CADDE 4.0 導入ガイド

第3編

データ利用者環境導入編

2023年3月版

2023年3月

作成者：　SIP分野間データ連携基盤　合同コンソーシアム



© NATIONAL INSTITUTE OF INFORMATICS, © 2023 NTT DATA Corporation, © JIP Techno Science Corporation　All rights reserved.

© Hitachi, Ltd. 2023 , © SB Technology Corp., © The University of Tokyo , © NEC Corporation 2023 , Copyright 2023 FUJITSU LIMITED. All rights reserved.

変更来歴

| # | 版数 | 発行年月日 | 変更内容 |
| --- | --- | --- | --- |
|  | 初版 | 2022/09/30 | 新規作成 |
|  | 2023年3月版 | 2023/03/31 | 3.1.2. 契約管理へのアカウント登録申請の追加、や、5. データ取得APの開発のAPIとAPI利用コメンドの記載例の改定　等 |
|  |  |  |  |

* はじめに

ガイド本文に入る前に、「CADDE4.0 導入ガイド(コネクタ環境)」、「CADDE4.0 運用ガイド(支援サービス群)」について、以下の表0-1に各ガイドの位置づけ、想定読者、ドキュメント体系等を示す。

まずは以下を確認したうえで、自身が読むべき適切なドキュメントを参照されたい。

表0-1 導入ガイド、運用ガイドの概要説明



目次

[1. 全体手順 2](#_Toc129792470)

[1.1. CADDEシステムの概要と本ガイドの範囲（データ利用者が準備すべき機能） 2](#_Toc129792471)

[1.2. CADDEシステムパターン 3](#_Toc129792472)

[1.3. データ利用者環境のCADDE導入ステップの概要 4](#_Toc129792473)

[1.4. CADDE設定変更フロー 5](#_Toc129792474)

[2. CADDE導入事前準備 6](#_Toc129792475)

[2.1. 前提環境の導入 6](#_Toc129792476)

[2.2. 通信系の準備 7](#_Toc129792477)

[2.2.1. プロキシの設置、URL取得 7](#_Toc129792478)

[2.2.2. TLSクライアント証明書発行申請 7](#_Toc129792479)

[3. CADDE導入作業 8](#_Toc129792480)

[3.1. CADDE利用に関する申請 8](#_Toc129792481)

[3.1.1. CADDE利用申請 8](#_Toc129792482)

[3.1.2. 契約管理へのアカウント登録申請 10](#_Toc129792483)

[3.2. 利用者コネクタの準備 11](#_Toc129792484)

[3.2.1. 利用者コネクタの導入 12](#_Toc129792485)

[3.2.2. 横断検索サービスとの接続設定 14](#_Toc129792486)

[3.2.3. 認証サービスとの接続設定 15](#_Toc129792487)

[3.2.4. ロケーションサービスとの接続設定 16](#_Toc129792488)

[3.2.5. 来歴機能との接続設定ガイド 17](#_Toc129792489)

[4. CADDEの動作確認 18](#_Toc129792490)

[4.1. 動作確認 18](#_Toc129792491)

[4.1.1. データカタログの検索や交換の動作確認 18](#_Toc129792492)

[4.1.2. 認証認可も含むコネクタの動作確認 18](#_Toc129792493)

[4.1.3. エラー発生時の切り分け方法 19](#_Toc129792494)

[5. データ取得APの開発 22](#_Toc129792495)

[5.1. 利用者コネクタのAPI一覧リスト 22](#_Toc129792496)

[5.1.1. 認証に関するAPI 22](#_Toc129792497)

[5.1.2. 利用者コネクタの利用に関するAPI 22](#_Toc129792498)

[5.1.3. 来歴管理に関するAPI 24](#_Toc129792499)

[5.2. バッチAPとしての利用 25](#_Toc129792500)

[5.3. データ取得のユースケース別スクリプト一覧（簡易パターン） 25](#_Toc129792501)

[5.4. データ取得のユースケース別スクリプト一覧（基本・市場利用パターン） 26](#_Toc129792502)

[5.5. データ取得のユースケース別スクリプト一覧（連続取得パターン） 29](#_Toc129792503)

# 全体手順

## CADDEシステムの概要と本ガイドの範囲（データ利用者が準備すべき機能）

CADDEシステムは、図 1.1‑1の機能コンポーネントからなるデータ連携基盤である。本ガイドは赤太枠で囲まれたデータ利用者が準備すべき機能の導入手順について記載している。ガイドの各項目では、その作業の前提となる条件や作業の完了条件を記載している。

ダイアグラム

自動的に生成された説明

図 1.1‑1 CADDEシステム全体像

## CADDEシステムパターン

CADDEのシステムパターンは表 1.2‑1に示す通り、基本パターン、簡易パターン、市場利用パターンの3種類があり、利用パターンによって導入する機能が異なる。各システムパターンの特徴や機能の詳細については、導入ガイド\_システムパターン編に記載している。

