型	説明
<code>listeners</code>	<code>listener</code>	<code>SalesforceInteractions.listener</code> 関数を使用して作成されたイベントリスナーの配列。このイベントは、Web SDK を介したイベントの送信をトリガする顧客インタラクションをリスンします。

項目名	項目の型	説明
locale	string	ISO 639 Alpha-2 言語コードと ISO 3166 Alpha-2 国コードで表される現在のページのロケール。例:en_US、de_DE
onActionEvent	(event: ActionEvent) => ActionEvent	ActionEvent オブジェクトが Web SDK に渡されるときに実行されるコールバック。このコールでは、送信前にイベントのプロパティを挿入または変更するためのフックが提供されます。

```

SalesforceInteractions.init().then(() => {
  const {
    listener,
    CatalogObjectInteractionName,
    resolvers
  } = SalesforceInteractions

  const global = {
    locale: 'en_US',
    listeners: [
      listener('click', '[title="Log In"]', (event) => {
        console.log(event)
      })
    ],
    onActionEvent: (actionEvent) => {
      console.log(actionEvent)
    }
  }

  const pageTypeDefault = {...}

  const pageTypes = [...]

  SalesforceInteractions.initSitemap({
    global,
    pageTypes,
    pageTypeDefault
  })
})

```

DefaultPageConfig

デフォルトページ設定は、登録された PageConfig に一致するものが対象の Web ページで見つからない場合に有効化されます。

項目名	項目の型	説明
listeners	listener	<i>SalesforceInteractions.listener</i> 関数を使用して作成されたイベントリスナーの配列。このイベントは、Web SDKを介したイベントの送信をトリガする顧客インターラクションをリスンします。
locale	string	ISO 639 Alpha-2 言語コードと ISO 3166 Alpha-2 国コードで表される現在のページのロケール。例:en_US、de_DE
name	string	デフォルトページ設定を識別する名前。
onActionEvent	(event:ActionEvent) => ActionEvent	ActionEvent オブジェクトが Web SDK に渡されるときに実行されるコールバック。このコールでは、送信前にイベントのプロパティを挿入または変更するためのフックが提供されます。

```

SalesforceInteractions.init().then(() => {
  const {
    listener,
    CatalogObjectInteractionName,
    resolvers
  } = SalesforceInteractions

  const global = {...}

  const pageTypeDefault = {
    name: 'default',
    locale: 'en_US',
    listeners: [
      listener('click', '.example-selector', (event) => {
        console.log(event)
      })
    ],
    onActionEvent: (actionEvent) => {
      console.log(actionEvent)
    }
  }

  const pageTypes = [...]

  SalesforceInteractions.initSitemap({
    global,
    pageTypes,
  })
}

```

```

    pageTypeDefault
  })
})

```

PageConfig

項目名	項目の型	説明
interaction	カートインタラクション (ページ 10) カタログインタラクション (ページ 13) 注文インタラクション (ページ 15)	ページから取得するインタラクション。ページ設定が一致すると、この項目が自動的に抽出されて送信されます。サイトマップの設定時に、抽出するデータが完全にレンダリングされていないことがあるため、静的値または関数でラップされた値を指定して、値を後で評価するようにします。
isMatch	() => Boolean	必須。特定のページ設定を適用するかどうかをテストする関数。
listeners	listener	<i>SalesforceInteractions.listener</i> 関数を使用して作成されたイベントリスナーの配列。このイベントは、Web SDKを介したイベントの送信をトリガする顧客インタラクションをリスンします。
locale	string	ISO 639 Alpha-2 言語コードと ISO 3166 Alpha-2 国コードで表される現在のページのロケール。例:en_US、de_DE
name	string	必須。ページ設定を識別する一意の名前。
onActionEvent	(event:ActionEvent) => ActionEvent	ActionEvent オブジェクトが Web SDK に渡されるときに実行されるコールバック。このコールでは、送信前にイベントのプロパティを挿入または変更するためのフックが提供されます。

```

SalesforceInteractions.init().then(() => {
  const {
    listener,
    CatalogObjectInteractionName,
    resolvers
  }
})

```

```
} = SalesforceInteractions

const global = {...}

const pageTypeDefault = {...}

const HomePage = {
  name: 'home',
  locale: 'en_US',
  isMatch: () => {
    return window.location.pathname === '/'
  },
  listeners: [
    listener('click', '#promo', (event) => {
      console.log(event)
    })
  ],
  onActionEvent: (actionEvent) => {
    console.log(actionEvent)
  }
}

const productPage = {
  name: 'product',
  isMatch: () => {
    return /product\/\d+/.test(window.location.pathname),
  },
  interaction: {
    name: CatalogObjectInteractionName.ViewCatalogObject,
    catalogObject: {
      type: 'Product',
      id: resolvers.fromSelectorAttribute('.product', 'data-id')
    }
  }
}

const pageTypes = [HomePage, productPage]

SalesforceInteractions.initSitemap({
  global,
  pageTypes,
  pageTypeDefault
})
})
```

サイトマップメソッド

```
initSitemap(siteMapConfig: SiteMapConfig): boolean
```

項目名	型	説明
global	GlobalPageConfig	必須。global 設定は、対象の Web ページに適用される PageConfig または DefaultPageConfig にマージされます。対象の Web ページに一致する設定がない場合、global 設定は適用されません。
pageTypes	PageConfig[]	必須。指定された、対象の Web ページの条件に一致するページでデータ収集ルールを実行するかどうかを指定するページ設定の配列。複数のページ設定が1つのページに一致する可能性があります。これらの設定は GlobalPageConfig (提供されている場合) と共にまとめてマージされます。
pageTypeDefault	DefaultPageConfig	デフォルトページ設定は、登録された PageConfig に一致するものが対象の Web ページで見つからない場合に有効化されます。

HTML ドキュメントからの正規リンクの抽出

```
resolvers.fromCanonical(
  transform?: (value: string | undefined) => string | undefined
): () => string | undefined
```

HTML ドキュメントから正規リンクを抽出するリゾルバを作成します。Transform は、解決された値を新しい値に変換する省略可能な関数です。

たとえば、次のような HTML があるとします。

```
<link rel="canonical" href="/some/url">
```

正規 href の値を抽出するには、次のコードを使用できます。

```
const fromCanonical = SalesforceInteractions.resolvers.fromCanonical
const toUpperCase = (value) => value.toUpperCase()

fromCanonical()()
// => "/some/url"

// extract and transform the value
fromCanonical(toUpperCase)()
// => "/SOME/URL"
```

HTML ドキュメントの `Location` オブジェクトからの `Href` 値の抽出

```
resolvers.fromHref(
  transform?: (value: string | undefined) => string |
  undefined
): () => string | undefined
```

HTML ドキュメントの `Location` オブジェクトから `href` 値を抽出するリゾルバを作成します。Transform は、解決された値を新しい値に変換する省略可能な関数です。

たとえば、次の URL に配置されている HTML ドキュメントがあるとします。

```
https://site.domain.com/search.html?page=3
```

`href` を抽出するには、次のコードを使用できます。

```
const fromHref = SalesforceInteractions.resolvers.fromHref
const toUpperCase = (value) => value.toUpperCase()

fromHref()()
// => "https://site.domain.com/search.html?page=3"

// extract and transform the value
fromHref(toUpperCase)()
// => "HTTPS://SITE.DOMAIN.COM/SEARCH.HTML?PAGE=3"
```

最初の要素からの `content` 属性の抽出

```
resolvers.fromItemProp(
  itemProp: string,
  transform?: (value: string | undefined) => string | undefined
): () => string | undefined
```

HTML ドキュメント内で対象の `itemprop` 属性値に一致する最初の要素から `content` 属性を抽出するリゾルバを作成します。

引数:

- `itemProp` - 照合する `itemprop` 属性の値
- `transform` - 解決された値を新しい値に変換する省略可能な関数

たとえば、次のような HTML があるとします。

```
<ul>
  <li itemprop="name" content="first">First</li>
  <li itemprop="name" content="second">Second</li>
</ul>
```

`<meta>` 要素から値を抽出するには、次のコードを使用できます。

```
const fromItemProp = SalesforceInteractions.resolvers.fromItemProp
const toUpperCase = (value) => value.toUpperCase()

fromItemProp("name")()
// => "first"

// extract and transform the value
```

```
fromItemProp("name", toUpperCase) ()  
// => "FIRST"
```

JSON Linked Data の抽出

```
resolvers.fromJsonLd(  
  path?: string,  
  transform?: (value: string | undefined) => string | undefined  
) : () => string | undefined
```

HTML ドキュメント内で *application/ld+json* の type 属性を持つ最初の <script> 要素から JSON Linked Data (JSON-LD) を抽出するリゾルバを作成します。

引数:

- path - 取得するプロパティのパス
- transform - 解決された値を新しい値に変換する省略可能な関数

たとえば、次のような HTML があるとします。

```
<script type="application/ld+json">  
{  
  "@context": "https://json-ld.org/contexts/person.jsonld",  
  "@id": "http://dbpedia.org/resource/John_Smith",  
  "name": "John Smith",  
  "born": "1940-10-09",  
  "spouse": "http://dbpedia.org/resource/Cynthia_Smith",  
  "gr:includes": {  
    "@type": [  
      "gr:Individual"  
    ]  
  }  
}</script>
```

<meta> 要素から値を抽出するには、次のコードを使用できます。

```
const fromJsonLD = SalesforceInteractions.resolvers.fromJsonLD  
const toUpperCase = (value) => value.toUpperCase()  
  
fromJsonLD() ()  
// => {@context: "https://json-ld.org/contexts/person.jsonld", @id: ...}  
  
// extract a property  
fromJsonLD("@id") ()  
// => "http://dbpedia.org/resource/John_Smith"  
  
// extract a nested property  
fromJsonLD("gr:includes.@type") ()  
// => ["gr:Individual"]  
  
// extract and transform the value  
fromJsonLD("name", toUpperCase) ()  
// => "JOHN SMITH"
```

最初のメタ要素からのコンテンツ属性の抽出

```
resolvers.fromMeta(
  name: string,
  transform?: (value: string | undefined) => string | undefined
): () => string | undefined
```

HTML ドキュメント内で特定の name 属性に一致する最初の <meta> 要素から content 属性を抽出するリゾルバを作成します。

引数:

- name - 照合する name 属性の値
- transform - 解決された値を新しい値に変換する省略可能な関数

たとえば、次のような HTML があるとします。

```
<meta name="id" content="product-1">
```

<meta> 要素から値を抽出するには、次のコードを使用できます。

```
const fromMeta = SalesforceInteractions.resolvers.fromMeta
const toUpperCase = (value) => value.toUpperCase()

fromMeta("id")()
// => "product-1"

// extract and transform the value
fromMeta("id", toUpperCase)()
// => "PRODUCT-1"
```

最初の要素からのテキストの抽出

```
resolvers.fromSelector(
  selector: string,
  transform?: (value: string | undefined) => string | undefined
): () => string | undefined
```

HTML ドキュメント内で特定の CSS セレクタに一致する最初の要素からテキストを抽出するリゾルバを作成します。

引数:

- selector - 照合する対象の CSS セレクタ
- transform - 解決された値を新しい値に変換する省略可能な関数

たとえば、次のような HTML があるとします。

```
<div id="product-1">
  <h1 class="name">Product XYZ</h1>
</div>
```

各 <h1> 要素のテキストを抽出するには、次のコードを使用できます。

```
const fromSelector = SalesforceInteractions.resolvers.fromSelector
const toUpperCase = (value) => value.toUpperCase()

fromSelector("#product-1 .name")()
```

```
// => "Product XYZ"

// extract and transform the value
fromSelector("#product-1 .name", toUpperCase)()
// => "PRODUCT XYZ"
```

最初の要素からのテキストの抽出

```
resolvers.fromSelectorMultiple(
  selector: string,
  transform?: (value: string[] | undefined) => string[] | undefined
): () => string[] | undefined
```

HTML ドキュメント内で特定の CSS セレクタに一致する各要素からテキストを抽出するリゾルバを作成します。

引数:

- selector - 照合する対象の CSS セレクタ
- transform - 解決された値を新しい値に変換する省略可能な関数

たとえば、次のような HTML があるとします。

```
<ul class="products">
  <li class="name">Product 1</li>
  <li class="name">Product 2</li>
  <li class="name">Product 3</li>
</ul>
```

各 `<h1>` 要素のテキストを抽出するには、次のコードを使用できます。

```
const fromSelectorMultiple = SalesforceInteractions.resolvers.fromSelectorMultiple
const mapToUpperCase = (values) => values.map(v => v.toUpperCase())

fromSelectorMultiple(".name")()
// => ["Product 1", "Product 2", "Product 3"]

// extract and transform the values
fromSelectorMultiple(".name", mapToUpperCase)
// => ["PRODUCT 1", "PRODUCT 2", "PRODUCT 3"]
```

最初の要素からの対象属性の値の抽出

```
resolvers.fromSelectorAttribute(
  selector: string,
  attribute: string,
  transform?: (value: string | undefined) => string | undefined
): () => string | undefined
```

HTML ドキュメント内で特定の CSS セレクタに一致する最初の要素から対象属性の値を抽出するリゾルバを作成します。

引数:

- selector - 照合する対象の CSS セレクタ
- attribute - 値の取得元である対象属性。
- transform - 解決された値を新しい値に変換する省略可能な関数

たとえば、次のような HTML があるとします。

```
<div id="product">
  <h1 class="name" data-id="product-1">Product 1</h1>
</div>
```

data-id 属性の値を抽出するには、次のコードを使用できます。

```
const fromSelectorAttribute = SalesforceInteractions.resolvers.fromSelectorAttribute
const toUpperCase = (value) => value.toUpperCase()

fromSelectorAttribute("#product .name", "data-id")()
// => "product-1"

// extract and transform the value
fromSelectorAttribute("#product .name", "data-id", toUpperCase)()
// => "PRODUCT-1"
```

対象属性の値の抽出

```
resolvers.fromSelectorAttributeMultiple(
  selector: string,
  attribute: string,
  transform?: (value: string[] | undefined) => string[] | undefined
): () => string[] | undefined
```

HTML ドキュメント内で特定の CSS セレクタに一致する各要素から対象属性の値を抽出するリゾルバを作成します。

引数:

- selector - 照合する対象の CSS セレクタ
- attribute - 値の取得元である対象属性。
- transform - 解決された値を新しい値に変換する省略可能な関数

たとえば、次のような HTML があるとします。

```
<ul class="products">
  <li class="name" data-id="product-1">Product 1</li>
  <li class="name" data-id="product-2">Product 2</li>
  <li class="name" data-id="product-3">Product 3</li>
</ul>
```

各 *data-id* 属性の値を抽出するには、次のコードを使用できます。

```
const fromSelectorAttributeMultiple =
  SalesforceInteractions.resolvers.fromSelectorAttributeMultiple;
const mapToUpperCase = (values) => values.map(v => v.toUpperCase())

fromSelectorAttributeMultiple(".name", "data-id")()
// => ["Product 1", "Product 2", "Product 3"]

// extract and transform the value
fromSelectorAttributeMultiple(".name", "data-id", mapToUpperCase)()
// => ["PRODUCT 1", "PRODUCT 2", "PRODUCT 3"]
```

オブジェクトの値の抽出

```
resolvers.fromWindow(
  path: string,
  transform?: (value: string[] | undefined) => string[] | undefined
): () => string[] | undefined
```

window オブジェクトに付加されているオブジェクト(通常はグローバル変数)の値を抽出するリゾルバを作成します。

引数:

- path - window オブジェクト内のオブジェクトへのパスを表す文字列。ドット表記を使用します。
- transform - 解決された値を新しい値に変換する省略可能な関数

たとえば、次のような window オブジェクトがあるとします。

```
window = {
  shop: {
    name: "SHOP 1",
    products: [
      "PRODUCT 1",
      "PRODUCT 2"
    ]
  },
  ...
}
```

shop グローバルオブジェクトからデータを抽出するには、次のコードを使用できます。

```
const fromWindow = SalesforceInteractions.resolvers.fromWindow;

fromWindow("shop.name")()
// => "SHOP 1"

fromWindow("shop.products[1]")()
// => "PRODUCT 2"
```

サンプル E コマースサイトマップ

このサンプルサイトマップを確認してください。

サンプル E コマースサイトマップ

このサンプルサイトマップを確認してください。

 例:

```
SalesforceInteractions.init({
  consents: new Promise(resolve => {
    const { OptIn, OptOut } = SalesforceInteractions.ConsentStatus
    const purpose = SalesforceInteractions.ConsentPurpose.Tracking
    const provider = 'Test Provider'

    // user clicks button that grants consent
```

```
document.getElementById('opt-in')
  .addEventListener(
    'click',
    () => resolve([{ purpose, provider, status: OptIn }]),
    { once: true }
  )

// user clicks button that revokes consent
document.getElementById('opt-out')
  .addEventListener(
    'click',
    () => resolve([{ purpose, provider, status: OptOut }]),
    { once: true }
  )
})

}).then(() => {
// set the log level during sitemap development to see potential problems
SalesforceInteractions.log.level = 'debug'

const {
  cashDom,
  listener,
  resolvers,
  sendEvent,
  util,
  CartInteractionName,
  CatalogObjectInteractionName,
  OrderInteractionName,
} = SalesforceInteractions

const global = {
  listeners: [
    // capture email address when a user signs up
    listener('submit', '.user-signup-form', (actionEvent) => {
      const emailAddress = cashDom("#user_email").val()
      if (emailAddress) {
        sendEvent({
          interaction: {
            name: 'Email Sign Up'
          },
          user: {
            attributes: { emailAddress }
          }
        })
      }
    })
  ],
}

// attach optional data to every actionEvent that is sent out
onActionEvent: (actionEvent) => {
  const email = window && window._userInfo && window._userInfo.email
  if (email) {
    actionEvent.user = actionEvent.user || {}
    actionEvent.user.attributes = actionEvent.user.attributes || {}
  }
}
```

```
        actionEvent.user.attributes.emailAddress = email
    }
    return actionEvent
}
}

const productIdResolver = resolvers.fromSelectorAttribute('.product', 'data-id')

const productPage = {
  name: 'product',
  isMatch: () => /products/.test(window.location.pathname),
  // capture the product being viewed when the page is opened
  interaction: {
    name: CatalogObjectInteractionName.ViewCatalogObject,
    catalogObject: {
      type: 'Product',
      id: productIdResolver,
      attributes: {
        name: resolvers.fromSelector('.product-title'),
        url: resolvers.fromHref(),
        imageUrl: resolvers.fromSelectorAttribute('.product img', 'src')
      },
      relatedCatalogObjects: {
        Color: resolvers.fromSelectorAttributeMultiple('.color-value',
          'data-attr-value')
      }
    }
  },
  listeners: [
    // capture when the user adds this product to their cart
    listener('click', '.add-to-cart', () => {
      sendEvent({
        interaction: {
          name: CartInteractionName.AddToCart,
          lineItem: {
            catalogObjectType: 'Product',
            catalogObjectId: productIdResolver(),
            quantity: parseInt(cashDom('.product .quantity input').val(), 10),
            price: parseFloat(cashDom('.product .price').text().trim())
          }
        }
      })
    }),
    // capture when the user shares the product to social media
    listener('click', '.share', () => {
      sendEvent({
        interaction: {
          name: CatalogObjectInteractionName.ShareCatalogObject,
          catalogObject: {
            type: 'Product',
            id: productIdResolver()
          }
        }
      })
    })
  ]
}
```

```
        })
    ]
}

const cartPage = {
  name: 'Cart',
  isMatch: () => /^\/cart/.test(window.location.href),
  listeners: [
    // capture when a user removes an item from their cart
    listener('click', '.remove-from-cart', (event) => {
      const $cartItem = cashDom(event.target).parents('.cart-item').first()
      sendEvent({
        interaction: {
          name: CartInteractionName.RemoveFromCart,
          lineItem: {
            catalogObjectType: 'Product',
            catalogObjectId: $cartItem.attr('data-id'),
            quantity: parseInt($cartItem.find('.quantity').text().trim(), 10),
          }
        }
      })
    })
  ]
}

const orderConfirmationPage = {
  name: 'Order Configuration',
  isMatch: /\confirmation/.test(window.location.href),
  // capture when a user completes an order
  interaction: {
    name: OrderInteractionName.Purchase,
    order: {
      id: resolvers.fromSelectorAttribute('.order', 'data-id'),
      totalValue: parseFloat(resolvers.fromSelector('.order .total').trim()),
      lineItems: () => cashDom('.order .line-items').map((index, el) => {
        const $lineItem = cashDom(el)
        return {
          catalogObjectType: 'Product',
          catalogObjectId: $lineItem.attr('data-id'),
          quantity: parseInt($lineItem.find('.quantity').text().trim(), 10)
        }
      })
    }
  }
}

const pageTypeDefault = {
  name: 'default'
}

SalesforceInteractions.initSitemap({
  global,
  pageTypeDefault,
  pageTypes: [cartPage, orderConfirmationPage, productPage]
```

```
    })
})
```

