分野間連携データ基盤

詳細設計書

(共通編)

第4.0版



変更来歴

|  |  |  |  |
| --- | --- | --- | --- |
| # | 版数 | 発行年月日 | 変更内容 |
| 1 | 第1.0版 | 2020/07/07 | 新規作成 |
| 2 | - | 2020/11/30 | 2020年09月版の記載を2021年03月版に修正 |
| 3 | 第2.0版 | 2021/02/09 | FTP認証情報もドメインをキーに取得するように変更(ドメイン情報が見つからない場合はanonymousを使用する)。 |
| 4 | 第3.0版 | 2021/11/19 | * 章構成などをはじめとした詳細設計書レイアウトの調整 * 基本設計書との文言ゆれを調整 |
| 5 | 第4.0版 | 2023/03/17 | * 適用範囲を2022年03月から2023年03月に修正 |

目次

[１． はじめに 1](#_Toc103954706)

[１．１ ドキュメント体系 1](#_Toc103954707)

[１．２ 適用範囲 1](#_Toc103954708)

[２． 共通内部仕様 2](#_Toc103954709)

[２．１ 使用プログラム言語、フレームワーク 2](#_Toc103954710)

[２．２ エラー共通処理 2](#_Toc103954711)

[２．２．１ 各サブシステム内で発生したエラー 2](#_Toc103954712)

[２．２．２ 必須パラメータチェックにて発生したエラー 3](#_Toc103954713)

[２．２．３ HTTPステータスコードについて 4](#_Toc103954714)

[２．２．４ ログ出力 5](#_Toc103954715)

[３． カタログ検索I/F(HTTPS CKAN)サブシステム 6](#_Toc103954716)

[３．１ 内部仕様 6](#_Toc103954717)

[３．１．１ データフロー 6](#_Toc103954718)

[３．１．２ 公開インタフェース 6](#_Toc103954719)

[３．２ 機能詳細 7](#_Toc103954720)

[３．２．１ 処理フロー 7](#_Toc103954721)

[３．２．２ 出力ログ 7](#_Toc103954722)

[４． データ提供I/F(HTTPS)サブシステム 8](#_Toc103954723)

[４．１ 内部仕様 8](#_Toc103954724)

[４．１．１ データフロー 8](#_Toc103954725)

[４．１．２ 公開インタフェース 8](#_Toc103954726)

[４．１．３ 内部データ一覧 9](#_Toc103954727)

[４．１．４ コンフィグ定義 9](#_Toc103954728)

[４．２ 機能詳細 10](#_Toc103954729)

[４．２．１ 処理フロー 10](#_Toc103954730)

[４．２．２ 出力ログ 10](#_Toc103954731)

[５． データ提供I/F(FTP)サブシステム 11](#_Toc103954732)

[５．１ 内部仕様 11](#_Toc103954733)

[５．１．１ データフロー 11](#_Toc103954734)

[５．１．２ 公開インタフェース 11](#_Toc103954735)

[５．１．３ 内部データ一覧 12](#_Toc103954736)

[５．１．４ コンフィグ定義 12](#_Toc103954737)

[５．２ 機能詳細 13](#_Toc103954738)

[５．２．１ 処理フロー 13](#_Toc103954739)

[５．２．２ 出力ログ 13](#_Toc103954740)

[６． データ提供IF(HTTPS NGSI)サブシステム 14](#_Toc103954741)

[６．１ 内部仕様 14](#_Toc103954742)

[６．１．１ データフロー 14](#_Toc103954743)

[６．１．２ 公開インタフェース 15](#_Toc103954744)

[６．１．３ 内部データ一覧 16](#_Toc103954745)

[６．１．４ コンフィグ定義 16](#_Toc103954746)

[６．２ 機能詳細 17](#_Toc103954747)

[６．２．１ 処理フロー 17](#_Toc103954748)

[６．２．２ 出力ログ 17](#_Toc103954749)