表 1.2‑1 CADDEシステム利用パターンと導入機能・設定テーブル

自動的に生成された説明

## データ利用者環境のCADDE導入ステップの概要

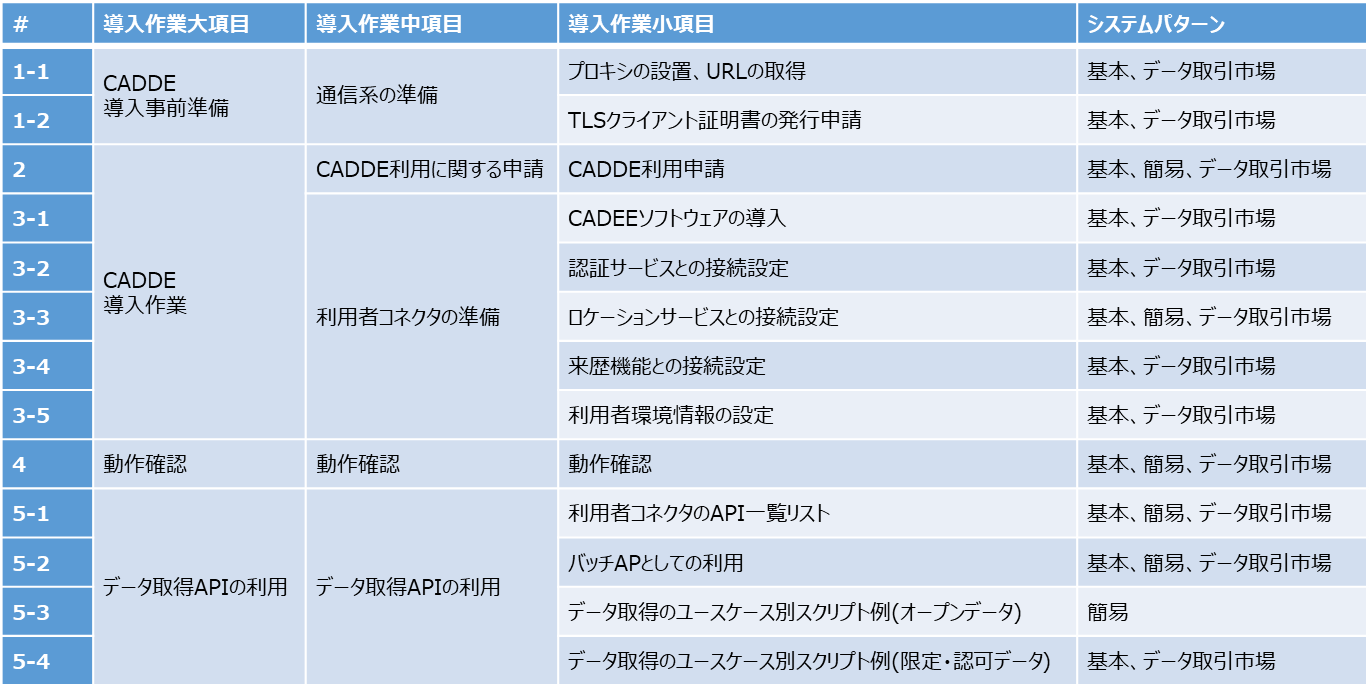
CADDEの導入には、事前にCADDEを導入するための環境を構築する必要がある。システムパターン共通の導入フローを図 1.3‑1　導入フローに示す。導入フロー内で実施する詳細作業は表 1.3‑1の導入作業小項目に記載してある。詳細作業はシステムパターンごとに必要なものが異なる。システムパターンの詳細は、導入ガイドの第1編を参照のこと。

ダイアグラム が含まれている画像

自動的に生成された説明

図 1.3‑1　導入フロー

表 1.3‑1 導入作業詳細



## CADDE設定変更フロー

CADDEを導入後、設定変更を実施する際には、SIP分野間に申請を行い、コンフィグ情報をもらったのち、コネクタの設定を変更する。設定変更後には、導入時と同様に動作確認環境を用いて稼働確認を実施することが必要である。

ダイアグラム が含まれている画像

自動的に生成された説明

図 1.4‑1 設定変更フロー

表 1.4‑1設定変更作業詳細

# CADDE導入事前準備

## 前提環境の導入

CADDE提供者コネクタ等のデータ提供者環境を構築するために、前提環境を導入する必要がある。  
前提環境とは図 2.1‑1の赤枠に示された機能群である。

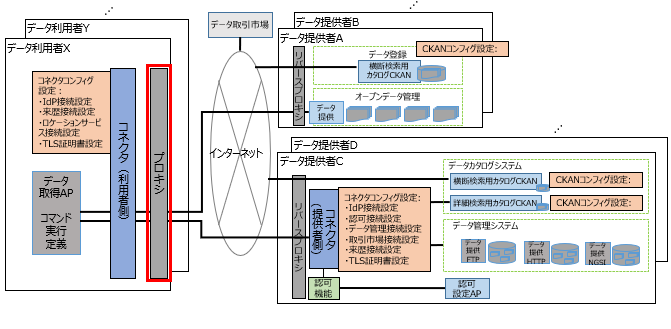


図 2.1‑1 CADDEのデータ利用者環境を構築するために前提環境（赤字枠）

データ利用者環境における前提環境の機能群を、表 2.1‑1で説明する。

表 2.1‑1　CADDEのデータ利用者環境を構築するために前提環境となるシステムの概説

|  |  |  |
| --- | --- | --- |
| # | システム名 | システムの機能概説 |
| 1 | プロキシ | 利用者コネクタへのアクセスを受付し、提供者コネクタとのTLS認証を行う。 |

## 通信系の準備

### プロキシの設置、URL取得

利用者コネクタから提供者コネクタへデータ取得を行う場合は、通信路で提供者コネクタとTLS認証を行う必要がある。

対象システムパターン：基本、市場利用

#### 導入前確認事項

プロキシを導入するサーバが準備されていること。

準備するサーバは下記の条件を満たしていること

・Docker、Docker Compose が事前インストールされていること。

・対応する Docker Version は以下の通りとする。

Docker 20.10.1

・対応する OS は、Linux の上記 Docker がサポートする OS とする。

#### 導入手順

GitHubよりプロキシ(Squid)を取得する。その後リバースプロキシを設置したサーバURLを取得する。

シェルで下記コマンドを流すことでプロキシを取得することができる。

git clone https://github.com/CADDE-sip/connector

その後、プロキシサーバのURLの設定、取得を行う。

#### 完了条件

プロキシを設置したサーバURLを取得できていること。

### TLSクライアント証明書発行申請

対象システムパターン：基本、市場利用

#### 導入前確認事項

公開鍵と秘密鍵のペア、及びCSRが作成済みであること

#### 導入手順

CADDE　TLS認証局から「TLS電子証明書発行申請書」受け取り、必要事項を記入しCADDE　TLS認証局へ提出する。CADDE　TLS認証局の審査を経て、発行されたTLS電子証明書を受け取る