インテグレーション

Salesforce Interaction SDK には、Web SDK に固有のイベントにリスナーをバインドするために使用できる組み込み CustomEvents があります。このイベントにより渡されたデータは、リスナーのコールバック関数に渡されるイベントオブジェクトの詳細プロパティ内にあります。

Salesforce Interaction SDK 組み込みカスタムイベント。

```
SalesforceInteractions.CustomEvents = {
  OnBeforeEventSend: "interactions:onBeforeEventSend",
  OnClearPersistedIdentities: "interactions:onClearPersistedIdentities",
  OnConsentRevoke: "interactions:onConsentRevoke",
  OnEventSend: "interactions:onEventSend",
  OnException: "interactions:onException",
  OnInit: "interactions:onInit",
  OnInitSitemap: "interactions:onInitSitemap",
  OnPageMatchStatusUpdated: "interactions:onPageMatchStatusUpdated",
  OnResetAnonymousId: "interactions:onResetAnonymousId",
  OnSetAnonymousId: "interactions:onSetAnonymousId",
  OnShutDown: "interactions:onShutDown"
}
```

カスタムイベント	説明
OnBeforeEventSend	この イベント (ページ 8) は、 <code>SalesforceInteractions.sendEvent</code> に渡されるデータが消費モジュールに配信される前に表示されます。
OnClearPersistedIdentities	このイベントは、 <code>SalesforceInteractions.clearPersistedIdentities</code> が呼び出されたときに表示されます。
OnConsentRevoke	<code>OnConsentRevoke</code> イベントは、 同意 (ページ 22) を提供したユーザが 後で同意を取り消した場合にディスパッチされます。
OnEventSend	<code>OnEventSend</code> イベントは、イベント API への要求が <code>SalesforceInteractions.sendEvent</code> を介して送信されるたびにディスパッチされます。このイベントには、イベントパイプラインで処理

カスタムイベント

説明

された `actionEvent` ペイロードが含まれます。

```
ActionEvent
```

OnException

`OnException` イベントは、操作が失敗した場合にディスパッチされ、エラーに関する情報とそのコンテンツを含みます。`OnException` イベントは、ランタイムエラーのデバッグと記録に使用できます。

```
{
  error: Error,
  context: string
}
```

OnInit

`OnInit` イベントは、Web SDK が初期化されたときにディスパッチされます。`OnInit` イベントは単一ページアプリケーション (SPA) のデバッグに使用できます。通常、単一ページアプリケーションでは、ページ読み込みイベントなしでビーコンとサイトマップを再初期化する必要があります。

OnInitSitemap

`OnInitSitemap` イベントは、Web SDK サイトマップが `SalesforceInteractions.initSitemap` 関数を介して初期化されるたびにディスパッチされます。

```
{
  currentKey: string,
  global: GlobalConfig,
  pageTypeDefault?: DefaultPageConfig,
  pageTypes: PageTypeConfig[],
  pageTypeConfig: PageTypeConfig[],
  settings: object
}
```

OnPageMatchStatusUpdated

`OnPageMatchStatusUpdated` イベントは、サイトマップのページ種別の一致状況が更新されるたびにディスパッチされます。

カスタムイベント

説明

matchStatus 配列は、イベントがディスパッチされるたびに、一致状況が更新された順序で蓄積されます。状況が *running* の場合、*PageMatchStatus* には開始時刻が含まれます。逆に、状況が *selected*、*rejected*、または *matched* の場合、*PageMatchStatus* には終了時刻が含まれます。*selected* ページ種別は、照合される最初のページ種別です。サイトマップでは、他の *matched* ページ種別は評価されません。

```
[{
    pageName: string;
    status: MatchStatus;
    startTime?: number;
    endTime?: number;
    _reject?: any;
}]
```

OnResetAnonymousId

このイベントは、*SalesforceInteractions.resetAnonymousId* が呼び出されたときに表示されます。

OnSetAnonymousId

OnSetAnonymousId は、Web SDK によって新しい追跡 ID が顧客に割り当てられたときに表示されます。

OnShutDown

OnShutDown イベントは、Web SDK にシャットダウン信号が送信されたときにディスパッチされます。



例:

```
document.addEventListener('salesforce:onEventSend', (customEvent) => {
    console.log('On Event Send: ', customEvent.detail)
});

SalesforceInteractions.sendEvent({
    interaction: {
        name: 'integration_hook_test'
```

```

    }
})

```

デバッグ

設定すると、Salesforce Interaction SDK でデバッグ目的でメッセージをブラウザコンソールに出力できます。デフォルトでは、ロガーは `none` のログレベルで設定されます。Web SDK で使用するロガーでは、標準ブラウザの `console.log` 関数がラップされます。このロガーには `SalesforceInteractions.log` でアクセスできます。

ログレベル

ログレベルを設定してデバッグ出力の量を制御できます。

Level	説明
trace	debug よりもきめ細かな情報メッセージを確認する場合に使用します。値: 5
debug	デバッグに役立つレベルのきめ細かな情報メッセージを確認する場合に使用します。値: 4
情報	大まかなレベルの情報メッセージを確認する場合に使用します。値: 3
warn	機能に影響する可能性がある潜在的な Web SDK の問題を確認する場合に使用します。値: 2
error	Web SDK が適切に機能していないことを示している可能性があるエラーイベントを確認する場合に使用します。値: 1
なし	ログを無効化することを示します。値: 0

この表では、最高のレベル(`trace`)から最低のレベル(`none`)の順でログレベルが示されています。各ログレベルで、その同じログレベル以下のすべてのメッセージが出力されます。

デバッグメソッド

```
getLoggingLevel(): LoggingLevel
```

Web SDK で使用されている現在のログレベルの数値を返します。

```
SalesforceInteractions.getLoggingLevel()
// => 0
```

```
setLoggingLevel(level?: LoggingLevel | keyof typeof LoggingLevel): void
```

Web SDK で使用するログレベルを設定します。

```
// using the logging level name
SalesforceInteractions.setLoggingLevel('error')
```

```
// using the logging level value
SalesforceInteractions.setLoggingLevel(1)
```

```
log.info(...args: any[]): void
```

info レベルでメッセージを出力します。

```
SalesforceInteractions.log.info("hello!")
```

```
log.debug(...args: any[]): void
```

debug レベルでメッセージを出力します。

```
SalesforceInteractions.log.debug("hello!")
```

```
log.error(...args: any[]): void
```

error レベルでメッセージを出力します。

```
SalesforceInteractions.log.error("hello!")
```

```
log.trace(...args: any[]): void
```

trace レベルでメッセージを出力します。

```
SalesforceInteractions.log.trace("hello!")
```

```
log.warn(...args: any[]): void
```

warn レベルでメッセージを出力します。

```
SalesforceInteractions.log.warn("hello!")
```

Salesforce Interactions SDK の Salesforce CDP モジュール

Salesforce Interactions SDK の Salesforce CDP モジュールは、SDK の機能を強化する拡張機能です。イベント構造を Salesforce CDP で取り込むことができる互換形式に変換し、イベントを認証して HTTP 経由で Salesforce CDP Web コネクタに配信できます。

Salesforce CDP で Web コネクタを設定するときに、提供される計測スクリプトにこの拡張モジュールが自動的に含まれ、設定されます。認識しておく必要があるいくつかの重要な詳細があります。

- [Web コネクタスキーマへの SDK イベントの変換](#) (ページ 51)
- [プロファイルイベントのルーティング](#) (ページ 69)
- [匿名および既知のプロファイルデータ](#) (ページ 73)
- [カスタムイベント](#) (ページ 72)

推奨スキーマ

Salesforce 全体で統合されたデータモデルの機能を使用するには、Web コネクタの設定時にこの[スキーマ](#)を使用します。

このスキーマには次の対応付けが含まれます。

- エンゲージメントイベント
 - カートインタラクションの対応付け
 - カタログインタラクションの対応付け
 - 注文インタラクションの対応付け
 - 同意イベントの対応付け
- プロファイルイベント
 - 連絡先メールの対応付け
 - 連絡先電話の対応付け
 - ID の対応付け
 - 関係者 ID の対応付け

[Web コネクタスキーマへの SDK イベントの変換](#)

Salesforce Interactions SDK のイベント構造が、Salesforce CDP に取り込むことができるデータにどのように変換されるかを理解することは、計測コードのトラブルシューティングを行う上で役立つ可能性があります。これを知ることで、スキーマを拡大し、推奨スキーマに含まれない追加のデータを取得することができます。

[プロファイルイベントのルーティング](#)

Web コネクタスキーマに記述されているプロファイルイベントは、リリース時に個々のデータストリームに対応付けられます。Salesforce Interactions SDK で収集されたプロファイルデータを適切なデータストリームにルーティングするには、取得したいいずれかの `user.attributes` データに `eventType` 項目を追加します。

[カスタムイベント](#)

Salesforce Interactions SDK で提供されている推奨のカートインタラクション、カタログインタラクション、および注文インタラクションデータモデルを使用して、Salesforce 全体で統合されたデータモデルを使用します。推奨のインタラクションデータモデルに加えて、カスタムイベントの作成と取得ができます。

[匿名および既知のプロファイルデータ](#)

Salesforce CDP で行動データを匿名ユーザに結合するには、まず匿名ユーザのプロファイルイベントを取得する必要があります。これを行わない場合、ログインなどのプロファイルイベントが取得されて匿名ユーザが既知になるまで、行動データを使用できません。

関連トピック:

[Web コネクタスキーマへの SDK イベントの変換](#)

Web コネクタスキーマへの SDK イベントの変換

Salesforce Interactions SDK のイベント構造が、Salesforce CDP に取り込むことができるデータにどのように変換されるかを理解することは、計測コードのトラブルシューティングを行う上で役立つ可能性があります。これをして、スキーマを拡大し、推奨スキーマに含まれない追加のデータを取得することができます。

必須 CDP 項目の自動入力

現在、Salesforce CDP イベントビーコンでは、すべてのイベントで 6 個の必須項目が必要とされます。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	category	text	必須。カテゴリがエンゲージメントであるかプロファイルであるかを定義するために使用します。
-	dateTime	date	必須。 yyyy-MM-ddTHH:mm:ss.SSSZ 形式のみをサポートします。 データのパーティション化に使用されるため、エンゲージメントイベントで重要です。
-	deviceId	text	必須。プロファイルイベントの主キー。
-	eventId	text	必須。エンゲージメントイベントの主キー。
-	eventType	text	必須。 Web コネクタスキーマ で定義されたイベントの developerName。
-	sessionId	text	必須。セッションの識別子。

Salesforce CDP に送信されるイベントでは、これらの項目は Salesforce CDP モジュールによって自動的に設定されます。これらの項目は、プロファイルイベントのルーティングやカスタムイベントなど、eventType を自動的に判別できない特定の場合を除き、指定する必要はありません。

Salesforce Web SDK 項目の自動入力

Salesforce Interactions SDK では、その[イベント構造](#)(ページ 8)の一部としていくつかの項目が自動的に取得されます。これらの項目はすべてのプロファイルイベントおよびエンゲージメントイベントに自動的に付加されます。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
interaction.name	interactionName	text	必須。インタラクションの内容を説明する名前。
pageView	pageView	text	Salesforce Interactions SDK により true または false の値が設定されます。この値は、イベントがページ読み込みの一環としてトリガされたか、ページとやり取りするユーザからトリガされたかを示します。
source.channel	sourceChannel	text	イベントの収集に使用された SDK チャネル。 Salesforce Interactions SDK の場合、この値は Web です。
source.locale	sourceLocale	text	ISO 639 Alpha-2 言語コードと ISO 3166 Alpha-2 国コードで表される現在のページのロケール。例: en_US、de_DE。
source.pageType	sourcePageType	text	イベントがサイトマップを介して収集されている場合、これは、イベントをトリガした一致する設定の pageType になります。
source.url	sourceUrl	text	イベントの送信元であるページの URL。
source.urlReferrer	sourceUrlReferrer	text	ユーザが訪問した以前の URL。

カートインタラクションの対応付け

カートインタラクション (ページ 10) を含むイベント構造は次のスキーマに対応付けられます。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	category = "Engagement"	text	必須。カテゴリがエンゲージメントであるかプロファ

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	dateTime	date	イルであるかを定義するために使用します。
-	deviceId	text	必須。必須。プロファイルイベントの主キー。
-	eventId	text	必須。エンゲージメントイベントの主キー。
-	eventType = "cart"	text	必須。 Web コネクタスキーマ で定義されたイベントの developerName。
-	sessionId	text	必須。セッションの識別子。
interaction.name	interactionName	text	必須。インタラクションの内容を説明する名前。
pageView	pageView	text	Salesforce Interactions SDK により true または false の値が設定されます。この値は、イベントがページ読み込みの一環としてトリガされたか、ページとやり取りするユーザからトリガされたかを示します。
source.channel	sessionId	text	イベントの収集に使用された SDK チャネル。 Salesforce Interactions SDK の場合、この値は Web です。
source.locale	sourceLocale	text	ISO 639 Alpha-2 言語コードと ISO 3166 Alpha-2 国コードで表される現在のページのロケール。例: en_US, de_DE。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
source.pageType	sourcePageType	text	イベントがサイトマップを介して収集されている場合、これは、イベントをトリガした一致する設定の pageType になります。
source.url	sourceUrl	text	イベントの送信元であるページの URL。
source.urlReferrer	sourceUrlReferrer	text	ユーザが訪問した以前の URL。

カートインタラクション (ページ 10) 項目の `interaction.lineItem` または `interaction.lineItems` に含まれる各品目データ (ページ 18) は、`cartEventId` が親 `cart` イベントの `eventId` に設定されている子イベントに対応付けられます。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	cartEventId	text	必須。この品目を含むカートイベントの <code>eventId</code> 。
-	category = "Engagement"	text	必須。カテゴリがエンゲージメントであるかプロファイルであるかを定義するために使用します。
-	dateTime	date	必須。 yyyy-MM-ddTHH:mm:ss.SSSZ 形式のみをサポートします。 データのパーティション化に使用されるため、エンゲージメントイベントで重要です。
-	deviceId	text	必須。プロファイルイベントの主キー。
-	eventId	text	必須。エンゲージメントイベントの主キー。
-	eventType = "cartItem"	text	必須。Web コネクタスキーマで定義されたイベントの <code>developerName</code> 。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	sessionId	text	必須。セッションの識別子。
lineItem.attributes.customFieldN	attributeCustomFieldN	any	ユーザ指定の値。
lineItem.attributes.customField0	attributeCustomField0	any	ユーザ指定の値。
lineItem.catalogObjectId	catalogObjectId	text	必須。品目内で参照されるカタログオブジェクトを表す一意の識別子。
lineItem.catalogObjectType	catalogObjectType	text	必須。品目内で参照されるカタログオブジェクトを表す種別名。
lineItem.currency	currency	text	価格項目の通貨。
lineItem.price	価格	number	品目内で参照されるカタログオブジェクトの価格。
lineItem.quantity	quantity	number	この品目内のカタログオブジェクトの数。



メモ: `lineItem.attributes` に追加されたプロパティは推奨スキーマでは自動的に定義されません。このデータを取得するには、これらのプロパティをスキーマに手動で追加してください。これらのプロパティは、ソースプロパティが対象スキーマにどのように対応付けられるかを示すためにのみここに示されています。

カタログインタラクションの対応付け

カタログインタラクション (ページ 13) を含むイベント構造 (ページ 8) は次のスキーマに対応付けられます。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	category = "Engagement"	text	必須。カテゴリがエンゲージメントであるかプロファイルであるかを定義するために使用します。
-	dateTime	date	必須。 yyyy-MM-ddTHH:mm:ss.SSSZ 形式のみをサポートします。 データのパーティション化に使用されるため、エンゲージメントイベントで重要です。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	deviceId	text	必須。プロファイルイベントの主キー。
-	eventId	text	必須。エンゲージメントイベントの主キー。
-	eventType = "catalog"	text	必須。Web コネクタスキーマで定義されたイベントの developerName。
-	sessionId	text	必須。セッションの識別子。
interaction.catalogObject.attributes.customFieldN	attributeCustomFieldN	any	ユーザ指定の値。
interaction.catalogObject.attributes.customField0	attributeCustomField0	any	ユーザ指定の値。
interaction.catalogObject.id	id	text	必須。カタログオブジェクトを表す一意の ID。
interaction.catalogObject.relatedCatalogObjectExample	relatedCatalogObjectExample0	any	関連するカタログオブジェクト ID のリストの最初の値。
interaction.catalogObject.type	type	text	必須。カタログオブジェクトを表す種別名。
interaction.name	interactionName	text	必須。インタラクションの内容を説明する名前。
pageView	pageView	text	Salesforce Interactions SDK により true または false の値が設定されます。この値は、イベントがページ読み込みの一環としてトリガされたか、ページとやり取りするユーザからトリガされたかを示します。
source.channel	sessionId	text	イベントの収集に使用された SDK チャネル。 Salesforce Interactions SDK の場合、この値は Web です。
source.locale	sourceLocale	text	ISO 639 Alpha-2 言語コードと ISO 3166 Alpha-2 国コードで表される現在のページ

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
			のロケール。例: en_US、de_DE。
source.pageType	sourcePageType	text	イベントがサイトマップを介して収集されている場合、これは、イベントをトリガした一致する設定の pageType になります。
source.url	sourceUrl	text	イベントの送信元であるページの URL。
source.urlReferrer	sourceUrlReferrer	text	ユーザが訪問した以前の URL。
...	relatedCatalogObjectExampleN	any	関連するカタログオブジェクト ID のリストの n 番目の値。



メモ: `interaction.catalogObject.attributes` および

`interaction.catalogObject.relatedCatalogObjects` に追加されたプロパティは推奨スキーマでは自動的に定義されません。このデータを取得するには、これらのプロパティをスキーマに手動で追加してください。これらのプロパティは、ソースプロパティが対象スキーマにどのように対応付けられるかを示すためにのみここに示されています。