# はじめに

## ドキュメント体系

本書はシステムの内部設計を記載する。構成ファイル一覧を表 １．１‑１に記載する。

表 １．１‑１詳細設計書構成ファイル一覧

|  |  |
| --- | --- |
| # | ドキュメント名 |
|  | 詳細設計書\_共通編(docx) |
|  | 詳細設計書\_提供者編(docx) |
|  | 詳細設計書\_利用者編(docx) |
|  | 基本設計書\_別紙2\_コンフィグパラメータ一覧(xlsx) |
|  | 詳細設計書\_別紙1\_メッセージ一覧(xlsx) |
|  | 詳細設計書\_別紙2\_利用者\_コネクタメイン(html) |
|  | 詳細設計書\_別紙3\_利用者\_カタログ検索(html) |
|  | 詳細設計書\_別紙4\_利用者\_データ交換(CADDE)(html) |
|  | 詳細設計書\_別紙5\_利用者\_認証(html) |
|  | 詳細設計書\_別紙6\_利用者\_来歴管理(html) |
|  | 詳細設計書\_別紙7\_提供者\_コネクタメイン(html) |
|  | 詳細設計書\_別紙8\_提供者\_カタログ検索(html) |
|  | 詳細設計書\_別紙9\_提供者\_データ交換 (CADDE)(html) |
|  | 詳細設計書\_別紙10\_提供者\_認可(html) |
|  | 詳細設計書\_別紙11\_提供者\_来歴管理(html) |
|  | 詳細設計書\_別紙12\_利用者コネクタ\_ディレクトリ構成(xlsx) |
|  | 詳細設計書\_別紙13\_提供者コネクタ\_ディレクトリ構成(xlsx) |

## 適用範囲

本書は2023年03月版の詳細設計を対象とする。

本書には利用者コネクタと提供者コネクタで共通する部分について記載する。

表 １．２‑１共通項目一覧

|  |  |  |
| --- | --- | --- |
| # | 項目 | 内容 |
|  | 共通内部仕様 | システムで共通するポリシーを記載する |
|  | データ提供I/F(HTTPS) | コネクタメインがデータ管理からHTTPSでファイルを取得する機能を記載する。 |
|  | データ提供I/F(FTP) | コネクタメインがデータ管理からFTPでファイルを取得する機能を記載する。 |
|  | データ提供I/F(HTTPS NGSI) | コネクタメインがデータ管理からHTTPS NGSIでデータを取得する機能を記載する。  ※NEC担当 |

# 共通内部仕様

## 使用プログラム言語、フレームワーク

本システムで使用するプログラム言語、フレームワークを下表に記載する。

表 ２．１‑１ プログラム言語・フレームワーク、文字コード

|  |  |  |
| --- | --- | --- |
| # | 対象 | ソフトウェア(規格) |
|  | プログラム言語 | Python3.8 64bit |
|  | Webフレームワーク | Connexion(Flask) |
|  | WebAPI定義 | OpenAPI 3.0 |
|  | 文字コード | UTF-8 |

## エラー共通処理

### 各サブシステム内で発生したエラー

各サブシステムのコントローラ、サービス処理内でエラーが発生した場合は、PythonのExceptionを継承した本機能独自Exceptionにメッセージコードとエラー時のHTTPステータスコードを設定して、Exceptionを発生させる。

独自ExceptionはPythonのExceptionを継承し、CaddeExceptionクラスを作成する。

エラー発生時は下記1.のようにメッセージコードを指定して、CaddeExceptionを投げる。個別にHTTPステータスコードやメッセージの置き換え文字列を設定する場合は、下記2.のようにメッセージコード、HTTPステータスコード、置き換え文字列を指定して、CaddeExceptionを投げる。

1. raise CaddeException((メッセージコード))

raise CaddeException(  
 (メッセージコード),  
 status\_code=(HTTPステータスコード),  
 replace\_str\_list=(置き換え文字列)  
 )

メッセージコードの一覧は「詳細設計書\_別紙1\_メッセージ一覧(xlsx)」を参照のこと

発生したExceptionは各サブシステムのメインファイル(\_\_main\_\_.py)でハンドリングし、以下の処理を行う。

1. 独自Exceptionに設定されたメッセージとHTTPステータスコードを取得する。
2. 発生したExceptionのエラー内容とスタックトレースをログ出力する。
3. 以下の形式でレスポンス用のJSON文字列を作成する。  
   {  
    "detail":(1.で取得したメッセージ),  
    "status":(1.で取得したHTTPステータスコード),  
    "title":"" (空文字),  
    "type":"" (空文字)  
   }
4. 上記で作成したレスポンス用JSON文字列と独自Exceptionに設定されたHTTPステータスコードをレスポンスに設定し、処理を終了する。

レスポンスはflaskライブラリを利用して作成する。レスポンスのContent-Typeは"application/json"に設定する。

### 必須パラメータチェックにて発生したエラー

各サブシステムのコンテナに通信を行った際に、別紙2～別紙11のParameters欄のDescriptionに"Required"の記載があるパラメータは必須パラメータチェックが行われる。パラメータチェックにて、当該パラメータ設定されていない場合、以下の形式のレスポンスを返す。