#### 完了条件

CADDE　TLS認証局からTLS電子証明書を受け取っていること

# CADDE導入作業

## CADDE利用に関する申請

### CADDE利用申請

対象システムパターン：基本、簡易、市場利用

#### 導入前確認事項

2.CADDE導入事前準備の手順が全て完了していること。

#### 導入手順

・コネクタを新規導入した場合

CADDE総合受付よりCADDE利用申請書を取得する。申請書に必要事項を記入しCADDE総合受付へ提出する。受付後にコンフィグファイルへ設定する情報（CADDEユーザID、コネクタID、コネクタのシークレット、支援サービス群の認証情報）をCADDE総合受付より受領する。

※申請書の記載内容については表 3.1‑1に記載している。

表 3.1‑1 CADDE利用申請書の記載項目



#### 完了条件

CADDE総合受付にCADDE利用申請書を提出し、コンフィグファイルへ設定する情報がCADDE総合受付より送付されてきていること。

### 契約管理へのアカウント登録申請

対象システム利用パターン：市場利用

#### 導入前確認事項

2.CADDE導入事前準備の作業が全て完了していること。

#### 導入手順

契約管理にアカウント登録を実施する。登録方法の詳細は利用する契約管理サービスに確認する。

#### 完了条件

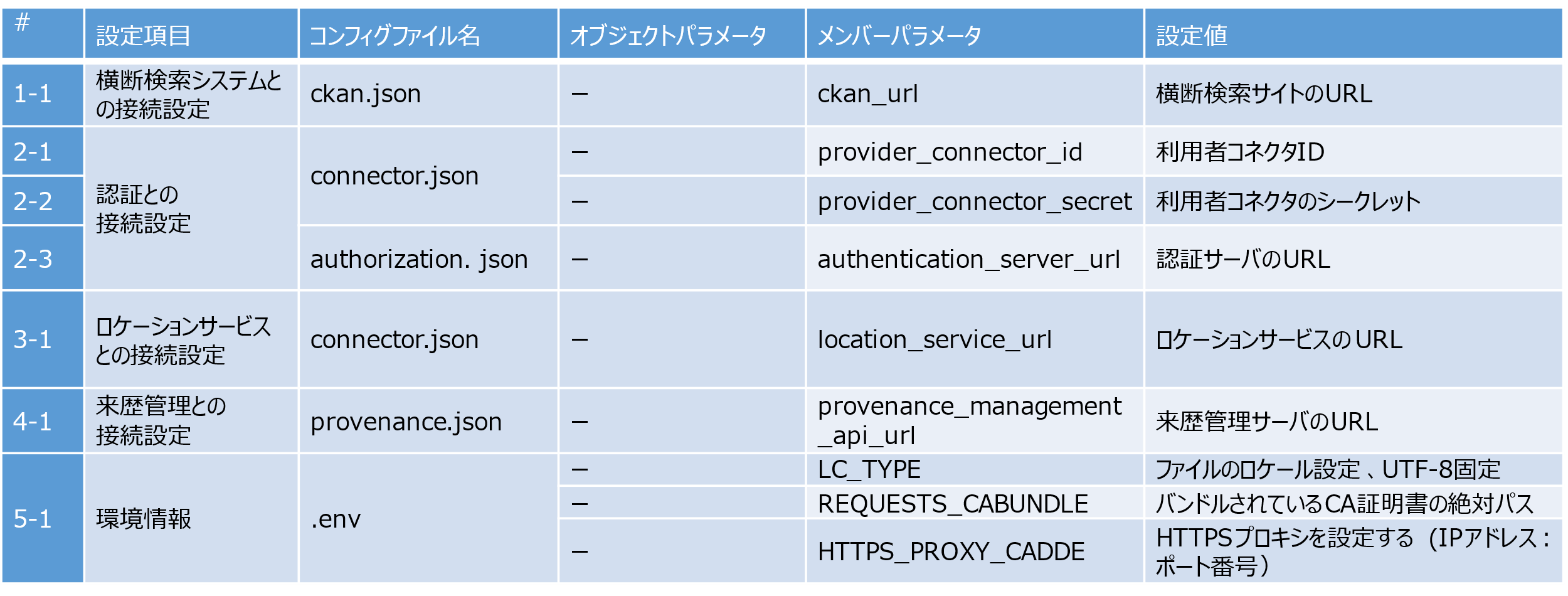
契約管理より登録完了通知が届き、契約管理サービスを利用できるようになっていること。

## 利用者コネクタの準備

利用者コネクタを利用するには、GitHubよりソフトウェアを入手し、設定を行う必要がある。設定ファイルとパラメータの一覧表 3.2‑1に示す。設定方法の詳細は各項にて説明する。(黄色背景はCADDE利用申請の際に受付窓口から受領する値、他は提供者自身で決めるもしくはGitHub記載の値) なお、設定ファイルを修正/更新した際には、コネクタを再起動することにより反映される。

支援サービスのURLが設定値となっている部分は、支援サービス一覧台帳として支援サービスの運営者から公開されているものを参照し、該当支援サービスのURLを記載する。

表 3.2‑1　コネクタへの設定項目の一覧

****

### 利用者コネクタの導入

対象システムパターン：基本、簡易、市場利用

#### 導入前確認事項

2.CADDE導入事前準備の作業が全て完了していること。

下記を満たす環境があること

OS：Linux

Docker、Docker Compose が事前インストールされていることを前提とする。

対応する Docker Version は以下の通り。

Docker 20.10.1

対応する OS は、Linux の上記 Docker がサポートする OS

#### 導入手順

提供者コネクタは、コネクタ導入環境のシェル上でコマンドを入力、実行することで、GitHub上よりコネクタを取得し、インストールが実施される。実行するコマンドの詳細はGitHubの提供者コネクタ構築手順(https://github.com/CADDE-sip/connector#%E6%8F%90%E4%BE%9B%E8%80%85%E3%82%B3%E3%83%8D%E3%82%AF%E3%82%BF%E7%92%B0%E5%A2%83%E6%BA%96%E5%82%99)の1.提供者コネクタの取得、2. setup.sh実行を参照すること。

#### 完了条件

利用者コネクタがコネクタ導入環境にダウンロードされている。

#### 補足情報

コネクタ導入作業において、Dockerコンテナ群の設定変更(docker-compose.ymlを編集して、各コンテナの利用ポートの変更)を行わない場合、図 3.2‑1の構成となる。