注文インタラクションの対応付け

注文インタラクション (ページ 15) を含む イベント構造 (ページ 8) は次のスキーマに対応付けられます。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	category = "Engagement"	text	必須。カテゴリがエンゲージメントであるかプロファイルであるかを定義するために使用します。
-	dateTime	date	必須。 yyyy-MM-ddTHH:mm:ss.SSSZ 形式のみをサポートします。 データのパーティション化に使用されるため、エンゲージメントイベントで重要です。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	deviceId	text	必須。プロファイルイベントの主キー。
-	eventId	text	必須。エンゲージメントイベントの主キー。
-	eventType = "order"	text	必須。Web コネクタスキーマで定義されたイベントの developerName。
-	sessionId	text	必須。セッションの識別子。
interaction.name	interactionName	text	必須。インタラクションの内容を説明する名前。
interaction.order.currency	orderCurrency	text	注文の合計金額の通貨。
interaction.order.attributes.customFieldN	attributeCustomFieldN	any	ユーザ指定の値。
interaction.order.attributes.customField0	attributeCustomField0	any	ユーザ指定の値。
interaction.order.id	orderId	text	注文を表す一意の識別子。
interaction.order.totalValue	orderTotalValue	number	注文の合計金額。
pageView	pageView	text	Salesforce Interactions SDK により true または false の値が設定されます。この値は、イベントがページ読み込みの一環としてトリガされたか、ページとやり取りするユーザからトリガされたかを示します。
source.channel	sessionId	text	イベントの収集に使用された SDK チャネル。 Salesforce Interactions SDK の場合、この値は Web です。
source.locale	sourceLocale	text	ISO 639 Alpha-2 言語コードと ISO 3166 Alpha-2 国コードで表される現在のページのロケール。例: en_US、de_DE。
source.pageType	sourcePageType	text	イベントがサイトマップを介して収集されている

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
			場合、これは、イベントをトリガした一致する設定の pageType になります。
source.url	sourceUrl	text	イベントの送信元であるページの URL。
source.urlReferrer	sourceUrlReferrer	text	ユーザが訪問した以前の URL。

[注文インタラクション](#) (ページ 15) 項目の `interaction.order.lineItems` に含まれる各 [品目データ](#) (ページ 18) は、`orderEventId` が親 `order` イベントの `eventId` に設定されている子イベントに対応付けられます。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	category = "Engagement"	text	必須。カテゴリがエンゲージメントであるかプロファイルであるかを定義するために使用します。
-	dateTime	date	必須。 yyyy-MM-ddTHH:mm:ss.SSSZ 形式のみをサポートします。 データのパーティション化に使用されるため、エンゲージメントイベントで重要です。
-	eventId	text	必須。エンゲージメントイベントの主キー。
-	eventType = "order"	text	必須。Web コネクタスキーマで定義されたイベントの developerName。
-	deviceId	text	必須。プロファイルイベントの主キー。
-	orderEventId	text	必須。この品目を含む注文イベントの eventId。
-	sessionId	text	必須。セッションの識別子。
lineItem.attributes.customFieldN	attributeCustomFieldN	any	ユーザ指定の値。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
lineItem.attributes.customField0	attributeCustomField0	any	ユーザ指定の値。
lineItem.catalogObjectId	catalogObjectId	text	必須。品目内で参照されるカタログオブジェクトを表す一意の識別子。
lineItem.catalogObjectType	catalogObjectType	text	必須。品目内で参照されるカタログオブジェクトを表す種別名。
lineItem.currency	currency	text	価格項目の通貨。
lineItem.price	価格	number	品目内で参照されるカタログオブジェクトの価格。
lineItem.quantity	quantity	number	この品目内のカタログオブジェクトの数。



メモ: `lineItem.attributes` に追加されたプロパティは推奨スキーマでは自動的に定義されません。このデータを取得するには、これらのプロパティをスキーマに手動で追加してください。これらのプロパティは、ソースプロパティが対象スキーマにどのように対応付けられるかを示すためにのみここに示されています。

同意イベントの対応付け

イベント構造 (ページ 8) の `consents` 項目に含まれる各 **同意データ** (ページ 22) は次のスキーマに対応付けられます。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	category = "Engagement"	text	必須。カテゴリがエンゲージメントであるかプロファイルであるかを定義するために使用します。
-	dateTime	date	必須。 yyyy-MM-ddTHH:mm:ss.SSSZ 形式のみをサポートします。 データのパーティション化に使用されるため、エンゲージメントイベントで重要です。
-	deviceId	text	必須。プロファイルイベントの主キー。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	eventId	text	必須。エンゲージメントイベントの主キー。
-	eventType = "consentLog"	text	必須。Web コネクタスキーマで定義されたイベントの developerName。
-	sessionId	text	必須。セッションの識別子。
consent.provider	provider	text	ユーザが何に同意しているか。例: tracking (追跡)
consent.purpose	purpose	text	値: Opt In (オプトイン) または Opt Out (オプトアウト)
consent.status	status	text	必須。SDK に同意を提供する同意管理システムを示す指定識別子。

プロファイルイベントの対応付け

ユーザデータ (ページ 18) を含む イベント構造 (ページ 8) は次のスキーマに対応付けられます。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	category = "Profile"	text	必須。カテゴリがエンゲージメントであるかプロファイルであるかを定義するために使用します。
-	dateTime	date	必須。 yyyy-MM-ddTHH:mm:ss.SSSZ 形式のみをサポートします。データのパーティション化に使用されるため、エンゲージメントイベントで重要です。
-	deviceId	text	必須。プロファイルイベントの主キー。
-	eventId	text	必須。エンゲージメントイベントの主キー。
-	sessionId	text	必須。セッションの識別子。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
interaction.name	interactionName	text	必須。インタラクションの内容を説明する名前
pageView	pageView	text	Salesforce Interactions SDK により true または false の値が設定されます。この値は、イベントがページ読み込みの一環としてトリガされたか、ページとやり取りするユーザからトリガされたかを示します。
source.channel	sessionId	text	イベントの収集に使用された SDK チャネル。 Salesforce Interactions SDK の場合、この値は Web です。
source.locale	sourceLocale	text	ISO 639 Alpha-2 言語コードと ISO 3166 Alpha-2 国コードで表される現在のページのロケール。例: en_US、de_DE。
source.pageType	sourcePageType	text	イベントがサイトマップを介して収集されている場合、これは、イベントをトリガした一致する設定の pageType になります。
source.url	sourceUrl	text	イベントの送信元であるページの URL。
source.urlReferrer	sourceUrlReferrer	text	ユーザが訪問した以前の URL。
user.attributes.customFieldN	customFieldN	any	ユーザ指定の値。
user.attributes.customField0	customField0	any	ユーザ指定の値。
user.attributes.eventType	eventType	text	必須。Web コネクタスキーマで定義されたイベントの developerName。



メモ: user.attributes.eventType 項目に追加されたプロパティは必須です。

連絡先メールの対応付け

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	dateTime	date	必須。 yyyy-MM-ddTHH:mm:ss.SSSZ 形式のみをサポートします。データのパーティション化に使用されるため、エンゲージメントイベントで重要です。
-	eventId	text	必須。エンゲージメントイベントの主キー。
user.attributes.eventType	eventType = "contactPointEmail"	text	必須。Web コネクタスキーマで定義されたイベントの developerName。
-	sessionId	text	必須。セッションの識別子。
-	deviceId	text	必須。プロファイルイベントの主キー。
-	category = "Profile"	text	必須。カテゴリがエンゲージメントであるかプロファイルであるかを定義するために使用します。
interaction.name	interactionName	text	必須。インタラクションの内容を説明する名前
pageView	pageView	text	Salesforce Interactions SDK により true または false の値が設定されます。この値は、イベントがページ読み込みの一環としてトリガされたか、ページとやり取りするユーザからトリガされたかを示します。
source.channel	sessionId	text	イベントの収集に使用された SDK チャネル。 Salesforce Interactions SDK の場合、この値は Web です。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
source.locale	sourceLocale	text	ISO 639 Alpha-2 言語コードと ISO 3166 Alpha-2 国コードで表される現在のページのロケール。例: en_US、de_DE。
source.pageType	sourcePageType	text	イベントがサイトマップを介して収集されている場合、これは、イベントをトリガした一致する設定の pageType になります。
source.url	sourceUrl	text	イベントの送信元であるページの URL。
source.urlReferrer	sourceUrlReferrer	text	ユーザが訪問した以前の URL。
user.attributes.email	email	text	ユーザのメールアドレス。
user.attributes.eventType	eventType = "contactPointEmail"	text	必須。Web コネクタスキーマで定義されたイベントの developerName。

連絡先電話の対応付け

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	category = "Profile"	text	必須。カテゴリがエンジニアメントであるかプロファイルであるかを定義するために使用します。
-	dateTime	date	必須。 yyyy-MM-ddTHH:mm:ss.SSSZ 形式のみをサポートします。データのパーティション化に使用されるため、エンゲージメントイベントで重要です。
-	deviceid	text	必須。プロファイルイベントの主キー。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	eventId	text	必須。エンゲージメントイベントの主キー。
-	sessionId	text	必須。セッションの識別子。
interaction.name	interactionName	text	必須。インタラクションの内容を説明する名前
pageView	pageView	text	Salesforce Interactions SDK により true または false の値が設定されます。この値は、イベントがページ読み込みの一環としてトリガされたか、ページとやり取りするユーザからトリガされたかを示します。
source.channel	sessionId	text	イベントの収集に使用された SDK チャネル。 Salesforce Interactions SDK の場合、この値は Web です。
source.locale	sourceLocale	text	ISO 639 Alpha-2 言語コードと ISO 3166 Alpha-2 国コードで表される現在のページのロケール。例: en_US、de_DE。
source.pageType	sourcePageType	text	イベントがサイトマップを介して収集されている場合、これは、イベントをトリガした一致する設定の pageType になります。
source.url	sourceUrl	text	イベントの送信元であるページの URL。
source.urlReferrer	sourceUrlReferrer	text	ユーザが訪問した以前の URL。
user.attributes.eventType	eventType = "contactPointPhone"	text	必須。Web コネクタスキーマで定義されたイベントの developerName。
user.attributes.phoneNumber	phoneNumber	text	必須。ユーザの電話番号。

ID の対応付け

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	category = "Profile"	text	必須。カテゴリがエンゲージメントであるかプロファイルであるかを定義するために使用します。
-	dateTime	date	必須。 yyyy-MM-ddTHH:mm:ss.SSSZ 形式のみをサポートします。 データのパーティション化に使用されるため、エンゲージメントイベントで重要です。
-	deviceId	text	必須。プロファイルイベントの主キー。
-	eventId	text	必須。エンゲージメントイベントの主キー。
-	sessionId	text	必須。セッションの識別子。
interaction.name	interactionName	text	必須。インタラクションの内容を説明する名前
pageView	pageView	text	Salesforce Interactions SDK により true または false の値が設定されます。この値は、イベントがページ読み込みの一環としてトリガされたか、ページとやり取りするユーザからトリガされたかを示します。
source.channel	sessionId	text	イベントの収集に使用された SDK チャネル。 Salesforce Interactions SDK の場合、この値は Web です。
source.locale	sourceLocale	text	ISO 639 Alpha-2 言語コードと ISO 3166 Alpha-2 国コードで表される現在のページのロケール。例: en_US、de_DE。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
source.pageType	sourcePageType	text	イベントがサイトマップを介して収集されている場合、これは、イベントをトリガした一致する設定の pageType になります。
source.url	sourceUrl	text	イベントの送信元であるページの URL。
source.urlReferrer	sourceUrlReferrer	text	ユーザが訪問した以前の URL。
user.attributes.eventType	eventType = "identity"	text	必須。Web コネクタスキーマで定義されたイベントの developerName。
user.attributes.firstName	firstName	text	必須。ユーザの名。
user.attributes.isAnonymous	isAnonymous	number	必須。このプロファイルイベントが匿名ユーザを識別するか既知のユーザを識別するかを示す値 (0 (false) または 1 (true))。
user.attributes.lastName	lastName	text	必須。ユーザの名前(姓)。

関係者 ID の対応付け

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	category = "Profile"	text	必須。カテゴリがエンゲージメントであるかプロファイルであるかを定義するために使用します。
-	dateTime	date	必須。 yyyy-MM-ddTHH:mm:ss.SSSZ 形式のみをサポートします。データのパーティション化に使用されるため、エンゲージメントイベントで重要です。
-	deviceId	text	必須。プロファイルイベントの主キー。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
-	eventId	text	必須。エンゲージメントイベントの主キー。
-	sessionId	text	必須。セッションの識別子。
interaction.name	interactionName	text	必須。インタラクションの内容を説明する名前
pageView	pageView	text	Salesforce Interactions SDK により true または false の値が設定されます。この値は、イベントがページ読み込みの一環としてトリガされたか、ページとやり取りするユーザからトリガされたかを示します。
source.channel	sessionId	text	イベントの収集に使用された SDK チャネル。 Salesforce Interactions SDK の場合、この値は Web です。
source.locale	sourceLocale	text	ISO 639 Alpha-2 言語コードと ISO 3166 Alpha-2 国コードで表される現在のページのロケール。例: en_US、de_DE。
source.pageType	sourcePageType	text	イベントがサイトマップを介して収集されている場合、これは、イベントをトリガした一致する設定の pageType になります。
source.url	sourceUrl	text	イベントの送信元であるページの URL。
source.urlReferrer	sourceUrlReferrer	text	ユーザが訪問した以前の URL。
user.attributes.eventType	eventType = "partyIdentification"	text	Web コネクタスキーマで定義されたイベントの developerName
user.attributes.IDNameWeb	IDNameWeb	text	必須。関係者 ID 名。

ソースイベントプロパティ	対象スキーマ API 参照名	項目の型	説明
user.attributes.IDType	IDType	text	必須。関係者 ID 種別。
user.attributes.userId	userId	text	必須。関係者のユーザ ID。

プロファイルイベントのルーティング

Webコネクタスキーマに記述されているプロファイルイベントは、リリース時に個々のデータストリームに対応付けられます。Salesforce Interactions SDK で収集されたプロファイルデータを適切なデータストリームにルーティングするには、取得したいいずれかの `user.attributes` データに `eventType` 項目を追加します。

`eventType` の値を、Web コネクタスキーマで指定されたプロファイルイベントの `developerName` と照合します。このプロファイルイベントに対してリストされたすべての必須のスキーマ項目を取得する必要があります。

 **例:** `contactPointEmail` イベントを含む次のスキーマがあるとします。

```
{
  "records": [
    {
      "developerName": "contactPointEmail",
      "masterLabel": "Contact Point Email",
      "category": "Profile",
      "externalDataTranFields": [
        {
          "masterLabel": "eventId",
          "dataType": "Text",
          "developerName": "eventId",
          "isDataRequired": true
        },
        {
          "masterLabel": "category",
          "dataType": "Text",
          "developerName": "category",
          "isDataRequired": true
        },
        {
          "masterLabel": "dateTime",
          "dataType": "Date",
          "developerName": "dateTime",
          "isDataRequired": true
        },
        {
          "masterLabel": "deviceId",
          "dataType": "Text",
          "developerName": "deviceId",
          "isDataRequired": true
        }
      ]
    }
  ]
}
```

```
        "developerName": "deviceId",
        "isDataRequired": true,
        "primaryIndexOrder": 1
    },
    {
        "masterLabel": "email",
        "dataType": "Text",
        "developerName": "email",
        "isDataRequired": true
    },
    {
        "masterLabel": "eventType",
        "dataType": "Text",
        "developerName": "eventType",
        "isDataRequired": true
    },
    {
        "masterLabel": "interactionName",
        "dataType": "Text",
        "developerName": "interactionName",
        "isDataRequired": true
    },
    {
        "masterLabel": "pageView",
        "dataType": "Text",
        "developerName": "pageView",
        "isDataRequired": false
    },
    {
        "masterLabel": "sessionId",
        "dataType": "Text",
        "developerName": "sessionId",
        "isDataRequired": true
    },
    {
        "masterLabel": "sourceChannel",
        "dataType": "Text",
        "developerName": "sourceChannel",
        "isDataRequired": false
    },
    {
        "masterLabel": "sourceLocale",
        "dataType": "Text",
        "developerName": "sourceLocale",
        "isDataRequired": true
    }
```

```

        "isDataRequired": false
    },
    {
        "masterLabel": "sourcePageType",
        "dataType": "Text",
        "developerName": "sourcePageType",
        "isDataRequired": false
    },
    {
        "masterLabel": "sourceUrl",
        "dataType": "Text",
        "developerName": "sourceUrl",
        "isDataRequired": false
    },
    {
        "masterLabel": "sourceUrlReferrer",
        "dataType": "Text",
        "developerName": "sourceUrlReferrer",
        "isDataRequired": false
    }
]
}
]
}

```

Salesforce Interactions SDK で `contactPointEmail` プロファイルイベントを取得するための計測環境を整えます。

```

SalesforceInteractions.sendEvent({
  user: {
    attributes: {
      eventType: 'contactPointEmail',
      email: 'user@domain.com'
    }
  }
})

```

Salesforce CDP モジュールにより Salesforce CDP および Salesforce Interactions SDK の必須項目が自動的に入力されます。

関連トピック:

[Web コネクタスキーマへの SDK イベントの変換](#)

[データ収集](#)

[ユーザデータ](#)

カスタムイベント

Salesforce Interactions SDKで提供されている推奨のカートインターラクション、カタログインターラクション、および注文インターラクションデータモデルを使用して、Salesforce全体で統合されたデータモデルを使用します。推奨のインターラクションデータモデルに加えて、カスタムイベントの作成と取得ができます。

Web コネクタスキーマでカスタムイベントの形式を定義します。次に例を示します。

```
{
  "records": [
    {
      "developerName": "myCustomEvent",
      "masterLabel": "My Custom Event",
      "category": "Engagement",
      "externalDataTranFields": [
        {
          "masterLabel": "My Custom Field",
          "dataType": "Text",
          "developerName": "myCustomField",
          "isDataRequired": true
        },
        {
          "masterLabel": "eventId",
          "dataType": "Text",
          "developerName": "eventId",
          "isDataRequired": true
        },
        {
          "masterLabel": "category",
          "dataType": "Text",
          "developerName": "category",
          "isDataRequired": true
        },
        {
          "masterLabel": "dateTime",
          "dataType": "Date",
          "developerName": "dateTime",
          "isDataRequired": true
        },
        {
          "masterLabel": "deviceId",
          "dataType": "Text",
          "developerName": "deviceId",
          "isDataRequired": true,
          "primaryIndexOrder": 1
        },
        {
          "masterLabel": "eventType",
          "dataType": "Text",
          "developerName": "eventType",
          "isDataRequired": true
        },
        {
          "masterLabel": "sessionId",
        }
      ]
    }
  ]
}
```

```

        "dataType": "Text",
        "developerName": "sessionId",
        "isDataRequired": true
    }
]
}
]
}

```

次に、Salesforce Interactions SDK で、`eventType` または `name` 項目と、必須 CDP 項目の自動入力では設定されない必須項目を指定して、このイベントを取得します。

```

SalesforceInteractions.sendEvent({
  interaction: {
    name: 'myCustomEvent',
    eventType: 'myCustomEvent',
    myCustomField: 'some value'
  }
})

```

 **メモ:** `eventType` を指定せずに `name` を指定した場合、`eventType` はデフォルトで `name` の値になります。

匿名および既知のプロファイルデータ

Salesforce CDP で行動データを匿名ユーザに結合するには、まず匿名ユーザのプロファイルイベントを取得する必要があります。これを行わない場合、ログインなどのプロファイルイベントが取得されて匿名ユーザが既知になるまで、行動データを使用できません。

 **例:** 推奨スキーマを使用すると、この目的で `identity` イベントを使用できます。たとえば、識別子が作成または変更されるたびに Web SDK で新しい `anonymousId` が顧客に割り当てられて空白の `identity` イベントが送信されます。

```

const { OnSetAnonymousId } = SalesforceInteractions.CustomEvents
document.addEventListener(OnSetAnonymousId, () => {
  SalesforceInteractions.sendEvent({
    user: {
      attributes: {
        eventType: 'identity',
        isAnonymous: 1,
        firstName: '',
        lastName: ''
      }
    }
  })
})

```

第4章

Salesforce CDP Web SDK

Salesforce CDP Web SDK は、イベントを Salesforce CDP に送信するように Web サイトを設定するための低レベル API です。顧客エンゲージメントデータとユーザプロファイルデータの両方を収集して Salesforce CDP アプリケーションに送信するように Web サイトを設定します。同意管理システムが組み込まれており、ユーザが許可した場合にのみデータを収集できるようになっています。これらのコールは Salesforce CDP アプリケーションインターフェースから使用することで最大限に活用できますが、テナント固有のエンドポイントを使用してメソッドを外部からコールすることもできます。

! **重要:** 新しいバージョンの Web SDK を使用できます。Salesforce CDP Web SDK バージョンをまだ実装していない場合は、最新の [Salesforce Interactions SDK バージョン](#) を実装することをお勧めします。

前提条件

Salesforce CDP Web SDK を使用するには、Salesforce データソースコネクタを作成します。Salesforce CDP にログインして [Web コネクタを設定](#) します。設定の詳細は、『[Salesforce CDP 実装ガイド](#)』を参照してください。

Web サイトコネクタのソース ID と [テナント固有のエンドポイント](#) を使用して Salesforce CDP Web SDK を初期化します。

SDK のダウンロード

Salesforce CDP Web SDK のソースをダウンロードします。

- [Salesforce CDP Web SDK](#)

SDK の設定

Web SDK は、この設定オブジェクトを消費してイベントリスナを自動的に生成し、関連 DOM 要素に割り当てます。設定オブジェクトの構造には、次のようにいくつかの属性があります。client 以外のすべての属性は省略可能です。

```
{  
  client: {},
```

```

    signals: [],
    selectors: {},
    schemas: {},
    dataProviders: {}
}

```

SDK の設定: クライアントパラメータ

設定	型	説明
appSourceId	string	必須。この値は、Salesforce システム管理者から入手してください。
beaconEndpoint	string	必須。次の構造を使用します。 http://<Tenant Specific Endpoint>/web/events
authEndpoint	string	次の構造を使用します。 http://<Tenant Specific Endpoint>/web/authentication
deviceId	string	クライアント設定で定義することもできます。指定しなかった場合は、SDKによって作成、管理される識別子が使用されます。
sessionId	string	Web SDKによって自動的に入力されます。
consentEventTypeName	string	同意イベント種別の名前を設定します。ほとんどの設定ではデフォルト名の consentLog で問題ありません。
retryAttempts	Integer	ビーコンサービスへの送信に失敗したイベントの再試行回数です。

設定	型	説明
retryDelayMS	Integer	ビーコンサービスへの送信に失敗したイベントの再試行間隔(ミリ秒)です。
automaticallyTrackNavigationEvents	Boolean	このフラグを有効にすると、Web SDK はユーザによるページナビゲーションイベントを追跡して、 <code>document.location.hash</code> と <code>document.location.pathname</code> を取得します。

SDK の設定: 信号

信号は、Web SDK エンジンがイベントリスナーをページ上の適切な DOM 要素に割り当てるために使用する、設定オブジェクトの主要部分です。信号は設定の重要なピースであり、初期化後に更新する必要があります。信号の `name`、`schema`、`category` は必須項目です。

```
{
  name: 'signal name',
  schema: 'developerName of schemaEvent',
  category: 'category', // example: engagement
  event: {
    type: 'click',
    selector: 'reference to selector name'
  },
  mapping: {
    "developerName of schemaEvent field": {
      from: 'page',
      selector: 'css selector or reference to selector name'
    }
  }
}
```

 **メモ:** `CDP.configure` メソッドをコールすると、新しい信号セットが登録される前に、ページ上のすべての登録済み信号が削除されます。

SDK の設定: 信号の対応付け

登録済み信号の `mapping` 項目は、Web ページからデータを取得して信号の対象スキーマセット項目に対応付けるために使用されます。2つの対応付け設定がサポートされます。

- ページ

```
signal: {
  ...
  mapping: {
    "schemaFieldDeveloperName": {
      from: "page",
      selector: "css selector or reference to selector name",
      scope: "event"
    }
  }
}
```

ページ対応付けでは、セレクタと一致する要素のテキスト値を取得して、スキーマイベント項目の値として使用します。`scope` 値は、DOM内のどの要素を使用してセレクタクエリを実行するかを制御します。`event` は、取得する信号をトリガした要素を示します。他のすべての値は、ドキュメントのルートから開始されたことを示します。

- データ

```
signal: {
  ...
  mapping: {
    "schemaFieldDeveloperName": {
      from: "data",
      provider: "dataProviderName",
      attribute: "attributeName"
    }
  }
}
```

データ対応付けは、データプロバイダから返された値を取得します。データプロバイダは、JavaScript オブジェクトを返します。`attribute` 値は、そのオブジェクトの対象属性をスキーマイベント項目に対応付けるために使用されます。

SDK の設定: セレクタ

信号内で参照されるすべてのセレクタはここで設定します。各セレクタは、1つ以上の DOM 要素を返します。Web SDK では 3 種類のセレクタがサポートされます。

- セレクタは、DOM要素へのパスを定義する CSS セレクタのみでも構成できます。
- セレクタは、CSS セレクタと、パスの最後の要素を示すテキスト属性で構成できます。
- CSS セレクタとテキストの一致だけでは必要な機能を実装できない特別なケースには、カスタム JavaScript を作成して対応できます。

```
selectors: {
```

```

    "unique selector name": {
        selector: '<CSS selector (e.g. ul li.productName)>'
    },
    "unique selector name": {
        selector: '<CSS selector (e.g. ul li.productName)>',
        containsText: "<Text pattern to match inside the DOM
Element(s).>"
    },
    "unique selector name": {
        selector: function() {
            // Creating a dynamic selector based on data from the
            page's data layer
            return document.getElementById("product_" +
dataLayer.get('currentProductId'));
        }
    }
}

```

SDK の設定: スキーマ

[スキーマ](#)をエクスポートして SDK に読み込みます。

```

schemas: {
    schemaId: {
        ...
    }
}

```

SDK の設定: データプロバイダ

データプロバイダは、画面上の要素から直接取得できない情報を提供します。これらは、データレイヤから JavaScript オブジェクトを返す関数です。この関数は、処理する信号をトリガしたイベントが含まれる `domEvent` パラメータを渡します。

```

dataProviders: {
    functionName (domEvent) {
        return <data based on the input param>
    }
}

```

設定オブジェクトの例

```
{
    signals: [
        {

```

```
        name: 'Remove from cart button',
        category: "Engagement",
        schema: 'RemoveFromCart',
        event: {
            type: 'click',
            selector: "remove from cart button selector"
        },
        providers: [
            'productCatalog'
        ],
        mapping: {
            ProductId: {
                from: 'data',
                provider: 'productCatalog',
                attribute: 'ProductId'
            },
            ShoppingCartId: {
                from: 'data',
                provider: 'productCatalog',
                attribute: 'ShoppingCartId'
            }
        }
    },
    {
        name: 'Add to cart button',
        schema: 'AddToCart',
        category: "Engagement",
        event: {
            type: 'click',
            selector: "add to cart button selector"
        },
        providers: [
            'productCatalog'
        ],
        mapping: {
            ProductId: {
                from: 'data',
                provider: 'productCatalog',
                attribute: 'ProductId'
            },
            ShoppingCartId: {
                from: 'data',
                provider: 'productCatalog',
                attribute: 'ShoppingCartId'
            }
        }
    },
    {
        name: 'Username typed',
        schema: 'usernameTyped',
        category: 'Profile',
        event: {
            type: 'keydown',
            selector: "username textbox"
        }
    }
]
```

```

        },
        mapping: {
            username: {
                from: 'page',
                selector: "username textbox",
                scope: 'document'
            }
        }
    },
{
    name: 'Product click',
    schema: 'ProductViewed',
    category: "Engagement",
    event: {
        type: 'click',
        selector: "product selectors"
    },
    providers: [
        'productCatalog'
    ],
    mapping: {
        ProductId: {
            from: 'data',
            provider: 'productCatalog',
            attribute: 'ProductId'
        }
    }
},
{
    name: 'opt-in',
    schema: 'consent-opt-in',
    category: "Consent",
    event: {
        type: 'click',
        selector: "consent opt-in selector"
    }
},
{
    name: 'opt-out',
    schema: 'consent-opt-out',
    category: "Consent",
    event: {
        type: 'click',
        selector: "consent opt-out selector"
    }
}
],
dataProviders: {
    productCatalog (domEvent) {
        return {
            "ProductId": "product-id-9aba47743f27",
            "ShoppingCartId": "shopping-cart-id-9aba47743f27"
        }
    }
}

```

```

        },
        selectors: {
            "remove from cart button selector": {
                selector: 'body > div.wrapper.svelte-1a3asx9 > main > div > div.slds-col.slds-size_2-of-3 > div:nth-child(3) > div:nth-child(2) > div:nth-child(6) > button:nth-child(2)'
            },
            "add to cart button selector": {
                selector: 'button.slds-button.slds-button_destructive',
                containsText: "ADD"
            },
            "product selectors": {
                selector: function() {
                    return document.querySelectorAll('.stencil');
                }
            },
            "consent opt-in selector": {
                selector: '#dialog-body-id-42 > a > div > p > button.slds-button.slds-button_brand'
            },
            "consent opt-out selector": {
                selector: '#dialog-body-id-42 > a > div > p > button.slds-button.slds-button_destructive'
            },
            "username textbox": {
                selector: '#username'
            }
        }
    }
}

```

SDK の初期化

ページ上で SDK のライフサイクルの最初のステップは初期化です。初期化では、パラメータを設定(例: Web サイトコネクタのソース ID を appSourceId に設定)し、テナント固有のエンドポイントから作成された承認エンドポイントやビーコンエンドポイントを SDK に渡します。

```

        window.addEventListener('load', (event) => {
const CONFIG = {
    client: {
        "appSourceId": "app source id as guid",
        "deviceId": "deviceId as guid", // optional
        "beaconEndpoint": "http://server.internal:8080/web/events",
        "authEndpoint":
        "http://server.internal:8080/web/authentication"
    }
};
CDP.initialize(CONFIG);
});

```

- CDP.initialize(CONFIG) - 初期読み込み時に SDK を設定します。クライアントの設定が必要です。
- CDP.register(CONFIG) - 設定パラメータを更新、作成、上書きします。

URL からのパラメータの取得

Web サイトのユーザ追跡コードで実行する一般的な動作として、URL で渡されたクエリパラメータから値を取得することができます。Salesforce CDP Web SDK には次の get 関数が用意されており、Web アドレスからパラメータを取得します。

```
CDP.getUrlParam('trackingId')
```

イベントの送信

Salesforce CDP Web SDK の主な機能は、Web サイトから Salesforce CDP に信号を送り戻すことです。複雑なバックグラウンド処理の大半は sendEvent() メソッドが担ってくれるため、開発者は承認の処理、ペイロードの作成、接続の管理などに煩わされることなく、送信するデータの収集に集中できます。送信イベントは [スキーマ](#) と一致する必要があります。

```
CDP.sendEvent("engagement", "ProductView", productData)
```

この例では、商品ビューエンゲージメントイベントを送信しています。商品データは、DOM、データオブジェクト、URL パラメータ、またはこれらのソースの組み合わせから取得されます。Salesforce CDP イベントのすべての必須項目は SDK によって入力され、提供されたイベントデータと組み合わされます。

デフォルトでは、SDK は GET でイベントを送信しますが、イベントペイロードが 1024 文字を超える場合は、コンテンツ種別の application/x-www-form-urlencoded を使用した POST を実行します。

同意の管理

Salesforce CDP Web SDK は、セッションを超えて同意設定を維持しません。各サイトビギナーごとに同意設定を渡してください。この設定により、Web サイトで使用している既存の同意管理ソリューションに SDK を容易に統合できます。

使用している同意管理システムからのコールバックで下記の関数を使用してください。SDK が保存している Cookie は、Web サイトの同意管理には使用できません。SDK は、イベントを送信する前に同意が得られているかどうかを確認するだけです。

```
CDP.consentOptIn();
```

このコールは、トリガされたアクションをイベントとして通知する許可を SDK に付与します。このコールは、`CDP.consentOptIn()`；イベントを送信して、ユーザがオプトインしたことを示す Cookie を作成します。以降のページ読み込み時にこの Cookie が存在すると、直ちにイベントの転送が開始されます。

```
CDP.consentOptOut();
```

このコールは、同意を削除してイベントの送信を中止します。Cookie が削除され、イベントの転送が直ちに停止します。

第5章

Salesforce CDP Mobile SDK

Salesforce CDP Mobile SDK を使用すると、モバイルアプリケーションを Salesforce CDP (カスタマーデータプラットフォーム)に統合して、アプリケーションのライフサイクルや画面ナビゲーションイベントなど、エンドユーザから豊富な行動データやデモグラフィックデータを取得できるようになります。Salesforce CDP が収集したデータから顧客の統一プロファイルを作成することで、複数の Salesforce クラウド(マーケティング、コマース、サービスなど)に渡って高度にパーソナライズされたエンゲージメントが可能になります。

前提条件

Salesforce CDP Mobile SDK を使用するには、Salesforce データソースコネクタを作成して、エンドポイントを自動生成します。Salesforce CDP アプリケーションにログインして、[Web サイトおよびモバイルアプリケーション] 画面の左側のパネルに移動します。ユーザインターフェースでアプリケーションコネクタを作成することで、データソースを設定します。詳細は、『[Salesforce CDP 実装ガイド](#)』を参照してください。

Web データコネクタから取得した appSourceId と生成されたエンドポイントを使用して Salesforce CDP Mobile SDK を初期化します。

SDK のダウンロード

Salesforce CDP Mobile SDK を次のいずれかのソースからダウンロードします。

- [Android](#)
- [iOS](#)

SDK のインストール

Salesforce CDP Mobile SDK をモバイルアプリケーションに追加するには、CocoaPods を使用するか、またはソースコードにコールを組み込みます。

CocoaPods

CocoaPodsは、プロジェクト用のサードパーティライブラリを容易に一元管理できるようにする場所を提供します。CocoaPodsを使用して Salesforce CDP Mobile SDK を iOS モバイルアプリケーションに統合するには、Podfileを作成して連動関係を追加します。

```
use_frameworks!

target '<Your Target>' do
    pod 'SALESFORCE-CDP-SDK-IOS-SDK', '~> 1.0'
end
```

手動でのインストール

手動でインストールするには、自分で書いたコードですべての連動関係を管理する必要があります。複数のライブラリが連動関係を共有していて、それが異なるバージョンを指定できる場合には、すべての連動関係を満足する最適なライブラリバージョンをモバイルアプリケーション開発者の責任で指定する必要があります。

SDK の設定

APIを使用するWebサービスをSalesforce CDP Mobile SDKを使用して呼び出すためには、事前に設定が必要です。CdpConfigBuilderヘルパークラスを使用してSDKを設定し、必要なappIdとエンドポイントを指定します。必要に応じてtrackScreens、trackLifecycle、sessionTimeoutも設定します。

```
let config = CdpConfigBuilder(appId: {REPLACE WITH APP ID},
endpoint: {REPLACE WITH ENDPOINT})
    .trackScreens({True|False})
    .trackLifecycle({True|False})
    .sessionTimeout({Int})
    .build()
```

SDK の設定手順

Salesforce CDP Mobile SDK は 2 ステップの手順で使用できるようになります。

1. Salesforce CDP Mobile SDK を設定して、Salesforce CDP の (UUID 形式の) アプリケーション ID と Salesforce Tenant Specific Endpoint (TSE) を指定します。

```
let config = CdpConfigBuilder(appId: {REPLACE WITH APP ID}, endpoint:
```

```
{REPLACE WITH ENDPOINT}).build()
```

2. Salesforce CDP Mobile SDK を初期化します。

```
_ = CdpSdk.configure(config)
Congratulations! The CDP SDK is now configured and ready
to use. You can verify that by inspecting the state of the CDP
SDK instance:
CdpSdk.shared.state
Example output:
{
  "name": "cdp",
  "config": {
    "sessionDuration": 600,
    "screenTrackingEnabled": false,
    "lifecycleEventTrackingEnabled": false,
    "appId": "YOUR APP ID",
    "endpoint": "YOUR ENDPOINT"
  },
  "consentManager": {
    "deviceId": "YOUR DEVICE ID",
    "consent": "pending"
  },
  "eventManager": {
    "queueSize": 0
  },
  "sessionManager": {
    "sessionId": "SESSION ID"
  }
}
```

設定しなかったプロパティではデフォルト値が使用されます。

設定パラメータ

Salesforce CDP Mobile SDK の設定パラメータを下表に示します。

設定	型	説明
appId	string	必須。 appId 設定は、モバイルアプリケーションを Salesforce CDP プラットフォームに対して一意に識別し、有効な UUID(ユニバーサル識別子) の形式で指定する必要があります。 CdpSdkBuilder に対して無効な appId を指定すると、致命的なエラーが発生します。
endpoint	string	必須。 endpoint 設定は、作成および追跡されるイベントを Salesforce CDP Mobile SDK Web サー

設定	型	説明
		<p>ビスがどこに送信すべきかを指定します。エンドポイントは、次の形式の有効な Salesforce CDP URL である必要があります。</p> <pre>https://{{tenant specific endpoint id}}.c360a.salesforce.com. [description here of allowed domains]</pre>
		CdpSdkBuilderに対して無効なエンドポイントを指定すると、致命的なエラーが発生します。
trackScreens	boolean	<p>省略可能。デフォルトは false です。trackScreensを有効化すると、モバイルアプリケーションの ScreenEntry イベントを自動的に追跡します。イベントは、イベントタイプが「ScreenView」、属性が「screenname」のエンゲージメントイベントとして収集されます。</p>
trackLifecycle	boolean	<p>省略可能。デフォルトは false です。trackLifecycleを有効化すると、アプリケーションライフサイクルイベントの AppForegrounded と AppVersionChanged を自動的に追跡します。イベントはエンゲージメントイベントとして収集されます。AppVersionChanged イベントのイベントタイプは「AppUpdate」、属性は「previousVersion」です。AppForegrounded イベントのイベントタイプは「AppFirstLaunch」または「AppLaunch」で、属性はありません。</p>
sessionTimeoutInSeconds	int	<p>省略可能。デフォルトは 600 です。sessionTimeoutInSeconds 設定は、モバイルアプリケーションがバックグラウンドの状態で何秒経過したら現在のセッションが期限切れになるかを表します。セッション管理は次のように行われます。まず、ユーザーがモバイルアプリケーションを起動すると、Salesforce CDP Mobile SDK は新しいセッションを開始します。このセッションは、アプリケーションがフォアグラウンドの状態のときに維持されます。ユーザーがモバイルア</p>

設定	型	説明
		プリケーションをバックグラウンドに送ると、SDKはタイマーを起動して、現在のセッションが期限切れになるまでの時間を計測します。セッションが期限切れになる前にユーザがモバイルアプリケーションをフォアグラウンドに戻すと、タイマーがクリアされてセッションが継続します。そうでなければセッションが終了します。

次に、`CdpSdk` オブジェクトに対して `configure` メソッドをコールして、定義されている設定で共有インスタンスをインスタンス化します。

```
let sdk = CdpSdk.configure(config)
```