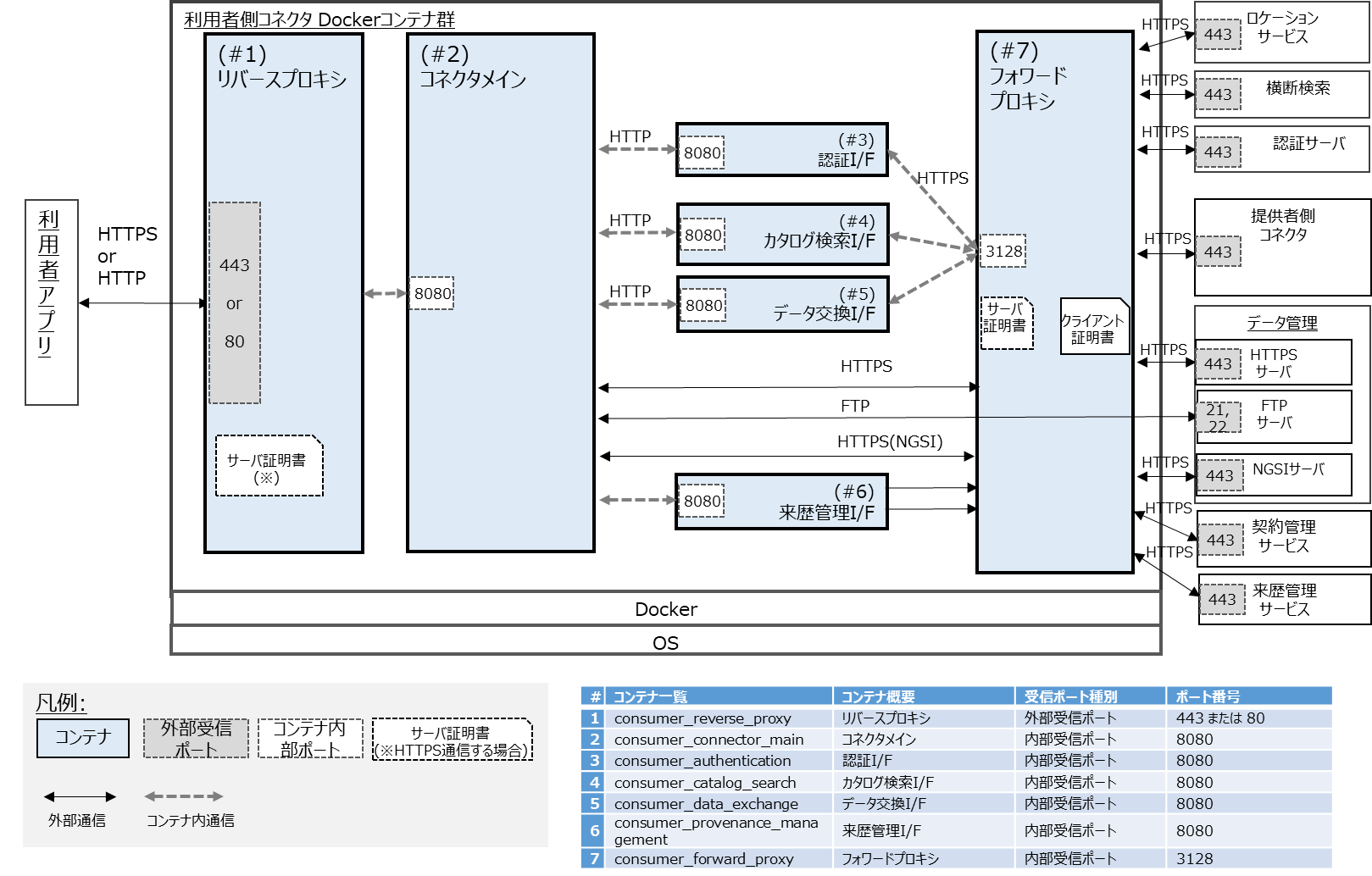


図 3.2‑1 利用者コネクタ環境の構成例

### 横断検索サービスとの接続設定

対象システムパターン：基本、簡易、市場利用

#### 導入前確認事項

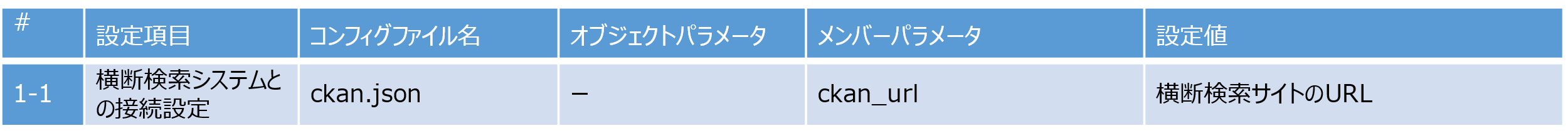
3.2.利用者コネクタの準備の手順が全て完了していること。

#### 導入手順

横断検索サービスとの接続を行うには、CKAN の横断検索時の接続先URLを記載する必要がある。

具体的には、GitHubの利用者コネクタ構築手順(https://github.com/CADDE-sip/connector#%E5%88%A9%E7%94%A8%E8%80%85%E3%82%B3%E3%83%8D%E3%82%AF%E3%82%BF%E6%A7%8B%E7%AF%89%E6%89%8B%E9%A0%86)の3. コンフィグファイルの設定を参照し、ckan.jsonの下記1項目を設定する必要がある。