これで、Salesforce CDP Mobile SDK がイベントを収集できるようになります。Salesforce CDP Mobile SDK シングルトンインスタンスは、`CdpSdk.shared` を使用して参照します。

```
CdpSdk.shared
```

初期化済みの Salesforce CDP Mobile SDK インスタンスの設定を変更することはできませんので注意してください。再び `CdpSdk.configure()` をコールすると、既存のインスタンスは破壊され、新しい Salesforce CDP Mobile SDK インスタンスが作成されます。

```
// Initially
<Cdp.CdpSdk: 0x600001854780>
// After reconfiguring
<Cdp.CdpSdk: 0x600001865da0>
```

ログレベル

ログは任意ですが重要な SDK の機能です。モバイルアプリケーション開発者は、Salesforce CDP Mobile SDK の出力の詳細レベルを選択できます。ログはデフォルトでは無効化されており、`CdpSdk` 共有インスタンスに対して `setLogLevel` メソッドで有効化して、ログレベルとログ出力を定義する必要があります。これにより、ネイティブな統合ログシステムを使用してログが取得されます。

ログパラメータ

```
CdpSdk.setLogLevel(MCLogLevel.debug, logOutputter:  
LogOutputter())
```

パラメータ	型	説明
logLevel	MCLogLevel	必須。ログ出力の詳細レベルを定義します。使用できるオプションは、error、warning、debug、none です。
logOutputter	LogOutputter	必須。出力を定義します。ネイティブな統合ログシステムを使用してログを取得します。

ログレベルの定義

ログレベル	デフォルト	説明
MCLogLevel.error	いいえ	このログレベルは、重要なビジネス使用事例が完了できなくなる、復旧不能なエラーの詳細を提供します。
MCLogLevel.warning	いいえ	このログレベルは、SDKインテグレーションに問題があるか、普段とは異なる状況に遭遇した状況の詳細を提供します。警告は、潜在的に危険ではあるものの復元可能なエラーに関連付けられます。
MCLogLevel.debug	いいえ	省略可能。このログレベルは、SDKがタスク、イベント、エラーの詳細をどのように処理したかに関する詳細な低レベルの情報を提供します。デバッグルогにより、開発者はアプリケーションを診断して問題をトラブルシューティングできます。
trackLifecycle	はい	省略可能。ログを無効化します。

SDK の state プロパティ

Salesforce CDP Mobile SDK の共有インスタンスの **state** プロパティは、現在の設定、セッションの詳細、イベントキューサイズ、同意情報が含まれた JSON を返します。これらは、デバッグとトラブルシューティングで重要な情報となります。

```
CdpSdk.shared.state
```

state プロパティは、開発とトラブルシューティングをサポートする Salesforce CDP Mobile SDK のツールの 1 つです。デバッグを行うときは必ず SDK の state プロパティで設定が想定どおりであることを確認してください。Salesforce CDP Mobile SDK の state プロパティの例:

```
{
    "name": "cdp",
    "config": {
        "screenTrackingEnabled": false,
        "lifecycleEventTrackingEnabled": false,
        "appId": "YOUR APP ID",
        "endpoint": "YOUR ENDPOINT",
        "sessionDuration": 600
    },
    "sessionManager": {
        "sessionId": "SESSION ID"
    },
    "eventManager": {
        "queueSize": 2
    },
    "consentManager": {
        "deviceId": "YOUR DEVICE ID",
        "consent": "opt_in"
    }
}
```

同意管理

EU の一般データ保護規則 (GDPR) やカリフォルニア州消費者プライバシー法 (CCPA)、あるいは会社のプライバシーポリシーなど、同意やプライバシーに関する義務の管理を支援するため、Salesforce CDP Mobile SDK にはイベントデータの収集を許可したり制限したりするための管理ソリューションが用意されています。

モバイルアプリケーションは、デバイスの所有者に対してデータの収集へのオプトインまたはオプトアウトの選択肢を表示し、選択に応じて Salesforce CDP Mobile SDK の consent プロパティを設定する必要があります。

同意の付与

Salesforce CDP Mobile SDK は、モバイルアプリケーションがデータ収集に対するユーザの同意を得るまでは、モバイルデバイスから Salesforce CDP アプリケーションにイベントを送信することができません。下記のように同意を付与することでアプリケーションをクイックスタートできます。

>

アプリケーションを最初に初期化するときに、データの収集に関する同意を付与するようにユーザにお願いします。Salesforce CDP Mobile SDK の state プロパティ (`CdpSdk.shared.state`)を見ると、consent プロパティはデフォルト値の `opt_in` になっています。

```
CdpSdk.shared.consent = Consent.optIn
```

Consent パラメータ

Salesforce CDP Mobile SDK の設定パラメータを下表に示します。

Consent 値	デフォルト	説明
<code>Consent.optIn</code>	いいえ	Consent が <code>optIn</code> に設定されていると、設定されている Salesforce CDP エンドポイントにイベントが送信されて収集されます。
<code>Consent.optOut</code>	いいえ	Consent が <code>optOut</code> に設定されていると、Salesforce CDP Mobile SDK はモバイルアプリケーションによって作成されたイベントを無視します。この場合は、イベントがキューに登録されたりデバイスから送信されたりすることはありません。Consent が <code>optOut</code> に設定された時点でキューに登録されているイベントが存在する場合は、Salesforce CDP Mobile SDK はそれらのイベントをメモリから削除します。
<code>Consent.pending</code>	はい	Salesforce CDP Mobile SDK が新たに初期化された時点では、同意の初期状態は <code>pending</code> (保留中) になります。この状態では、イベントはローカルに収集されますが、CDP には転送されません。Consent が <code>pending</code> から <code>optIn</code> に変更された時点でキューに登録されているイベントが存在する場合は、Salesforce CDP Mobile

Consent 値	デフォルト	説明
		SDKはそれらのイベントをCDPエンドポイントに送信します。

デバイスのユーザから同意情報が得られたら、Salesforce CDP Mobile SDK の consent プロパティを次のように設定します。

```
// grant consent
CdpSdk.shared.consent = Consent.optIn

// revoke consent
CdpSdk.shared.consent = Consent.optOut
```

ロケーションの追跡

Salesforce CDP Mobile SDK では、すべてのイベントでロケーションを追跡できます。この機能を有効化するには、Salesforce CDP Mobile SDK の共有インスタンスで `setLocation` メソッドを使用して、ロケーション座標と有効期限(秒数)を指定します。

```
// prepare the coordinates
let coordinates = Coordinates(latitude: 54.187738,
longitude: 15.554440)

// set the location coordinates and expiration time
CdpSdk.shared.setLocation(coordinates: coordinates,
expiresIn: 60)
```

SDK でロケーション座標の有効期限が残っている間は、イベント属性の `latitude` と `longitude` というキー名のキー-値ペアに、ロケーション属性が自動的に付加されます。

有効期限が過ぎると、SDK ロケーション座標はクリアされ、イベントには付加されなくなります。座標の有効期限が過ぎる前にロケーションの付加を止めたい場合は、`setLocation` メソッドをコールして、座標値として `nil` を指定します。

注意: アプリケーション開発者として、ロケーションを追跡する許可をエンドユーザから得る必要があります。アプリケーション開発者は、ロケーションの追跡許可を求め、ロケーションを取得して、必要な精度範囲を保証する責任を負います。

イベントの追跡

Salesforce CDP Mobile SDK では、イベントを追跡することで、エンドユーザによるモバイルアプリケーションとのインタラクションに関する詳細なデータを収集できます。

イベントは、イベントタイプとイベント属性(省略可能)で構成されます。Salesforce CDP Mobile SDK では、現時点ではエンゲージメントとプロファイルの2つのイベントオブジェクトがサポートされています。

イベントの追跡

エンゲージメントイベントなど、異なるタイプのイベントを作成できます。イベントタイプを作成するには、イベントをインスタンス化して、`eventType`と属性のコレクション(省略可能)を渡します。

```
let event = Event.engagement(eventType: "CartAbandonment",
    attributes:
        ["sku": "COFFEE-NTR-06", "tag price": 19.99])
```

イベントを Salesforce CDP アプリケーションに送信するには、`CdpSdk` 共有インスタンスで `track` メソッドをコールします。

```
CdpSdk.shared.track(event: event)
```

追跡するイベントを確認するには、Salesforce CDP Mobile SDK の `state` プロパティ(`CdpSdk.shared.state`) の `eventManager` セクションで `queueSize` を調べます。

これで Salesforce CDP Mobile SDK の設定が完了し、イベントを作成して、そのイベントを追跡するために Salesforce CDP アプリケーションに送信しました。

Salesforce CDP Mobile SDK 管理ソリューションの Consent は、`pending`、`optIn`、`optOut` という3つの値から構成される列挙型です。Consent は、SDK の初期化時に内部で `pending` に設定されます。SDK を使用するモバイルアプリケーションは、Consent を `optIn` または `optOut` に更新できますが、`pending` に戻すことはできません。

Salesforce CDP Mobile SDK による顧客データの収集に関する同意管理の詳細を下表に示します。

イベントパラメータ

パラメータ	型	説明
<code>eventType</code>	string	必須。予約されている値: <code>AppFirstLaunch</code> 、 <code>AppLaunch</code> 、 <code>AppUpdate</code> 、 <code>ScreenView</code> 。イベントタイプを空白またはスペースのみにすることはできません。
<code>attributes</code>	dictionary	省略可能。イベント属性。いくつかの制限が適用されます。 <ul style="list-style-type: none"> キー

パラメータ	型	説明
		<ul style="list-style-type: none"> - 型: string - 必須: はい - 予約されている値: userId、deviceId、eventId、sessionId、dateTime、eventType、category • 値 <ul style="list-style-type: none"> - サポートされている型: string、int、float、double、bool、NSNull - サポートされている型として認識されない値を指定すると、名前-値ペアが属性のコレクションから削除され、アクションはエラーとして記録されます。

予約されている値と一致する eventType を使用してイベントを作成するとエラーになります。予約されている値と一致する値を使用して属性ディクショナリにキーを追加すると、そのキー-値ペアは削除されます。キー-値ペアにサポートされていない型の値を指定した場合も、キー-値ペアは削除されます。例:

```
// create engagement event
let event = Event.engagement(eventType: "CartAbandonment",
attributes: ["sku": "COFFEE-NTR-06", "tag price": 19.99])

// track engagement event
CdpSdk.shared.track(event: event)

// create engagement event (no attributes)
let simpleEvent = Event.engagement(eventType:
"SortPriceLowToHigh")
CdpSdk.shared.track(event: simpleEvent)

// create profile event
let profileEvent = Event.profile(eventType: "UserUpdate",
attributes: ["name": "John"])

// track profile event
CdpSdk.shared.track(event: profileEvent)
```

第6章

Salesforce CDP メタデータ API

Salesforce CDP メタデータ API の応答には、すべてのエンティティに関するメタデータが含まれます。これには、計算済みインサイト、エンゲージメント、プロファイルなどのエンティティのほか、メタデータと他のオブジェクトのリレーションも含まれます。

GET /api/v1/metadata

計算済みインサイト、エンゲージメント、プロファイル、その他のオブジェクトのメタデータを返します。

構文

HTTP のメソッド

GET

使用可能なバージョン

Salesforce CDP v1.0、Salesforce API バージョン 51.0 以降

形式

REST

URI

/api/v1/metadata/

例

要求

```
http://[TSE].360a.salesforce.com/api/v1/metadata/
```

応答

```
{
  "metadata": [
    {
      "fields": [
        {
          "label": "Name"
        }
      ]
    }
  ]
}
```

```
        "name": "AccountId__c",
        "type": "STRING"
    }
{
    "name": "DataSourceId__c",
    "type": "STRING"
}
{
    "name": "IndividualId__c",
    "type": "STRING"
}
{
    "name": "DataSourceObjectId__c",
    "type": "STRING"
}
{
    "name": "Id__c",
    "type": "STRING"
}
],
{
    "category": "Profile",
    "name": "AccountContact__dlm",
    "relationships": [
        {
            "fromEntity": "AccountContact__dlm",
            "toEntity": "Account__dlm",
            "fromEntityAttribute": "AccountId__c",
            "toEntityAttribute": "Id__c",
            "cardinality": "NTOONE",
        },
        {
            "fromEntity": "AccountContact__dlm",
            "toEntity": "Individual__dlm",
            "fromEntityAttribute": "IndividualId__c",
            "toEntityAttribute": "Id__c",
            "cardinality": "NTOONE"
        },
        {
            "fromEntity": "Case__dlm",
            "toEntity": "AccountContact__dlm",
            "fromEntityAttribute": "AccountContactId__c",
            "toEntityAttribute": "Id__c",
            "cardinality": "NTOONE"
        }
    ]
}
```

HTTP 応答

応答	理由
200 OK	オブジェクトのメタデータ。

第 7 章

Salesforce CDP 取り込み API

トピック:

- [使用開始](#)
- [ストリーミング取り込み](#)
- [一括取り込み](#)

取り込み API は、プログラムでデータを Salesforce CDP に読み込むためのストリーミングインターラクションと一括インターラクションの両方のパターンをサポートする RESTful インターフェースを提供します。データは 1 時間のケイデンスで非同期で処理されます。

一括取り込み API とストリーミング取り込み API の違いは?

使用する API がわからない場合は、データの取得元と、解決しようとしている問題を考慮することをお勧めします。

Bulk API

- 毎日、毎週、または毎月のスケジュールでの大量データの移動。
 - オフピーク時にのみデータをエクスポートできる従来のシステム。
 - 30/60/90/X 日分のデータで埋め戻す必要がある新しい Salesforce CDP 組織。

ストリーミング API

- ほぼリアルタイムで更新する少量のレコードのマイクロバッチ。
 - 現代のストリーミングアーキテクチャに基づくソースシステム。
 - 変更データキャプチャイベント。
 - Web フックからのデータの消費。



例: あなたは Northern Trail Outfitters (NTO) の仕事をしているインテグレータです。

Track My Run モバイルアプリケーションから取得したランナーのプロファイルと活動ログを Salesforce CDP に読み込む作業をしています。マーケティング担当者は、過去 90 日間のプロファイルおよびイベントデータと、その後のすべての新規データおよび更新されたデータを必要としています。

このシナリオでは、データベースアーキテクト (DBA) は過去 90 日間のプロファイルおよび活動をデータベースから CSV ファイルにエクスポートできます。これで、あなたは Bulk API を使用してデータを Salesforce CDP に読み込むことができます。Bulk API の使用は、Salesforce CDP のデータを初期化するための 1 回限りの操作になります。その後、エクスポート以降のこれらのテーブルへの変更と、新しいおよび更新されたプロファイルおよび活動データをストリーミング API

を介してマイクロバッチで同期できます。この更新は、ソースシステムから差分を取得できる限り、頻繁に実行することができます。

使用開始

API を介した Salesforce CDP 組織へのデータの読み込みを開始する前に、前提条件を完了し、認証を設定し、取り込み API に適用される API 制限を把握しておく必要があります。

前提条件

- 取り込み API コネクタを設定する。データを消費するエンドポイントとペイロードを定義します。
- データストリームを作成およびリリースする。取り込みジョブを設定し、外部消費用の API を公開します。
- 接続アプリケーションを設定する。外部アプリケーションが OAuth を使用して Salesforce と統合できるようにします。
- Salesforce OAuth 認証フローから Salesforce CDP アクセストークンを要求する。API 要求の実行に必要なアクセストークンを取得および更新するための OAuth ダンスを実装します。

認証

Salesforce CDP 取り込み API は、他の Salesforce REST API でサポートされるすべての OAuth 2.0 フローをサポートする RESTful API のセットです。API を介してデータを読み込むには、接続アプリケーションに次の範囲が必要です。

- Salesforce CDP 取り込み API データへのアクセスと管理 (cdp_ingest_api)。
- データへのアクセスと管理 (api)。
- ユーザに代わっていつでも要求を実行 (refresh_token, offline_access)。

接続アプリケーションに適切な範囲が設定されたら、Salesforce にアクセストークンを要求し、Salesforce CDP アクセストークンと交換します。

レコードで有効な日付形式

dateTime 項目を指定するには、yyyy-MM-ddTHH:mm:ss.SSSZ 形式を使用します。例: 2021-07-05T09:31:44.457Z

- yyyy は 4 桁の年号
- MM は 2 桁の月 (01 ~ 12)
- dd は 2 桁の日付 (01 ~ 31)
- 「T」はこの後に時刻が記述されることを示す区切り文字
- HH は 2 桁の時間 (00 ~ 23)
- mm は 2 桁の分 (00 ~ 59)
- ss は 2 桁の秒 (00 ~ 59)
- SSS は省略可能な 3 桁のミリ秒 (000 ~ 999)
- 「Z」はタイムゾーンが UTC であることを示す

取り込み API の制限

Item	説明
API の使用制限	各要求の後、アプリケーションで応答コードを確認する必要があります。HTTP 429 Too Many Requests 状況コードは、アプリケーションの要求頻度を減らす必要があることを示しています。
一括ジョブの存続期間	経過日数が 7 日を超えてる未完了 (state = Open/UploadComplete)の一括ジョブは取り込みキューから削除されます。
ジョブごとのファイル	一括ジョブごとに一度に 1 つのファイルをアップロードできます。1 つのジョブで最大 100 ファイルを保持できます。
ペイロードサイズ	Bulk API を介してアップロードされる CSV ファイルの最大サイズは 150 MB です。ストリーミング API を介してアップロードされる JSON データの最大ボディサイズは要求ごとに 200 KB です。

状況コードとエラー応答

HTTP 応答コード	説明
200 OK	要求が正常に完了しました。
201 Created	リソースが正常に作成されたことを示します。
202 Accepted	要求が受諾されました。データは非同期で処理されます。
204 No Content	ジョブが正常に削除されました。
400 Bad Request	クライアントエラーのためサーバが要求を処理できません。原因として、不正な要求構文または無効なリクエストボディが考えられます。
401 Unauthorized	JWT が無効または期限切れのため、認証に失敗しました。トークンを更新してください。
404 Not Found	クライアントエラー: 要求されたリソースが存在しません。
409 Conflict	クライアントエラー: この状況ではジョブの状況を更新できません。

HTTP 応答コード	説明
429 Conflict	一定の時間内にユーザが送信した要求数が多すぎます。バックオフポリシーを実装して要求数を減らしてください。
500 Internal Server Error	内部サーバエラー。要求を再試行してください。

ストリーミング挿入に関するウォーカスルー

このウォーカスルーでは、Bulk API を使用してレコードを読み込む手順について説明します。

一括挿入のウォーカスルー

このウォーカスルーでは、Bulk API を使用してレコードを読み込む手順について説明します。

ストリーミング挿入に関するウォーカスルー

このウォーカスルーでは、Bulk API を使用してレコードを読み込む手順について説明します。

開始する前に、取り込み API の設定に必要な前提条件を完了していることを確認します。

- 取り込み API コネクタを設定する
- データストリームを作成およびリリースする
- 接続アプリケーションを設定する

1. データストリームの一部として定義したソースオブジェクトのスキーマに一致する JSON オブジェクトのコレクションを作成します。コレクションをデータエンベロープでラップします。ペイロードは 200 KB 未満である必要があります。orders.json の例を次に示します。

```
{
  "data": [
    {
      "id": 1,
      "contact_name": "Joe Smith",
      "created_date": "2021-07-22T08:57:08.118Z",
      "tax_exempt": false,
      "ship_address": "9818 Main",
      "total": 77506,
      "tax_rate": 84,
      "modified_date": "2021-07-05T09:31:44.457Z"
    },
    {
      "id": 2,
      "contact_name": "Tina Smith",
      "created_date": "2021-07-08T22:57:02.008Z",
      "tax_exempt": false,
      "ship_address": "245 Main",
      "total": 16986,
      "tax_rate": 61,
      "modified_date": "2021-07-03T06:20:52.886Z"
    }
  ]
}
```

2. Salesforce CDP アクセストークンを要求します。トークン交換応答の `access_token` プロパティには、認証ヘッダーで使用するベアラートークンが含まれます。`instance_url` は、API をホストしている Salesforce CDP インスタンスです。
3. 取り込み要求が正しく設定されていることを検証します。検証に失敗した要求は処理されず、データレイクに取り込まれません。要求の検証が失敗した場合、API からレポートが返され、問題のあるレコードと理由が示されます。検証のフィードバックに基づいて要求を修正し、API から 200 番台の状況コードを受け取るまで再送信します。

```
curl --location --request POST
'https://{{instance_url}}/api/v1/ingest/sources/ecomm/Order/actions/test' \
--header 'Authorization: Bearer {access_token}' \
--header 'Content-Type: application/json' \
--data @orders.json
```

4. インテグレーションが適切に設定されていることを確認したら、ストリーミング API を使用して JSON データをアップロードします。データの処理が受諾されたことを示す 202 Accepted 応答を受信する必要があります。

```
curl --location --request POST 'https://{{instance_url}}/api/v1/ingest/sources/ecomm/Order/' \
\ --header 'Authorization: Bearer {access_token}' \
--header 'Content-Type: application/json' \
--data @orders.json
```

一括插入のウォークスルー

このウォークスルーでは、Bulk API を使用してレコードを読み込む手順について説明します。

開始する前に、取り込み API の設定に必要な前提条件を完了していることを確認します。

- 取り込み API コネクタを設定する
- データストリームを作成およびリリースする
- 接続アプリケーションを設定する

1. 定義したデータストリームに一致するデータを含む CSV ファイルを作成します。例: orders.csv

```
id,contact_name,created_date,tax_exempt,ship_address,total,tax_rate,modified_date
f19cae7e-5cdb-41ce-92ba-539afea04f9d,Jadyn Kozey,2021-07-22T08:57:08.118Z,false,9818
Walter Course Apt. 811,77506,84,2021-07-05T09:31:44.457Z
48c8b6a6-8179-4a15-ba19-ec5c8206ce28,Kristina
Jakubowski,2021-07-08T22:57:02.008Z,false,245 Schowalter Port Suite
245,16986,61,2021-07-03T06:20:52.886Z
```

2. Salesforce CDP アクセストークンを要求します。トークン交換応答の `access_token` プロパティには、認証ヘッダーで使用するベアラートークンが含まれます。`instance_url` は、API をホストしている Salesforce CDP インスタンスです。

3. データをアップロードするには、ジョブを作成します。この例では、ecomm取り込みAPIコネクタからOrdersデータをアップロードします。レコードを削除する場合は、代わりに delete 操作を使用します。ジョブの状況が Open のジョブ ID を含む JSON 応答が返されます。このジョブ ID は次のステップで必要です。

```
curl --location --request POST 'https://{{instance_url}}/api/v1/ingest/jobs' \
--header 'Authorization: Bearer {access_token}' \
--header 'Content-Type: application/json' \
--data-raw '{
  "object": "Orders",
  "sourceName": "ecomm",
  "operation": "upsert"
}、
{
  "object": "Order",
  "id": "913a8b00-7a9e-4331-aaf6-a019ae5021ab",
  "operation": "upsert",
  "sourceName": "ecomm_api",
  "createdById": "005R000000zqc8IAA",
  "createdDate": "2021-08-12T17:06:42.921387Z",
  "systemModstamp": "2021-08-12T17:06:42.921387Z",
  "state": "Open",
  "contentType": "CSV",
  "apiVersion": "v1",
  "contentUrl": "/api/v1/ingest/jobs/913a8b00-7a9e-4331-aaf6-a019ae5021ab/batches"
}'
```

4. CSV データをアップロードします。ジョブを作成したら、データをアップロードできます。前に作成した CSV ファイルを使用してレコードデータを指定します。ジョブの一環として最大 100 個の CSV ファイルを送信できます。各ファイルは 150 MB を超えることはできません。

```
curl --location --request PUT
'https://{{instance_url}}/api/v1/ingest/jobs/913a8b00-7a9e-4331-aaf6-a019ae5021ab/batches
' \
--header 'Authorization: Bearer {access_token}' \
--header 'Content-Type: text/csv' \
--data-binary '@orders.csv'
< HTTP/1.1 201 Created
```

5. アップロードが完了します。データの送信が完了したら、ジョブを終了して Salesforce CDP にジョブの処理準備ができたことを伝達できます。

```
curl -v --location --request PATCH
'https://{{instance_url}}/api/v1/ingest/jobs/913a8b00-7a9e-4331-aaf6-a019ae5021ab' \
--header 'Authorization: Bearer {access_token}' \
--header 'Content-Type: application/json' \
--data-raw '{
  "state": "UploadComplete"
}'
{
  "object": "Order",
  "id": "913a8b00-7a9e-4331-aaf6-a019ae5021ab",
  "operation": "upsert",
  "sourceName": "ecomm_api",
  "createdById": "005R000000zqc8IAA",
```

```

    "createdDate": "2021-08-12T17:06:42.921387Z",
    "systemModstamp": "2021-08-12T17:36:42.921387Z",
    "state": "UploadComplete",
    "contentType": "CSV",
    "apiVersion": "v1",
    "contentUrl": "/api/v1/ingest/jobs/913a8b00-7a9e-4331-aaf6-a019ae5021ab/batches"
}

```

6. ジョブの状況と結果を確認します。全体的なジョブの状況や処理されたレコード数など、ジョブに関する基本的な状況情報を取得するには、ジョブ情報の取得要求を使用します。ファイルがデータレイクに読み込まれるとジョブの状況は UploadComplete から JobComplete または Failed に変わります。

```

curl -v --location --request GET
'https://{{instance_url}}/api/v1/ingest/jobs/913a8b00-7a9e-4331-aaf6-a019ae5021ab' \
--header 'Authorization: Bearer {access_token}' \
{
  "object": "Order",
  "id": "913a8b00-7a9e-4331-aaf6-a019ae5021ab",
  "operation": "upsert",
  "sourceName": "GreasyConnector",
  "createdById": "005R000000zqc8IAA",
  "createdDate": "2021-08-12T17:06:42.921387Z",
  "systemModstamp": "2021-08-12T18:06:42.921387Z",
  "state": "JobComplete",
  "contentType": "CSV",
  "apiVersion": "v1",
  "contentUrl": "/api/v1/ingest/jobs/913a8b00-7a9e-4331-aaf6-a019ae5021ab/batches",
  "retries": 0,
  "totalProcessingTime": 0
}

```

ストリーミング取り込み

Salesforce CDP ストリーミング API では、ファイアアンドフォーゲットパターンを使用して、ソースシステムと Salesforce CDP 間で更新のマイクロバッチをほぼリアルタイムで同期します。

レコードの挿入

Salesforce CDP データレイクテーブルに新規オブジェクトを読み込みます。

レコードの削除

このリソースコレクションで表されるデータレイクテーブルからレコードを削除します。

同期レコード検証

ストリーミング取り込み API では、できるだけ早くデータを取得するためにファイアアンドフォーゲットパターンが使用されます。データは非同期で処理されるため、アップロードが正常に終了するまで最大 1 時間かかる場合があります。

レコードの挿入

Salesforce CDP データレイクテーブルに新規オブジェクトを読み込みます。

構文

URI

```
/api/v1/ingest/sources/{name}/{object-name}
```

適用開始バージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

JSON

HTTP のメソッド

POST

認証

Authorization: Bearer **access_token**

リクエストボディ

```
curl --location --request POST
'https://{{instance_url}}/api/v1/sources/ntotest/exercises_734a9fa2' \
--header 'Authorization: Bearer {access_token}' \
--header 'Content-Type: application/json' \
--data-raw '{
"data": [
{"maid":1,"first_name":"Rachel","last_name":"Rodriquez","email":"rachel@nto.com","gender":"Female","city":"New
York City","state":"NY","created":"2020-04-27T02:48:33Z"}
]
}'
```

要求パラメータ

パラメータ	説明
Content-Type	application/json
name	取り込み API データコネクタの名前。
object-name	Salesforce CDP に送信するリソース種別の名前。

レスポンスボディ

```
{
"accepted": true
}
```

なし。状況コード 202(承諾済み)を返します。これは、要求が承諾され、非同期で処理されることを示します。

レコードの削除

このリソースコレクションで表されるデータレイクテーブルからレコードを削除します。

構文

URI

```
/api/v1/ingest/sources/{name}/{object-name}
```

適用開始バージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

JSON

HTTP のメソッド

DELETE

認証

Authorization: Bearer **access_token**

リクエストボディ

```
curl --location --request DELETE
'https://{{instance_url}}/api/v1/sources/{{name}}/{{object-name}}?ids=001xx000003DGb2AAG,003xx000004TmiQAAS'
\
--header 'Authorization: Bearer {access_token}'
```

要求パラメータ

パラメータ	説明
name	取り込み API データコネクタの名前。
object-name	Salesforce CDP に送信するリソース種別の名前。

レスポンスボディ

```
{
  "accepted": true
}
```

なし。状況コード 202(承諾済み)を返します。これは、要求が承諾され、非同期で処理されることを示します。

同期レコード検証

ストリーミング取り込み API では、できるだけ早くデータを取得するためにファイアアンドフォーゲットパターンが使用されます。データは非同期で処理されるため、アップロードが正常に終了するまで最大1時間かかる場合があります。

この方法では、API インテグレーションに関する即時フィードバックを提供するためにオブジェクトペイロードの同期検証が提供されます。この機能は開発段階での使用のみが意図されています。検証が失敗した要求で

は、要求ペイロードの修正に必要な詳細が API 応答により提供されます。検証を通過した要求では、ペイロードは下流で保持されません。

構文

URI

```
/api/v1/ingest/sources/{name}/{object-name}/actions/test
```

適用開始バージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

JSON

HTTP のメソッド

POST

認証

Authorization: Bearer **access_token**

リクエストボディ

```
curl --location --request POST
'https://{{instance_url}}/api/v1/sources/ntotest/exercises/actions/test' \
--header 'Authorization: Bearer {access_token}' \
--header 'Content-Type: application/json' \
--data-raw '{
"data": [
{"id": "f19cae7e-5cdb-41ce-92ba-539afea04f9d", "contact_name": "Joe Smith", "created_date": "2021-07-22T08:57:08.118Z", "tax_exempt": false, "ship_address": "9818 Main", "total": 77506, "tax_rate": 84, "modified_date": "2021-07-05T09:31:44.457Z"}, {"id": "48c8b6a6-8179-4a15-ba19-ec5c8206ce28", "contact_name": "Tina Smith", "created_date": "2021-07-08T22:57:02.008Z", "tax_exempt": false, "ship_address": "245 Main", "total": 16986, "tax_rate": 61, "modified_date": "2021-07-03T06:20:52.886Z"}
]
}'
```

要求パラメータ

パラメータ	説明
Content-Type	application/json
name	取り込み API データコネクタの名前。
object-name	Salesforce CDP に送信するリソース種別の名前。

レスポンスボディ

```
{
  "timestamp": "2021-08-20T16:34:26.047666Z",
  "error": "Bad Request",
```

```

    "message": "Bad Request: JSON is properly formatted but has a mismatch with
Schema.",
    "path": "uri=/api/v1/ingest/sources/ecomm/Orders/actions/test",
    "tenantId": "{tenantId}",
    "internalErrorCode": "COMMON_ERROR_GENERIC",
    "details": [
        "validationReport": [
            "record": "{\"contact_name\": \"Joe
Smith\", \"total\": 7506, \"tax_exempt\": false, \"id\": \"f19ae7e-5cb-41c-9ba-59afe049d\", \"created_date\": \"2021-07-22T08:57:08.118Z\", \"ship_address\": \"9818
Main\", \"modified_date\": \"2021-07-05T09:31:44.457Z\", \"tax_rate\": 84}",
            "causingExceptions": ["#/id: expected type: Number, found: String"]
        ],
        {
            "record": "{\"contact_name\": \"Tina
Smith\", \"total\": 1006, \"tax_exempt\": false, \"id\": \"48d6a6-8179-4a5b-9ec5-8206e28\", \"created_date\": \"2021-07-08T22:57:02.008Z\", \"ship_address\": \"245
Main\", \"modified_date\": \"2021-07-03T06:20:52.886Z\", \"tax_rate\": 61}",
            "causingExceptions": ["#/id: expected type: Number, found: String"]
        }
    ]
}

```

リクエストボディの検証が失敗した場合、レスポンスボディで validationReport を調べて、失敗したレコードと失敗の理由を確認します。

一括取り込み

Salesforce CDP 一括取り込み API を使用すると、大量のデータセットを更新/挿入または削除できます。CSV ファイルで表現されたアップロードデータを準備し、ジョブを作成して、ジョブデータをアップロードします。残りは Salesforce に処理させます。

CSV ファイルの準備

処理しているオブジェクトの項目名を CSV ファイルの 1 行目でリストします。それ以降の行は、Salesforce CDP データレイクの各レコードに対応します。CSV ファイル内のレコードは、すべて同一オブジェクトのレコードである必要があります。ジョブを最初に作成するときにこのオブジェクトを指定します。

- レコードを作成する場合は、すべての必須項目を含めます。
- ファイルの各項目名ヘッダーはデータソースオブジェクトの項目名と同じである必要があります。結果には、一致した列のみが含まれます。
- レコードの更新は完全置換として機能します。パッチセマンティックはサポートされていません。
- ファイルでは UTF-8 形式を使用します。アップロードデータは 150 MB を超えることはできません。
- CSV ファイルは「RFC 4180 - CSV ファイルの一般書式および MIME タイプ」に従って書式設定されることが予期されています。
- カンマ項目区切り文字のみをサポートします。
- 空の項目値は null に設定されます。

例

```
id,contact_name,created_date,tax_exempt,ship_address,total,tax_rate,modified_date
f19cae7e-5cdb-41ce-92ba-539afea04f9d,Jadyn Kozey,2021-07-22T08:57:08.118Z,false,9818 Walter
Course Apt. 811,77506,84,2021-07-05T09:31:44.457Z
48c8b6a6-8179-4a15-ba19-ec5c8206ce28,Kristina Jakubowski,2021-07-08T22:57:02.008Z,false,245
Schowalter Port Suite 245,16986,61,2021-07-03T06:20:52.886Z
56e005af-cc6c-4f9c-b628-ae6cbbc909cc,Aletha Wilkinson,2021-07-21T04:01:29.838Z,true,7494
Hudson Ports Apt. 123,23907,99,2021-07-18T23:21:43.240Z
```

ジョブの作成

Salesforce CDP にデータを送信するために使用する一括取り込み操作を表すジョブを作成します。ジョブデータはジョブデータのアップロード要求を介して提供されます。この機能の前提条件として Salesforce CDP でデータストリームを作成する必要があります。

ジョブデータのアップロード

処理するジョブの CSV データをアップロードします。ジョブは 1 つ以上のデータファイルで構成できます。すべてのデータがアップロードされたら、ジョブを閉じて、データを処理する準備が整ったことを示す必要があります。

ジョブの終了または中止

ジョブを終了または中止します。ジョブを終了するには、ジョブの状況を `UploadComplete` に更新します。ジョブを終了すると、アップロード済みのデータは処理用のキューに追加されます。ジョブを中止するには、ジョブの状況を `Aborted` に更新します。ジョブを中止すると、そのジョブはキューに追加されず、処理もされません。中止したジョブの一環としてアップロードされたデータは削除されます。

ジョブの削除

ジョブを削除するには、そのジョブの状況が `UploadComplete`、`JobComplete`、`Aborted`、または `Failed` である必要があります。ジョブが削除されると、保存されているジョブデータも削除され、ジョブメタデータ情報が削除されます。

ジョブ情報の取得

指定したジョブに関する詳細情報を取得します。

すべてのジョブの取得

Salesforce CDP のすべてのジョブを取得します。

ジョブの作成

Salesforce CDP にデータを送信するために使用する一括取り込み操作を表すジョブを作成します。ジョブデータはジョブデータのアップロード要求を介して提供されます。この機能の前提条件として Salesforce CDP でデータストリームを作成する必要があります。

この機能の前提条件として Salesforce CDP で [データストリーム](#) を作成する必要があります。

構文

URI

/api/v1/ingest/jobs

適用開始バージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

TEXT/CSV

HTTP のメソッド

POST

認証

Authorization: Bearer **access_token**

リクエストボディ

```
{
  "object": "my_object",
  "sourceName": "my_api_connector",
  "operation": "upsert"
}
```

要求パラメータ

パラメータ	説明
object	処理中のデータのオブジェクト種別。使用するオブジェクト種別はジョブあたり1つのみです。
operation	ジョブの処理操作。有効な値は <i>upsert</i> および <i>delete</i> です。
sourceName	オブジェクトのソースを表すコネクタの名前。

レスポンスボディ

```
HTTP/1.1 201 Created
{
  "id": "4d5a4477-e27f-4bc1-a219-31e79407eade",
  "operation": "upsert",
  "object": "my_object",
  "sourceName": "my_api_connector",
  "createdById": "005R0000000sJ1hIAE",
  "createdDate": "2021-04-05T13:35:31.000Z",
  "systemModstamp": "2021-04-05T13:35:31.000Z",
  "state": "Open",
  "contentType": "CSV",
  "apiVersion": v1,
  "contentUrl": "api/v1/ingest/jobs/4d5a4477-e27f-4bc1-a219-31e79407eade/batches"
}
```

ジョブデータのアップロード

処理するジョブの CSV データをアップロードします。ジョブは1つ以上のデータファイルで構成できます。すべてのデータがアップロードされたら、ジョブを閉じて、データを処理する準備が整ったことを示す必要があります。

構文

URI

/api/v1/ingest/jobs/{id}/batches

適用開始バージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

TEXT/CSV

HTTP のメソッド

PUT

認証

Authorization: Bearer **access_token**

リクエストボディ

レコードデータを含む CSV ファイルを含めます。

要求パラメータ

パラメータ	説明
id	ジョブの作成要求からレスポンスボディに返されたジョブ ID。

リクエストボディ

なし。状況コード 202(承諾済み) を返します。このコードは、Salesforce によってジョブデータが正常に受信されたことを示します。

例

```
curl --location --request PUT 'https://{{instance_url}}/api/v1/ingest/jobs/{id}/batches' \
--header 'Content-Type: text/csv' \
--header 'Authorization: Bearer {access_token}' \
--data @type_name-1.csv

HTTP/1.1 202 Accepted
```

ジョブの終了または中止

ジョブを終了または中止します。ジョブを終了するには、ジョブの状況を *UploadComplete* に更新します。ジョブを終了すると、アップロード済みのデータは処理用のキューに追加されます。ジョブを中止するには、ジョブの状況を *Aborted* に更新します。ジョブを中止すると、そのジョブはキューに追加されず、処理もされません。中止したジョブの一環としてアップロードされたデータは削除されます。

ジョブが正常に終了または中止されると、ジョブデータを追加できなくなります。

構文

URI

/api/v1/ingest/jobs/{id}

適用開始バージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

JSON

HTTP のメソッド

PATCH

認証

Authorization: Bearer **access_token**

リクエストボディ

```
{
    "state" : "UploadComplete"
}
```

要求パラメータ

パラメータ	説明
id	ジョブの作成要求からレスポンスボディに返されたジョブ ID。
state	ジョブの更新後の状況。ジョブを終了する場合は <i>UploadComplete</i> 、ジョブを中止する場合は <i>Aborted</i> を使用します。