表 3.2‑2 コネクタへの設定項目（横断検索サービス）



#### 完了条件

ckan.json内の下記1項目が設定されていること

・ckan\_url 横断検索時の横断検索サーバのURLを記載

### 認証サービスとの接続設定

対象システムパターン：基本、簡易、市場利用

#### 導入前確認事項

3.2.利用者コネクタの準備の手順が全て完了していること。

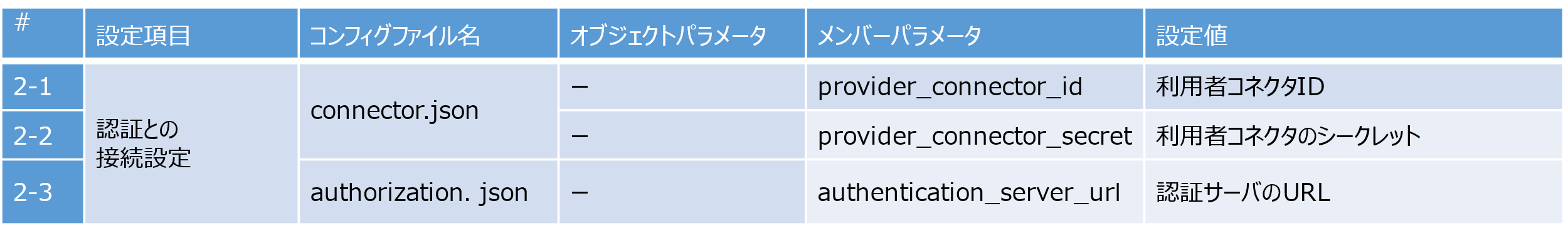
#### 導入手順

認証サービスを利用するには、認可サーバのURL、利用者コネクタのID、シークレット、認証サーバのURLを記載する必要があり。他の値は固定値が入力されているため設定不要。

具体的には、GitHubの利用者コネクタ構築手順(https://github.com/CADDE-sip/connector#%E5%88%A9%E7%94%A8%E8%80%85%E3%82%B3%E3%83%8D%E3%82%AF%E3%82%BF%E6%A7%8B%E7%AF%89%E6%89%8B%E9%A0%86)の3. コンフィグファイルの設定を参照し、

connector.json内の下記2項目と、authentication.jsonの下記1項目を設定する必要がある。

表 3.2‑3 コネクタへの設定項目（認証サービス）



#### 完了条件

connector.json内の下記2項目と、authentication.jsonの下記1項目が設定されていること。

connector.jsonの設定項目

・consumer\_connector\_id　利用者コネクタのID

・consumer\_connector\_secret　利用者コネクタのシークレット

authentication.jsonの設定項目

・authentication\_server\_url　認証サーバのURL

### ロケーションサービスとの接続設定

対象システムパターン：市場利用

#### 導入前確認事項

3.2.利用者コネクタの準備の手順が全て完了していること。

#### 導入手順

提供者コネクタから、ロケーションサービスのURLを設定する必要がある。

具体的には、GitHubの提供者コネクタ構築手順(https://github.com/CADDE-sip/connector#%E6%8F%90%E4%BE%9B%E8%80%85%E3%82%B3%E3%83%8D%E3%82%AF%E3%82%BF%E7%92%B0%E5%A2%83%E6%BA%96%E5%82%99)の3-(5) 認証および認可をおこなう場合 を参照し、connector. json内の下記1項目を設定する。

表 3.2‑4 コネクタへの設定項目（ロケーションサービス）



#### 完了条件

connector. json内の下記1項目が設定されていること。

・location\_service\_url ロケーションサービスのURL

### 来歴機能との接続設定ガイド

対象システムパターン：基本、市場利用

#### 導入前確認事項

3.2.利用者コネクタの準備の手順が全て完了していること。

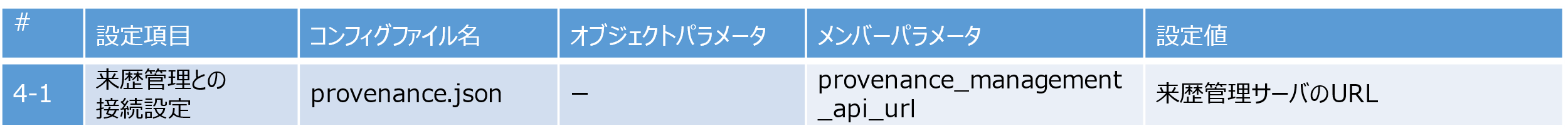
#### 導入手順

来歴機能との接続を行うには、来歴管理サーバのURLを記載する必要がある。

具体的には、GitHubの利用者コネクタ構築手順(https://github.com/CADDE-sip/connector#%E5%88%A9%E7%94%A8%E8%80%85%E3%82%B3%E3%83%8D%E3%82%AF%E3%82%BF%E6%A7%8B%E7%AF%89%E6%89%8B%E9%A0%86)の3. コンフィグファイルの設定を参照し、

provenance.jsonの下記1項目を設定する必要がある。

表 3.2‑5 コネクタへの設定項目（来歴機能）



#### 完了条件

provenance.jsonの下記1項目が設定されていること。

・provenance\_management\_api\_url　来歴管理サーバのURL

# CADDEの動作確認

## 動作確認

CADDEの導入作業が完了した後に、導入した機能が正しく動いているか確認する必要がある。動作確認は総合試験の環境を利用する。

なお、CADDEコネクタはdockerを用いて作成されているため動作確認時にログを確認したい際は下記のコマンドでログを見ることができる。

* docker compose logs -f

対象システムパターン：基本、簡易、市場利用

### データカタログの検索や交換の動作確認

#### 事前確認事項

3.CADDE導入作業が完了していること

#### 確認手順

1. 導入したCADDE利用者コネクタを用いて総合試験環境上のデータカタログを検索する。
2. 検索したデータカタログを取得する

#### 完了条件

データカタログを検索・取得できること

### 認証認可も含むコネクタの動作確認

#### 事前確認事項

3.CADDE導入作業が完了していること

#### 確認手順

1. 総合試験環境のデータ提供者環境のデータを取得できるように、認可サーバに今回構築したCADDE利用者IDを認可サーバへ認可登録しておく。
2. 構築した利用者コネクタを用いて総合試験環境の提供者コネクタへ、利用者コネクタ利用ガイド(https://github.com/CADDE-sip/connector/blob/master/doc/ConsumerManual.md)の(1-3) ファイル取得を参考に提供者コネクタへファイル取得のコマンドを実行する。
3. 正常にファイルを取得できたことを確認する。