レスポンスボディ

```
{
    "id": "7505e000001NLkPAAW",
    "operation": "upsert",
    "object": "Contact",
    "createdById": "0055e000003PfMYAA0",
    "createdDate": "2021-06-10T13:25:40.000Z",
    "systemModstamp": "2021-06-10T13:25:40.000Z",
    "state": "UploadComplete",
    "contentType": "CSV",
    "apiVersion": "v1"
}
```

例

```
curl --location --request PATCH 'https://{{instance_url}}/api/v1/ingest/jobs/{id}' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer <access_token>' \
--data-binary '{
    "state": "UploadComplete"
}'
```

```
{
}'

< HTTP/1.1 200 OK
< Content-Type: application/json; charset=UTF-8
< Transfer-Encoding: chunked
<
{"id": "00000000000000000000000000000000", "status": "UploadComplete", "object": "Customer", "dateCreated": "2016-01-25T00:00:00Z", "dateLastModified": "2016-01-25T00:00:00Z", "lastModifiedDate": "2016-01-25T00:00:00Z", "stage": "Upload", "stageIndex": 1, "version": 1}
```

ジョブの削除

ジョブを削除するには、そのジョブの状況がUploadComplete、JobComplete、Aborted、またはFailedである必要があります。ジョブが削除されると、保存されているジョブデータも削除され、ジョブメタデータ情報が削除されます。

構文

URI

/api/v1/ingest/jobs/{id}

適用開始バージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

JSON

HTTP のメソッド

DELETE

認証

Authorization: Bearer *access_token*

リクエストボディ

不要。

要求パラメータ

パラメータ	説明
id	ジョブの作成要求からレスポンスボディに返されたジョブ ID。

レスポンスボディ

なし。状況コード 204(コンテンツがありません)を返します。このコードは、ジョブが正常に削除されたことを示します。

例

```
curl --location --request DELETE \
https://'{instance_url}/api/v1/ingest/jobs/{id}' \
--header 'Authorization: Bearer {access_token}'
```

```
HTTP/1.1 204 No Content
```

ジョブ情報の取得

指定したジョブに関する詳細情報を取得します。

構文

URI

```
/api/v1/ingest/jobs/{id}
```

適用開始バージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

JSON

HTTP のメソッド

GET

認証

```
Authorization: Bearer access_token
```

リクエストボディ

```
curl --location --request GET  
'https://{{instance_url}}/api/v1/ingest/jobs/4d5a4477-e27f-4bc1-a219-31e79407eade' \  
--header 'Authorization: Bearer {access_token}'
```

要求パラメータ

パラメータ	説明
id	ジョブの作成要求からレスポンスボディに返されたジョブ ID。

レスポンスボディ

```
HTTP/1.1 200 OK
{
  "id": "4d5a4477-e27f-4bc1-a219-31e79407eade",
  "operation": "upsert",
  "object": "runners__c",
  "createdById": "005R0000000sJ1hIAE",
  "createdDate": "2021-04-08T14:29:53.000Z",
  "systemModstamp": "2021-04-08T14:29:53.000Z",
  "state": "Open",
  "contentType": "CSV",
  "apiVersion": 1,
  "contentUrl": "api/v1/ingest/jobs/4d5a4477-e27f-4bc1-a219-31e79407eade/batches",
  "retries": 0,
```

```

        "totalProcessingTime": 0
    }
}
```

すべてのジョブの取得

Salesforce CDP のすべてのジョブを取得します。

構文

URI

/api/v1/ingest/jobs

適用開始バージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

JSON

HTTP のメソッド

GET

認証

Authorization: Bearer **access_token**

リクエストボディ

```
curl --location --request GET 'https://{{instance_url}}/api/v1/ingest/jobs/' \
--header 'Authorization: Bearer {access_token}'
```

要求パラメータ

パラメータ	説明
limit	返すレコードの数。デフォルトは 20 です。要求ごとに最大 100 レコード。
offset	返す前にスキップする行数。
orderby	ジョブ定義結果の並び替えの基準となる項目。デフォルトの並び替え基準項目は systemModstamp です。
state	特定の状況になっているジョブを取得します。有効な状況は Open、UploadComplete、Failed、Aborted、JobComplete です。カンマ区切りリストにすることができます。

レスポンスボディ

```
HTTP/1.1 200 OK
{
  "data": [
    {
      "id": "4d5a4477-e27f-4bc1-a219-31e79407eade",
      "operation": "upsert",
      "status": "Open"
    }
  ]
}
```

```
"object": "MyCdpObject",
"createdById": "005R0000000sJ1hIAE",
"createdDate": "2021-04-08T14:29:53.000Z",
"systemModstamp": "2021-04-08T14:29:53.000Z",
"state": "Open",
"contentType": "CSV",
"apiVersion": "v1",
"contentUrl": "api/v1/ingest/jobs/4d5a4477-e27f-4bc1-a219-31e79407eade/batches"
}, {
    "id": "9d5a4477-e27f-4bc1-a219-31e79407eade",
    ...
}]
}
```

第 8 章

Salesforce CDP MuleSoft コネクタの実装

MuleSoft の機能とデータコネクタのエコシステムを利用して、データを Salesforce CDP に取り込みます。Salesforce CDP MuleSoft コネクタは、外部システムとの Salesforce CDP インスタンスの統合に役立ちます。取り込み API を使用すると、サードパーティからのデータの取り込みを自動化して、既存の Salesforce CDP レコードを強化できます。

1. [Salesforce CDP 取り込み API](#) (ページ 98) の一括取り込みまたはストリーミング取り込みを設定します。
2. [MuleSoft コネクタ](#)を設定します。

第9章

Amazon S3 データストリームの作成

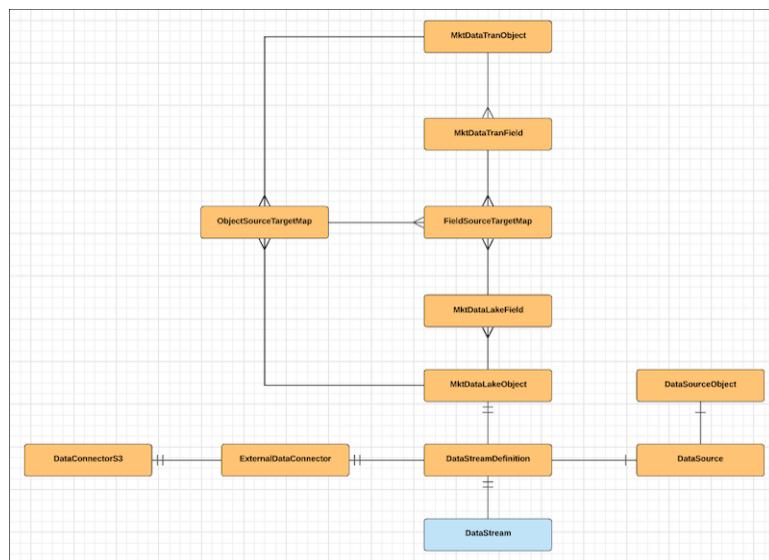
トピック:

- Post
[/services/data/v50.0/lightningrecords](#)

プログラムで Amazon S3 データソースからデータをストリーミングするためのデータストリームを作成します。データストリームにより、Salesforce CDP データレイクにデータを取り込むことができます。

1. ローカルクライアントアプリケーションでメタデータを定義します。

次の図は、各オブジェクトと他のオブジェクトとの関連を示しています。



2. package.xml を使用して ZIP ファイルを作成します。
3. `deploy()` コールを使用して、メタデータパッケージをローカルファイルシステムから Salesforce CDP に移動します。データストリーム定義 ID を POST 要求で使用できるようにメモしておきます。

メモ: POST 要求の作成後に Salesforce CDP アプリケーションの Amazon S3 ログイン情報を更新するか、またはデータコネクタ S3 XML メタデータにログイン情報を含めます。

4. データストリームを作成するための UI API の POST 要求を作成します。

関連トピック:

[Salesforce ヘルプ: クラウドストレージデータストリームの作成](#)

[DataStreamDefinition](#)

[DataSource](#)

[MktDataTranObject](#)

[ExternalDataConnector](#)

[ObjectSourceTargetMap](#)

[DataConnectorS3](#)

[ユーザインターフェース API の使用開始](#)

Post /services/data/v51.0/ui-api/records

POST メソッドを使用して、Amazon S3 データソースからデータをストリーミングします。

構文

HTTP のメソッド

POST

使用可能なバージョン

510

形式

REST

LIRI

/services/data/v51.0/ui-api/records

例

要求

<https://{{TSE}}.salesforce.com/services/data/v51.0/ui-api/records>

心答

```
{  
    "apiName": "DataStream",  
    "fields": {  
        "Name": "{DATA STREAM NAME}",  
        "DataStreamDefinitionId" : "{DATA STREAM DEFINITION ID}",  
        "RefreshFrequency": "Hourly",  
        "RefreshMode" : "Upsert"  
    }  
}
```

HTTP 忔答

```
{  
    "apiName": "DataStream",  
    "childRelationships": {},  
    "eTag": "9e9cb5c9a3e9db2e75918fa9b5b20815",  
    "fields": {  
        "CreatedBy": {  
            "displayValue": "Admin User",  
            "value": {  
                "id": "1",  
                "label": "Admin User",  
                "value": "1"  
            }  
        }  
    }  
}
```

```
"apiName": "User",
"childRelationships": {},
"eTag": "8941f8da5501cf307f51e59c936e434b",
"fields": {
    "Id": {
        "displayValue": null,
        "value": "005R0000000d9SfIAI"
    },
    "Name": {
        "displayValue": null,
        "value": "Admin User"
    }
},
"id": "005R0000000d9SfIAI",
"lastModifiedById": "005R0000000d9SfIAI",
"lastModifiedDate": "2021-03-24T19:03:56.000Z",
"recordTypeId": null,
"recordTypeInfo": null,
"systemModstamp": "2021-03-31T22:47:39.000Z",
"weakEtag": 1617230859000
},
"CreatedById": {
    "displayValue": null,
    "value": "005R0000000d9SfIAI"
},
"CreatedDate": {
    "displayValue": "4/2/2021, 2:50 PM",
    "value": "2021-04-02T21:50:23.000Z"
},
"DataStreamStatus": {
    "displayValue": "Processing",
    "value": "PROCESSING"
},
"ImportRunStatus": {
    "displayValue": "None",
    "value": "NONE"
},
"IsRefreshDeliveryHourAutomatic": {
    "displayValue": null,
    "value": false
},
"LastNumberOfRowsAddedCount": {
    "displayValue": null,
    "value": null
},
"LastRefreshDate": {
    "displayValue": null,
    "value": null
},
"Name": {
    "displayValue": null,
    "value": "s3-ui-api"
},
```

```
"RefreshDayOfMonth": {  
    "displayValue": null,  
    "value": null  
},  
"RefreshDayOfWeek": {  
    "displayValue": null,  
    "value": null  
},  
"RefreshFrequency": {  
    "displayValue": "Hourly",  
    "value": "HOURLY"  
},  
"RefreshHours": {  
    "displayValue": null,  
    "value": null  
},  
"RefreshMode": {  
    "displayValue": "Upsert",  
    "value": "UPSERT"  
},  
"TotalNumberOfRowsAdded": {  
    "displayValue": null,  
    "value": null  
},  
"TotalRowsProcessed": {  
    "displayValue": null,  
    "value": null  
}  
},  
"id": "1dsR000000002zzIAA",  
"lastModifiedById": "005R0000000d9SfIAI",  
"lastModifiedDate": "2021-04-02T21:50:23.000Z",  
"recordTypeId": "012000000000000AAA",  
"recordTypeInfo": null,  
"systemModstamp": "2021-04-02T21:50:23.000Z",  
"weakEtag": 1617400223000  
}
```

第 10 章

Salesforce CDP Profile API ライブラリ

トピック:

- GET /api/v1/profile/metadata
- GET /api/v1/profile/metadata/{dataModelName}
- GET /api/v1/profile/{dataModelName}
- GET /api/v1/profile/{dataModelName}/{dataModelName}
- GET /api/v1/profile/{dataModelName}/{dataModelName}/{dataModelName}
- GET /api/v1/profile/{dataModelName}/{dataModelName}/{dataModelName}/{dataModelName}

Salesforce CDP Profile API コールは、顧客のプロファイル情報の参照と検索に使用します。これらの API コールを外部の Web/モバイルアプリケーションに含めることで、顧客のプロファイル情報を参照できます。

Salesforce CDP Profile API コールを使用すると、計算済みエンゲージメントインサイトと共にプロファイル情報を取得して、完全な Salesforce CDP ページを構築できます。

これらのコールでは、プロファイルカテゴリ内のデータモデルオブジェクトのみがサポートされます。これには、個人、連絡先メール、統合個人、連絡先住所などがあります。また、[個人] 項目に関連した計算済み属性も取得できます。たとえば、顧客プロファイルから過去 6か月間の売上合計を計算して取得できます。

これらのコールでは、エンゲージメントカテゴリのデータモデルオブジェクト(販売注文など)はサポートされません。

Salesforce CDP Profile API コールは、REST 標準仕様に準拠します。

主なプロパティ

- 取得する項目を明示的に指定しないと、エンティティの項目や属性がアルファベット順に並び替えられて先頭の 10 件のデータのみが応答に含まれます。この 10 件という制限は、項目名を明示的に指定する場合には適用されません。
- 最大で 50 件の項目を取得できます。
- 論理演算子の AND と比較演算子の等号のみがサポートされます。
- クエリで使用できる検索条件の数に制限はありません。
- 返される行の上限はデフォルトで 100 行に設定されています。最大で 4999 行を返すことができます。
- 日付と時刻の項目では次の形式を使用してください。

```
Date/DateTime BirthDate__c=1990-07-12 00:00:00
```

コールリファレンス

GET /api/v1/profile/metadata

GET /api/v1/profile/metadata/{dataModelName}

GET /api/v1/profile/{dataModelName}
GET /api/v1/profile/{dataModelName}/{Id}
GET /api/v1/profile/{dataModelName}/{Id}/{childDataModelName}
GET /api/v1/profile/{dataModelName}/{Id}/calculated-insights/{ci-name}

接続アプリケーションの設定

顧客に関する新しいインサイトを見つけるように接続アプリケーションを設定します。

Salesforce CDP のアクセストークンの取得と交換

アクセストークン (ページ 4) を取得して交換します。

トラブルシューティング

「User hasn't approved this consumer yet (ユーザがまだコンシューマを承認していません)」のエラーメッセージが返された場合は、次の設定パラメータを確認してください。

- 接続アプリケーションの設定で [ユーザは自己承認可能] が有効化されていることを確認します。
- refresh_token範囲のほかに、1つ以上の(カスタム範囲以外の)範囲が選択されていることを確認します。
- 自己承認 URL で、scope パラメータ値が api refresh_token cdpquery cdpprofile に設定されていることを確認します。
- 自己承認 URL に対して最初にコールを実行したときには、URL で要求されている各範囲への権限を求める同意ページに応答する必要があります。

関連トピック:

プロファイル、クエリ、計算済みインサイトに関する Salesforce CDP API の制限
プロファイル、クエリ、計算済みインサイトに関する Salesforce CDP API の制限

GET /api/v1/profile/metadata

データモデルオブジェクト、それぞれの項目、カテゴリのリストを返します。

構文

HTTP のメソッド

GET

使用可能なバージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

REST

URI

/api/v1/profile/metadata

要求パラメータ

なし。

例

要求

```
https://{{TSE}}.360a.salesforce.com/api/v1/profile/metadata
```

応答

```
{
  "metadata": [
    {
      "indexes": [
        {
          "fields": [
            {
              "name": "FirstName__c",
              "type": "STRING"
            }
          ]
        }
      ],
      "name": "Individual__dlm",
      "category": "Profile",
      "displayName": "Individual",
      "fields": [
        {
          "name": "BirthDate__c",
        }
      ]
    }
  ]
}
```

```

        "displayName": "Birth Date",
        "type": "DATE"
    },
    {
        "name": "FirstName__c",
        "displayName": "First Name",
        "type": "STRING"
    }
],
"relationships": [
    {
        "fromEntity": "ContactPointEmail__dlm",
        "toEntity": "Individual__dlm",
        "fromEntityAttribute": "PartyId__c",
        "toEntityAttribute": "Id__c",
        "cardinality": "NTOONE"
    }
],
"primaryKeys": [
    {
        "name": "Id__c",
        "displayName": "Individual Id",
        "indexOrder": "1"
    }
]
}
]
```

HTTP 応答

応答	理由
200 OK	コールバックが正常に検証されました。
400 Bad Request	無効な要求です。
403 Forbidden	承認の受け渡しに失敗しました。
404 Not Found	コールバックIDまたは検証キー、あるいはその両方が正しくありません。
422 Unprocessable Entity	項目/検索条件パラメータで使用されている属性名が存在しません。
500 Server Error	内部エラーです。

GET /api/v1/profile/metadata/{dataModelName}

データモデルオブジェクトのメタデータを返します。メタデータには、ルックアップで使用できる項目、データ型、インデックスのリストが含まれます。

構文

HTTP のメソッド

GET

使用可能なバージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

REST

URI

/api/v1/profile/metadata/{dataModelName}

要求パラメータ

項目名	項目種別	説明
dataModelName	string	必須。データモデルオブジェクトの API 参照名。例: UnifiedIndividual_dlm

例

要求

```
https://{{TSE}}.360a.salesforce.com/api/v1/profile/metadata/{{dataModelName}}
```

応答

```
{
  "metadata": [
    {
      "indexes": [
        {
          "fields": [
            {
              "name": "FirstName__c",
              "type": "STRING"
            }
          ]
        }
      ],
    }
  ],
}
```

```

    "name": "Individual__dlm",
    "category": "Profile",
    "displayName" : "Individual",
    "fields": [
        {
            "name": "BirthDate__c",
            "displayName": "Birth Date",
            "type": "DATE"
        },
        {
            "name": "FirstName__c",
            "displayName" : "First Name",
            "type": "STRING"
        }
    ],
    "relationships" : [
        {
            "fromEntity" : "ContactPointEmail__dlm",
            "toEntity" : "Individual__dlm",
            "fromEntityAttribute" : "PartyId__c",
            "toEntityAttribute" : "Id__c",
            "cardinality" : "NTOONE"
        }
    ],
    "primaryKeys" : [
        {
            "name" : "Id__c",
            "displayName" : "Individual Id",
            "indexOrder" : "1"
        }
    ]
}
}

```

HTTP 応答

応答	理由
200 OK	コールバックが正常に検証されました。
400 Bad Request	無効な要求です。
403 Forbidden	承認の受け渡しに失敗しました。
404 Not Found	コールバックIDまたは検証キー、あるいはその両方が正しくありません。
422 Unprocessable Entity	項目/検索条件パラメータで使用されている属性名が存在しません。
500 Server Error	内部エラーです。

GET /api/v1/profile/{dataModelName}

検索条件に基づいてデータモデルオブジェクトを返します。この API コールは、選択された(1つ以上の)検索条件を適用して絞り込んだオブジェクトレコードを取得します。

構文

HTTP のメソッド

GET

使用可能なバージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

REST

URI

/api/v1/profile/{dataModelName}

要求パラメータ

項目名	項目種別	説明
dataModelName	string	必須。データモデルオブジェクトの API 参照名。例: UnifiedIndividual__dlm
fields	string	結果に含める個別の項目名のカンマ区切りリスト。このパラメータを指定しないと、任意の項目のセットが返されます。例: Id__c,FirstName__c,GenderId__c,Occupation__c
filters	string	必須。各括弧内に指定した等式のカンマ区切りリスト。例: [FirstName__c=DON]
limit	int	返すレコード数。例: 10
offset	int	結果を返す最初の行までスキップする行数。例: 100 この項目は、API バージョン 52.0 以降で使用できます。
orderby	string	結果セットを昇順または降順に並び替えるための値のカンマ区切りリスト。例: GenderId__c ASC,Occupation__c DESC この項目は、API バージョン 52.0 以降で使用できます。