#### 完了条件

構築した利用者コネクタで総合試験環境のデータ提供者環境のデータを取得できていること。

### エラー発生時の切り分け方法

#### 通信路を含むインフラ環境の確認方法

詳細検索またはデータ取得コマンド実行時にエラーが発生した場合、利用者コネクタ及び提供者コネクタ双方のログを確認し、ログの出力有無で原因箇所を切り分けることが可能である。イメージを図 4.1‑1に示す。

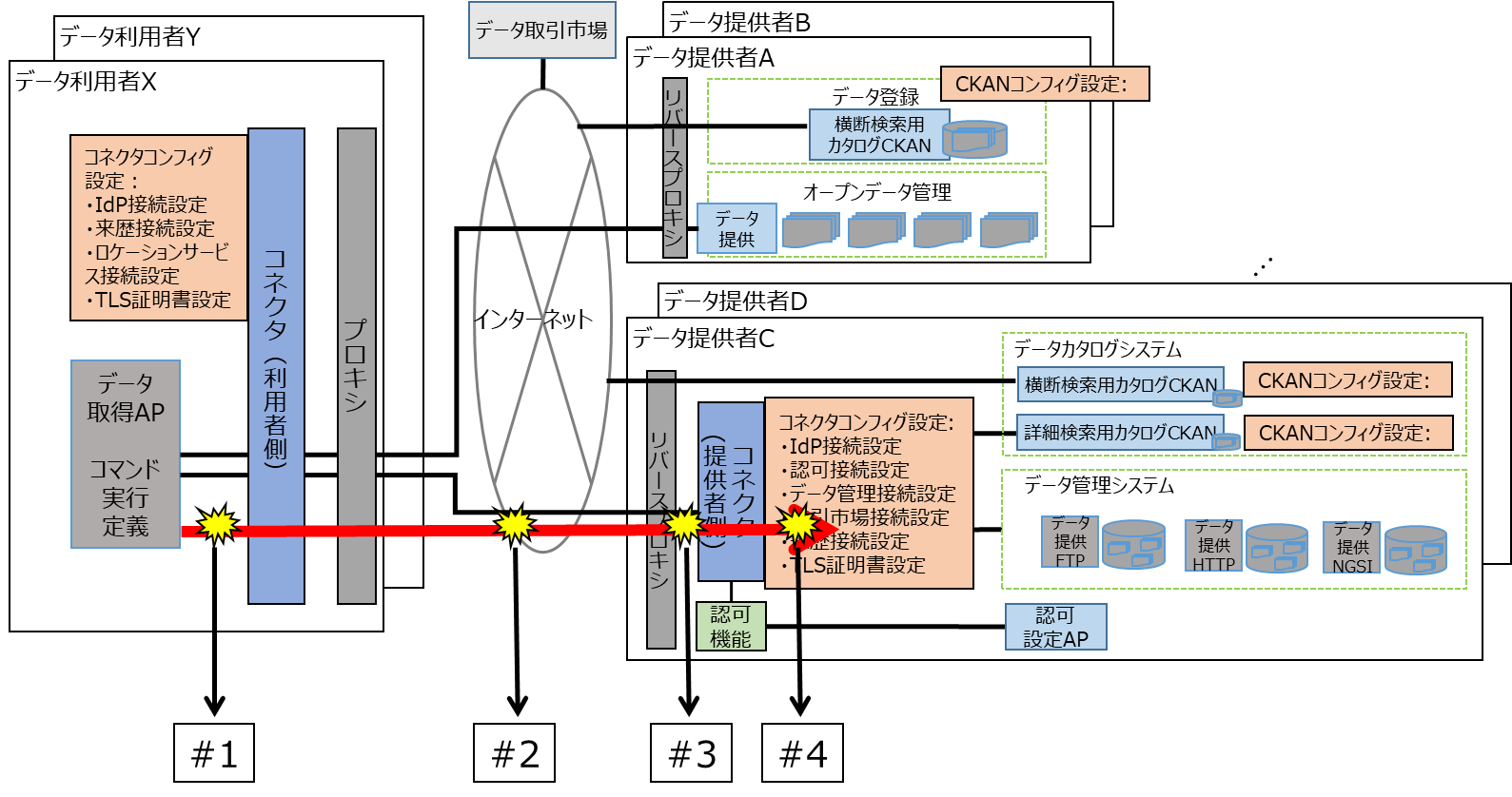


図 4.1‑1 ログ出力に関するエラー切り分けイメージ

上図にて示したエラー箇所について、切り分け方法を表 4.1‑1に示す。

表 4.1‑1 切り分け方法

テーブル

自動的に生成された説明

なお、利用者環境のプロキシから提供者コネクタへのアクセスについては、図 4.1‑2に示す方法においても確認が可能である。

【 proxy指定での提供者コネクタへのアクセス】

コマンド：curl –k https://[提供者コネクタURL]/cadde/api/v4/file --proxy localhost:3128

---------------------コマンド結果-------------------

{

"detail": "Missing header parameter 'x-cadde-resource-url'",

"status": 400,

"title": "Bad Request",

"type": "about:blank"

}

----------------------------------------------------

コネクタのエラーメッセージが返却されればクライアント認証に成功し、コネクタへアクセス出来ていることが確認可能。

【【proxy指定なしで提供者コネクタへのアクセス】

コマンド：curl –k https:// [提供者コネクタURL] /cadde/api/v4/file

---------------------コマンド結果-------------------

<html>

<head><title>400 No required SSL certificate was sent</title></head>

<body>

<center><h1>400 Bad Request</h1></center>

<center>No required SSL certificate was sent</center>

<hr><center>nginx/1.23.1</center>

</body>

</html>

----------------------------------------------------

クライアント認証に失敗したエラーが返却される。

図 4.1‑2 利用者環境のプロキシから提供者コネクタへのアクセス確認方法

#### アプリケーション内の確認方法

(1)より、コネクタ間通信のインフラ環境に問題がない場合、アプリケーション内の確認を実施する。コネクタ内の各通信が正常に実行できているかどうか、ログより確認する。確認内容については、GitHubを参照する。

# データ取得APの開発

## 利用者コネクタのAPI一覧リスト

### 認証に関するAPI

#### 利用者トークンの発行

認証サーバへアクセスし利用者トークンを取得する。

$ curl -X POST -k -s https://example.com/cadde/api/v4/token \

-H "Content-Type: application/json" \

-H "Authorization:Basic aHR0cHMvL2F1dGgwMi5zYWFzLmRhdGEtbGlua2FnZS5qcC86WkpoWVBzSHJHdThQY2lYS1NRRlFqb0hweG5zT04zb1E=" \

-d '{"user\_id": "cadde.user.aa", "password": "xxx", otop: "123456"}'

### 利用者コネクタの利用に関するAPI

#### 横断検索

横断検索サーバに対してカタログ検索を実行する。

実行例

$ curl -X GET -v 'https://CADDE.Y.co.jp/cadde/api/v4/catalog?q=\*:\*'

-H "x-cadde-search:meta"

{

"help": "http://harvest.ckan.jp:8000/api/package\_list?start=0&rows=10",

"success": true,

"result": {

　…

}

}

#### 詳細検索

提供者が持つ詳細検索用カタログサイト(CKAN)対してカタログ検索を実行する。

実行例

$ curl -X GET -v 'https://CADDE.Y.co.jp/cadde/api/v4/catalog?q=\*:\*'

-H "x-cadde-search:detail"

-H "x-cadde-provider:cadde.provider.c"

-H " Authorization:Bearer eyJhbGciOiJSUzI1NiIsInR5cCI　・・・"

{

"help": "http://harvest.ckan.jp:8000/api/package\_list?start=0&rows=10",

"success": true,

"result": {

　…

}

}

#### ファイル取得(オープンデータ)

カタログ検索結果の情報を元にオープンデータを取得する。

実行例

curl -v -X GET "localhost/cadde/api/v4/file"

-H "x-cadde-resource-url:リソースURL"

-H "x-cadde-resource-api-type:リソース提供識別子"

-H "x-cadde-provider: 提供者ID"

-o ファイル名

#### ファイル取得(限定提供データ：FTP/HTTP)

カタログ検索結果の情報を元に提供者データ管理(HTTPサーバ or FTPサーバ)からファイルを取得する。

実行例

curl -v -X GET "localhost/cadde/api/v4/file"

-H "x-cadde-resource-url:リソースURL"

-H "x-cadde-resource-api-type:リソース提供識別子"

-H "x-cadde-provider: 提供者ID"

-H "Authorization:Bearer 利用者トークン"

-o ファイル名

#### ファイル取得(限定提供データ：NGSI)

カタログ検索結果の情報を元に提供者データ管理(NGSIサーバ)からファイルを取得する。

実行例

curl -v -X GET "localhost/cadde/api/v4/file"

-H "x-cadde-resource-url:リソースURL"

-H "x-cadde-resource-api-type:リソース提供識別子"

-H "x-cadde-provider: 提供者ID"

-H “x-cadde-options: Fiware-Service: {NGSIテナント}, Fiware-ServicePath: {NGSIサービスパス}”

-H "Authorization:Bearer 利用者トークン" -o ファイル名

### 来歴管理に関するAPI

#### 来歴確認

来歴管理モジュールに対してAPI履歴ID検索を行う。

実行例

curl -v -X POST "http://{来歴管理サービスのFQDN}:{ポート番号}/v2/searchevents" \

-H "Cache-Control: no-cache" \

-H "Authorization: {利用者トークン}" \

-H "Content-Type: application/json" \

-d '{"selector": {検索条件}}''

来歴管理モジュールに対して来歴確認を行う。

実行例

curl -v -X GET "http://{来歴管理サービスのFQDN}:{ポート番号}/v2/lineage/{イベントID}" \

-H "Cache-Control: no-cache" \

-H "Authorization: {利用者トークン}" \

-H "x-direction:{探索の方向}" \

-H "x-depth:{探索の深さ}"

## バッチAPとしての利用

5.1利用者コネクタのAPI一覧リストに記載のスクリプトを組み合わせることでバッチを作成することでデータ取得の際に都度コマンドを入力する必要がなくなる。5.3、5.4、5.5ではそれぞれどのようにAPIを組み合わせればよいかユースケース別で記載している。

## データ取得のユースケース別スクリプト一覧（簡易パターン）

簡易パターンでオープンデータの取得を行う場合は、横断検索で取得したいデータを検索し、データ取得を実行することで取得できる

#### 横断検索

横断検索サーバに対してカタログ検索を実行する。

* x-cadde-searchヘッダには、'meta'を指定
* 検索クエリ内の{検索キー}には、検索文字列を指定。

実行例

curl -v -X GET "localhost/cadde/api/v4/catalog?q=検索条件" -s -S

-H "Cache-Control: no-cache"

-H "x-cadde-search: meta"

#### ファイル取得(HTTP,FTP)

横断検索カタログ結果を元に提供者データ管理から直接オープンデータを取得

実行例

curl -v -X GET "localhost/cadde/api/v4/file"

-H "x-cadde-resource-url:リソースURL"

-H "x-cadde-resource-api-type:リソース提供識別子"

-H "x-cadde-provider: 提供者ID"

-o ファイル名

## データ取得のユースケース別スクリプト一覧（基本・市場利用パターン）

基本・市場利用パターンでデータを取得する場合は、横断検索で取得したいデータを検索し、利用者・認証トークンを発行後に、詳細カタログを取得し、詳細カタログの情報をもとにデータを取得する。

#### 横断検索

横断検索サーバに対してカタログ検索を実行する。

* x-cadde-searchヘッダには、'meta'を指定
* 検索クエリ内の{検索キー}には、検索文字列を指定。

実行例

横断検索サーバに対してカタログ検索を実行する。

実行例

curl -v -X GET "localhost/cadde/api/v4/catalog?q=検索条件" -s -S

-H "Cache-Control: no-cache"

-H "x-cadde-search: meta"

#### 利用者トークンの発行

認証サーバへ利用者トークンの取得要求を行う

* client\_idとclien\_secretにはコネクタIDとコネクタシークレットを指定
* user\_idとpasswordにはCADDEユーザIDとパスワードを指定。