例

要求

```
https://{{TSE}}.360a.salesforce.com/api/v1/profile/{dataModelName}
```

応答

```
{
  "data": [
    {
      "GenderId__c": "F",
      "Id__c": "05d01a06-0c61-3990-808d-2aa602dd3e7f",
      "Occupation__c": "Staffing",
      "FirstName__c": "Don"
    },
    {
      "GenderId__c": "M",
      "Id__c": "11bcc846-1a80-3293-8cc4-3449835d8d4c",
      "Occupation__c": "Operational",
      "FirstName__c": "Don"
    },
    {
      "GenderId__c": "M",
      "Id__c": "18213566-df86-34a8-97aa-2cb75b70ae3b",
      "Occupation__c": "Organizational",
      "FirstName__c": "Don"
    }
  ]
}
```

HTTP 応答

応答	理由
200 OK	コールバックが正常に検証されました。
400 Bad Request	無効な要求です。
403 Forbidden	承認の受け渡しに失敗しました。
404 Not Found	コールバック ID または検証キー、あるいはその両方が正しくありません。
422 Unprocessable Entity	項目/検索条件パラメータで使用されている属性名が存在しません。
500 Server Error	内部エラーです。

GET /api/v1/profile/{dataModelName}/{Id}

インデックスと検索条件に基づいてデータモデルオブジェクトを返します。このAPIコールは、プライマリキーまたはセカンダリキーに基づいてオブジェクトレコードを取得します。レコードが見つからない場合は空のリストを返します。

構文

HTTP のメソッド

GET

使用可能なバージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

REST

URI

/api/v1/profile/{dataModelName}/{Id}

要求パラメータ

項目名	項目種別	説明
dataModelName	string	必須。データモデルオブジェクトの API 参照名。例: UnifiedIndividual_dlm
fields	string	省略可能。結果に含める個別の項目名のカンマ区切りリスト。このパラメータを指定しないと、任意の項目のセットが返されます。例: Id__c, FirstName__c, GenderId__c, Occupation__c
filters	string	各括弧内に指定した等式のカンマ区切りリスト。例: [FirstName__c=DON]
Id	string	必須。プライマリ/セカンダリキー項目の値。指定しなかった場合のデフォルトはプライマリキーになります。例: John
limit	int	返すレコード数。例: 10
offset	int	結果を返す最初の行までスキップする行数。例: 100 この項目は、API バージョン 52.0 以降で使用できます。
orderby	string	結果セットを昇順または降順に並び替えるための値のカンマ区切りリスト。例: GenderId__c ASC, Occupation__c DESC この項目は、API バージョン 52.0 以降で使用できます。
searchKey	string	プライマリキー以外を使用する場合のキー項目の名前。例: FirstName__c

例

要求

```
https://{{TSE}}.360a.salesforce.com/api/v1/profile/{dataModelName}/{Id}
```

応答

```
{
  "data": [
    {
      "GenderId__c": "F",
      "Id__c": "05d01a06-0c61-3990-808d-2aa602dd3e7f",
      "Occupation__c": "Staffing",
      "FirstName__c": "Don"
    },
    {
      "GenderId__c": "M",
      "Id__c": "11bcc846-1a80-3293-8cc4-3449835d8d4c",
      "Occupation__c": "Operational",
      "FirstName__c": "Don"
    },
    {
      "GenderId__c": "M",
      "Id__c": "18213566-df86-34a8-97aa-2cb75b70ae3b",
      "Occupation__c": "Organizational",
      "FirstName__c": "Don"
    }
  ]
}
```

HTTP 応答

応答	理由
200 OK	コールバックが正常に検証されました。
400 Bad Request	無効な要求です。
403 Forbidden	承認の受け渡しに失敗しました。
404 Not Found	コールバックIDまたは検証キー、あるいはその両方が正しくありません。
422 Unprocessable Entity	項目/検索条件パラメータで使用されている属性名が存在しません。
500 Server Error	内部エラーです。

GET /api/v1/profile/{dataModelName}/{id}/{childDataModelName}

インデックスと検索条件に基づいてデータモデルオブジェクトと子オブジェクトを返します。

構文

HTTP のメソッド

GET

使用可能なバージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

REST

URI

/api/v1/profile/{dataModelName}/{id}/{childDataModelName}

要求パラメータ

項目名	項目種別	説明
childDataModelName	string	必須。子データモデルオブジェクトの API 参照名。例: UnifiedContactPointEmail_dlm
dataModelName	string	必須。データモデルオブジェクトの API 参照名。例: UnifiedIndividual_dlm
fields	string	子エンティティから返される結果に含める項目名のカンマ区切りリスト。指定しなかった場合は、アルファベット順に並び替えられた項目の最初の 10 件が返されます。例: Id_c,EmailAddress_c
filters	string	検索条件は親オブジェクトにのみ適用されます。例: [FirstName_c=Mary]
id	string	必須。インデックスが付けられた列の値。
limit	int	返される項目の数。例: 10
offset	int	結果を返す最初の行までスキップする行数。例: 100 この項目は、API バージョン 52.0 以降で使用できます。
orderby	string	結果セットを昇順または降順に並び替えるための値のカンマ区切りリスト。例: GenderId_c ASC,Occupation_c DESC この項目は、API バージョン 52.0 以降で使用できます。
searchKey	string	オブジェクト ID がプライマリキーと異なる場合のインデックス名。例: FirstName_c

例

要求

```
https://[TSE].360a.salesforce.com/api/v1/profile/{dataModelName}/{Id}/{childDataModelName}
```

応答

```
{
  "data": [
    [
      {
        "GenderId__c": "F",
        "ExternalRecordId__c": "003f200002a3213541",
        "UnifiedContactPointEmail__dlm": [
          {
            "EmailAddress__c": "mstubblefeld@gmail.marketingclouddemo.com",
            "Id__c": "c0ed30ac-c33c-3388-8197-8e7000529478"
          }
        ],
        "MaritalStatusId__c": "Other",
        "LastName__c": "Stubblefield",
        "ChildrenCount__c": 2,
        "Id__c": "05d01a06-0c61-3990-808d-2aa602dd3e7f",
        "BirthDate__c": "1963-07-08T00:00:00",
        "Occupation__c": "Staffing",
        "InternalOrganizationId__c": "",
        "FirstName__c": "Don"
      }
    ]
  ]
}
```

HTTP 応答

応答	理由
200 OK	コールバックが正常に検証されました。
400 Bad Request	無効な要求です。
403 Forbidden	承認の受け渡しに失敗しました。
404 Not Found	コールバックIDまたは検証キー、あるいはその両方が正しくありません。

応答	理由
422 Unprocessable Entity	項目/検索条件パラメータで使用されている属性名が存在しません。
500 Server Error	内部エラーです。

GET /api/v1/profile/{dataModelName}/{id}/calculated-insights/{ci-name}

インデックスと検索条件に基づいてデータモデルオブジェクトと計算済みビューを返します。

構文

HTTP のメソッド

GET

使用可能なバージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

REST

URI

/api/v1/profile/{dataModelName}/{id}/calculated-insights/{ci-name}

要求パラメータ

項目名	項目種別	説明
ci_name	string	必須。計算済みインサイトの名前。例: IndividualChildrenCount_cio
dataModelName	string	必須。データモデルオブジェクトの API 参照名。例: UnifiedIndividual_dlm
dimensions	string	計算済みビューから返される結果に含めるディメンション名のカンマ区切りリスト。デフォルトではすべてのディメンションが含まれられます。例: Id_c
filters	string	検索条件は親オブジェクトにのみ適用されます。例: [FirstName_c=Angel]
id	string	必須。インデックスが付けられた列の値。例: 003726a3-061d-33e7-a5a5-96ecae866a61
limit	int	返される項目の数。例: 10

項目名	項目種別	説明
measures	string	計算済みビューから返される結果に含める基準名のカンマ区切りリスト。デフォルトではすべての基準が含まれられます。例: TotalSales__c
offset	int	結果を返す最初の行までスキップする行数。例: 100 この項目は、API バージョン 52.0 以降で使用できます。
orderby	string	結果セットを昇順または降順に並び替えるための値のカンマ区切りリスト。例: GenderId__c ASC,Occupation__c DESC この項目は、API バージョン 52.0 以降で使用できます。
searchKey	string	オブジェクトIDがプライマリキーと異なる場合のインデックス名。例: FirstName__c

例

要求

```
https://{{TSE}}.360a.salesforce.com/api/v1/profile/{dataModelName}/{Id}/calculated-insights/{ci-name}
```

応答

```
{
  "data": [
    {
      "GenderId__c": "Male",
      "DataSourceId__c": "Customers",
      "Sales_And_Discounts_By_Customer__cio": {
        "TotalDiscounts__c": "0.600000000000000100",
        "Id__c": "233280554",
        "Product": "Multi grain bread"
      },
      "MaritalStatusId__c": "Widowed",
      "LastName__c": "Kathman",
      "ChildrenCount__c": "1",
      "Id__c": "233280554",
      "BirthDate__c": "1962-08-08T00:00:00",
      "DataSourceObjectId__c": "Individual_Id",
      "InternalOrganizationId__c": "",
      "FirstName__c": "Angel"
    },
    {
      "GenderId__c": "Male",
      "DataSourceId__c": "Customers",
      "Sales_And_Discounts_By_Customer__cio": {
        "TotalDiscounts__c": "0.600000000000000100",
        "Id__c": "233280554",
        "Product": "Multi grain bread"
      }
    }
  ]
}
```

```

    "Sales_And_Discounts_By_Customer_cio": {
        "TotalDiscounts__c": "0.6000000000000000100",
        "Id__c": "233280554",
        "Product": "Yogurt"
    },
    "MaritalStatusId__c": "Widowed",
    "LastName__c": "Kathman",
    "ChildrenCount__c": "1",
    "Id__c": "233280554",
    "BirthDate__c": "1962-08-08T00:00:00",
    "DataSourceObjectId__c": "Individual_Id",
    "InternalOrganizationId__c": "",
    "FirstName__c": "Angel"
}
]
}

```

HTTP 応答

応答	理由
200 OK	コールバックが正常に検証されました。
400 Bad Request	無効な要求です。
403 Forbidden	承認の受け渡しに失敗しました。
404 Not Found	コールバックIDまたは検証キー、あるいはその両方が正しくありません。
422 Unprocessable Entity	項目/検索条件パラメータで使用されている属性名が存在しません。
500 Server Error	内部エラーです。

第 11 章

Salesforce CDP Query API ライブラリ

トピック:

- [POST /api/v1/query](#)

この API ファミリは、v1 の同期クエリと ANSI 標準の SQL クエリのみをサポートします。データストリーム、プロファイル/エンゲージメントデータモデルオブジェクト、統合データモデルオブジェクトなどのオブジェクトを使用した自由形式の SQL を使用できます。この API は、データの抽出、外部アプリケーションの統合、データレイクに対するインタラクティブな照会など、さまざまな使用事例をサポートします。

主なプロパティ

- 1回のコールで返される最大行数は、デフォルトで最大 4999 行に制限されています。done フラグを使用して他のレコードが存在するかどうかを確認します。
- SQL コール長には明示的な制限はありません。
- ネストされたサブクエリ数には明示的な制限はありません。
- 結合の数には明示的な制限はありません。
- 検索条件の数には明示的な制限はありません。
- 列投影の数には明示的な制限はありません。
- VARCHAR、DECIMAL、TIMESTAMP など、特定のデータ型には明示的な制限はありません。
- ページネーションは、limit、offset、および orderby パラメータでサポートされます。
 - 応答内で done が true としてフラグ付けされている場合、クエリする他のレコードはありません。応答が false の場合、offset パラメータを使用して orderby をコールし、次の数値セットを取得します。

接続アプリケーションの設定

顧客に関する新しいインサイトを見つけるように接続アプリケーションを設定します。

Salesforce CDP のアクセストークンの取得と交換

アクセストークン (ページ 4) を取得して交換します。

コールリファレンス

POST /api/v1/query

関連トピック:

[プロファイル、クエリ、計算済みインサイトに関する Salesforce CDP API の制限](#)

POST /api/v1/query

この API は、データモデルオブジェクト、レイクオブジェクト、統合オブジェクト、リンクされたオブジェクト全体で Salesforce CDP データレイクを照会します。

構文

HTTPのメソッド

POST

使用可能なバージョン

Salesforce CDP v1.0, Salesforce v51.0

形式

REST

URI

/api/v1/query

要求パラメータ

項目名	項目の型	説明
limit	int	返すレコード数。例: 10
offset	int	結果を返す最初の行までスキップする行数。例: 100
orderby	string	結果セットを昇順または降順に並び替えるための値のカンマ区切りリスト。例: GenderId__c ASC,Occupation__c DESC

例

要求

```
        {
            "sql": "select BirthDate__c, DataSourceId__c, DataSourceObjectId__c,
ExternalRecordId__c, FirstName__c, GenderId__c, Id__c, InternalOrganizationId__c,
LastName__c from Individual__dlm where Id__c='100470169'"
        }
```

心管

```
{  
    "data": [  
        {  
            "Id__c": "100470169",  
            "GenderId__c": "M",
```

```
"FirstName__c": "Joshua",
"LastName__c": "Carrier",
"BirthDate__c": "1994-01-21T00:00:00",
"ExternalRecordId__c": "003f200002a3227163",
"DataSourceId__c": "SubscribedUsers",
"InternalOrganizationId__c": "",
"DataSourceObjectId__c": "Subscribers_for_CDP"
},
],
"startTime": "2020-12-19T06:50:05.251Z",
"endTime": "2020-12-19T06:50:05.251Z",
"rowCount": 1,
"queryId": "3442db1a-ecef-42b3-b0d8-971ee459efa4"
}
```

HTTP 応答

応答	理由
200 OK	コールバックが正常に検証されました。
400 Bad Request	無効な要求です。
403 Forbidden	承認の受け渡しに失敗しました。
404 Not Found	コールバック ID または検証キー、あるいはその両方が正しくありません。
422 Unprocessable Entity	項目/検索条件パラメータで使用されている属性名が存在しません。
500 Server Error	内部エラーです。

第 12 章

Salesforce CDP 計算済みインサイト API ライブライ

トピック:

- GET
[/api/v1/insight/metadata/\(ci-name\)](#)
- GET
[/api/v1/insight/calculated-insights/\(ci-name\)](#)

計算済みインサイト API コールは、Salesforce CDP 計算済みインサイトオブジェクトから取得したデータコレクションのみをサポートします。この API を使用すると、選択した SQL ディメンションと基準を含む算済みインサイトを照会できます。検索条件も使用できます(省略可能)。

この API を使用すると、選択した SQL ディメンションと基準を含む算済みインサイトを照会できます。検索条件も使用できます(省略可能)。

Salesforce CDP ユーザインターフェース内から計算済みインサイトコールを使用する方法については、オンラインヘルプトピックの「[CalculatedInsights\(計算済みインサイト\)](#)」を参照してください。

この API ライブライは、標準 REST 仕様に準拠しています。

主なプロパティ

- Salesforce CDP 計算済みインサイト API ライブライは、最大 5 件の基準と最大 10 件のディメンションをサポートします。
- 1 回のコールで返される最大行数は、デフォルトで最大 4999 行に制限されています。
- ページネーションでは、パラメータによって制限、オフセット、順序を指定できます。

コールリファレンス

GET [/api/v1/insight/metadata/\(ci-name\)](#)

GET [/api/v1/insight/calculated-insights/\(ci-name\)](#)

GET /api/v1/insight/metadata/{ci-name}

メタデータには、計算済みインサイトの一部であるディメンションと基準が含まれます。

構文

HTTP のメソッド

GET

使用可能なバージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

REST

URI

/api/v1/insight/metadata/{ci-name}

要求パラメータ

項目名	項目種別	説明
ci_name	string	必須。計算済みインサイトの名前。例: IndividualChildrenCount__cio

例

要求

```
https://{{TSE}}.360a.salesforce.com/api/v1/insight/metadata/{{ci-name}}
```

応答

```
{
  "metadata": [
    {
      "name": "Avg_Spends__cio",
      "dimensions": [
        {
          "name": "Id__c",
          "type": "STRING"
        },
        {
          "name": "FirstName__c",
          "type": "STRING"
        }
      ],
    }
  ]
},
```

```

  "measures": [
    {
      "name": "Avg_Spend__c",
      "type": "NUMBER"
    }
  ],
}
]
}

```

HTTP 応答

応答	理由
200 OK	コールバックが正常に検証されました。
400 Bad Request	無効な要求です。
403 Forbidden	承認の受け渡しに失敗しました。
404 Not Found	コールバック ID または検証キー、あるいはその両方が正しくありません。
422 Unprocessable Entity	項目/検索条件パラメータで使用されている属性名が存在しません。
500 Server Error	内部エラーです。

GET /api/v1/insight/calculated-insights/(ci-name)

この API は、計算済みインサイトを照会するために使用します。異なるディメンション、基準、検索条件を選択することで、スライス、ダイス、絞り込みを行うことができます。

構文

HTTP のメソッド

GET

使用可能なバージョン

Salesforce CDP v1.0、Salesforce v51.0

形式

REST

URI

/api/v1/insights/calculated-insight/(ci-name)

要求パラメータ

項目名	項目種別	説明
ci_name	string	必須。計算済みインサイトの名前。
dimensions	string	投影するディメンションのカンマ区切りのリスト。デフォルトでは、このパラメータには使用可能なすべてのディメンションが含まれます。例: GenderId__c
measures	string	投影する基準のカンマ区切りのリスト。デフォルトでは、このパラメータには使用可能なすべての基準が含まれます。例: TotalSales__c
limit	int	返される項目の数。デフォルトでは、最大値である 4999 件の項目が返されます。例: 1000
offset	int	結果を返す最初の行までスキップする行数。例: 100
filters	string	このパラメータをクエリに追加することで、特定のデータ型の結果のみを返すことができます。例: [GenderId__c=Male, FirstName__c=Angel]
orderby	string	結果セットを昇順または降順に並び替えるための値のカンマ区切りリスト。例: GenderId__c ASC, Occupation__c DESC

例

要求

```
https://{{TSE}}.360a.salesforce.com/api/v1/insights/calculated-insights/{{ci-name}}
```

応答

```
{
  "data": [
    {
      "GenderId__c": "F",
      "Id__c": "05d01a06-0c61-3990-808d-2aa602dd3e7f",
      "Occupation__c": "Staffing",
      "FirstName__c": "Don"
    },
    {
      "GenderId__c": "M",
      "Id__c": "11bcc846-1a80-3293-8cc4-3449835d8d4c",
      "Occupation__c": "Operational",
      "FirstName__c": "Don"
    },
    {
      ...
    }
  ]
}
```

```
        "GenderId__c": "M",
        "Id__c": "18213566-df86-34a8-97aa-2cb75b70ae3b",
        "Occupation__c": "Organizational",
        "FirstName__c": "Don"
    }
]
```

HTTP 応答

応答	理由
200 OK	コールバックが正常に検証されました。
400 Bad Request	無効な要求です。
403 Forbidden	承認の受け渡しに失敗しました。
404 Not Found	コールバックIDまたは検証キー、あるいはその両方が正しくありません。
422 Unprocessable Entity	項目または検索条件パラメータで使用されている属性名が存在しません。
500 Server Error	内部エラーです。