トークン取得用のスクリプトファイル（ファイル名の例：getToken.sh）

#!/bin/bash

client\_id="クライアントID"

client\_secret="クライアントシークレット"

basic=$(echo -n "${client\_id}:${client\_secret}" | base64)

curl -X POST -s \

https://auth01.saas.data-linkage.jp/cadde/api/v4/token \

-H "Content-Type: application/json" \

-H "Authorization: Basic ${basic}" \

-d '{"user\_id": "ユーザーID", "password": "パスワード"}'>output

jq -r '.access\_token' < output > output2

export access\_token=`cat output2`

rm output output2

上記スクリプトファイルの実行例

source getToken.sh

#### 詳細検索

提供者が持つ詳細検索用カタログサイト(CKAN)対してカタログ検索を実行する。

* x-cadde-searchヘッダには、'detail'を指定。
* x-cadde-providerヘッダには、横断カタログ検索結果(extras:caddec\_provider\_id)から取得した提供者IDを指定。
* Authorizationヘッダには、認証認可を行う場合はIdPが発行したトークンの値を指定。認証認可を行わない場合はヘッダ未設定。
* 検索クエリ内の{詳細検索用データセットID} には、横断カタログ検索結果(extras:caddec\_dataset\_id\_for\_detail)の値を設定

実行例

curl -v -X GET "localhost/cadde/api/v4/catalog?q=:**{詳細検索用データセットID}**" -s -S

-H "Cache-Control: no-cache"

-H "x-cadde-search: detail"

-H "x-cadde-provider: **{提供者ID}**"

-H "Authorization:Bearer ${consumer\_token}"

#### ファイル取得(HTTP、FTP)

カタログ検索結果の情報を元に提供者データ管理(HTTPサーバ or FTPサーバ)からファイルを取得する。

* x-cadde-providerヘッダには、詳細カタログ検索結果(extras:caddec\_provider\_id)から取得した提供者IDを指定。
* x-cadde-resource-urlヘッダには、詳細カタログ検索結果(resources:download\_url)から取得したファイルのダウンロードURLを指定。
* x-cadde-resource-api-typeヘッダには、詳細カタログ検索結果(resources:caddec\_resource\_type)から取得したリソース提供手段の識別子(file/http or file/ftp)を指定。
* Authorizationヘッダには、認証認可を行う場合はIdPが発行したトークンの値を指定。認証認可を行わない場合はヘッダ未設定。

実行例

curl -v -X GET "localhost/cadde/api/v4/file"

-H "x-cadde-resource-url:リソースURL"

-H "x-cadde-resource-api-type:リソース提供識別子"

-H "x-cadde-provider: 提供者ID"

-H "Authorization:Bearer 利用者トークン"

-o ファイル名

#### ファイル取得(NGSI)

カタログ検索結果の情報を元に提供者データ管理(HTTPサーバ or FTPサーバ)からファイルを取得する。

パラメータの内容はCADDEコネクタを利用した NGSIデータの取得方法(https://github.com/CADDE-sip/connector/blob/master/doc/README\_NGSI.md)を参照

実行例

curl -v -X GET "localhost/cadde/api/v4/file"

-H "x-cadde-resource-url:リソースURL"

-H "x-cadde-resource-api-type:リソース提供識別子"

-H "x-cadde-provider: 提供者ID"

-H “x-cadde-options: Fiware-Service: {NGSIテナント}, Fiware-ServicePath: {NGSIサービスパス}”

-H "Authorization:Bearer 利用者トークン" -o ファイル名来歴確認

#### 来歴確認

来歴管理モジュールに対してAPI履歴ID検索を行う。

* selectorには、{"cdleventid":"検索したいイベントID"}や{"datauser":"利用者ID"}等の検索条件を記載する。（couch dbの構文に準拠している）

実行例

curl -v -X POST "http://{来歴管理サービスのFQDN}:{ポート番号}/v2/searchevents" \

-H "Cache-Control: no-cache" \

-H "Authorization: {利用者トークン}" \

-H "Content-Type: application/json" \

-d '{"selector": {検索条件}}''

来歴管理モジュールに対して来歴確認を行う。

* direction=検索方向 BACKWARD: 過去方向へ検索 FORWARD: 未来方向へ検索
* depth=検索深度 -1または省略時は全ての履歴を取得

実行例

curl -v -X GET "http://{来歴管理サービスのFQDN}:{ポート番号}/v2/lineage/{イベントID}" \

-H "Cache-Control: no-cache" \

-H "Authorization: {利用者トークン}" \

-H "x-direction:{探索の方向}" \

-H "x-depth:{探索の深さ}"

## データ取得のユースケース別スクリプト一覧（連続取得パターン）

定期的にデータを取得したい場合は、下記のような繰り返し実行するバッチを作成することで実現することができる。各APIはそれぞれ独立したバッチとすることでメンテナンスが容易になる。

スクリプトファイルの例（各APIはそれぞれ独立した形でバッチにしている、各バッチについては省略しているが5.4を参考に作成可能）

#!/bin/bash

startTime=`date +%s`

echo "$startTime 秒(開始時刻(UNIX TIME))"

nowTime=`date +%s`

elapsedTime=0

n=1

#規定時間まで実行する

while [ $elapsedTime -lt 43200 ]; do

echo $n"回目"

#横断検索

echo ''

echo '###### 横断検索 #####'

sh searchMeta.sh

#詳細検索

echo ''

echo '###### 詳細検索 #####'

sh searchDetail.sh

#データ取得

echo ''

echo '###### データ取得 #####'

sh getFiles.sh

#来歴取得

echo ''

echo '###### 来歴取得 #####'

sh historySearchevents.sh

sh historyLineage.sh

echo ''

echo $n"回目終了"

echo '##### 経過時間 #####'

nowTime=`date +%s`

elapsedTime=`expr $nowTime - $startTime `

i=$elapsedTime

((sec=i%60, min=(i%3600)/60, hrs=i/3600))

timestamp=$(printf "%d:%02d:%02d" $hrs $min $sec)

echo $timestamp

#間隔をあける

sleep 1m

n=`expr $n + 1`