{  
 "detail": Missing header parameter (必須パラメータ名),  
 “status”: 400,  
 "title": Bad Request,  
 "type": about:blank  
}

### HTTPステータスコードについて

各処理で応答を返す際のHTTPステータスコードについて以下の方針で設定する。

また、他コンテナへ処理要求を行った際のステータスコード値については、受け取ったものをそのまま返却するものとする。

表 ２．２．３‑１ HTTPステータスコード一覧

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | 定義名 | 処理結果 | ステータスコード | レスポンス条件 |
|  | STATUS\_CODE\_SUCCESS | 正常終了 | 200 | 正常終了時 |
|  | STATUS\_CODE\_PARAMETER\_INVALID | パラメータ不正 | 400 | パラメータ確認時に必須パラメータが存在しない場合 |
|  | STATUS\_CODE\_DATA\_ACCESS\_FAILURE | アクセス権不正 | 403 | HTTPS接続で認証失敗した場合  FTP接続で認証失敗した場合  認可および契約確認に失敗した場合 |
|  | STATUS\_CODE\_FILE\_REFERENCE\_FAILURE | データ参照失敗 | 404 | FTP接続にて指定したディレクトリにファイルが存在しない場合  HTTPS接続にて指定したURLにファイルが存在しない場合  HTTPS NGSI接続にて指定したエンティティが存在しない場合 |
|  | STATUS\_CODE\_REQUEST\_TIMEOUT | リクエストタイムアウト | 408 | HTTPリクエスト、FTPリクエストがタイムアウトした場合 |
|  | STATUS\_CODE\_CONFLICT | データ競合 | 409 | HTTPS NGSI接続にて指定したリソースが複数存在する場合 |
|  | STATUS\_CODE\_OTHER\_ERROR | その他エラー | 500 | その他のエラー発生時 |

### ログ出力

本システムは次の目的でログを出力する。デフォルトはWarningレベル以上を出力する。

ログは標準出力に出力することで、dockerが保存し、「docker logs」コマンドを使用して、ユーザが確認する。

表 ２．２．４‑１ ログ種別一覧

|  |  |  |
| --- | --- | --- |
| # | 目的 | ログレベル |
| 1 | 正常動作時の動作内容の確認 | Debug |
| 2 | エラー内容の確認 | Warning |
| 3 | スタックトレースによるエラー発生個所の確認 | Warning |

ログのフォーマットは以下の通りとする。

YYYY-MM-DD hh:mm:ss,SSS (ログレベル) (発生関数) :(説明文字列)

例:

2020-07-01 12:34:56,789 Warning \_\_main\_\_: エラーが発生しました。エラー内容:XXX

# カタログ検索I/F(HTTPS CKAN)サブシステム

## 内部仕様

### データフロー

コネクタ

メイン

カタログ検索I/F

(HTTPS CKAN)

CKAN

凡例:

: 対象サブシステム

: その他システム

: HTTPS通信

: 関数呼び出し

コネクタメインコンテナ

カタログ

検索要求

検索結果

検索結果

データ管理

カタログ

検索要求

図 ３．１．１‑１ システム構成

### 公開インタフェース

表 ３．１．２‑１ 公開インタフェース

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | 機能 | 関数名 | 概要 | 引数 | 戻り値 |
|  | カタログ検索 | ckan\_search | CKANサーバからカタログ検索結果を取得して返却する。 | 1. 検索条件 2. CKAN URL | 検索結果文字列 |

## 機能詳細

### 処理フロー

#### カタログ検索

1. 検索条件と対象カタログサイトURL （CKAN URL）をメッセージコードとともにログ出力する。
2. CKAN URLとクエリストリングから次の方式で検索用URLを作成する。  
   検索用URL = CKAN URL + クエリストリング
3. 検索用URLに対してGET通信でHTTPリクエストを発行する。
4. 検索結果を確認し、次のいずれかの処理を行う。

* 検索結果のステータスコードが「400」の場合は、パラメータが不正の値である独自Exceptionを発生させる。
* 検索結果のステータスコードが「400」および「2xx」以外の場合は、独自Exceptionを発生させる。
* 検索結果のステータスコードが「2xx」の場合、検索結果文字列を呼び出し元に返却する。

### 出力ログ

独自Exceptionに付与するエラーメッセージおよび出力ログの情報は「詳細設計書\_別紙1\_メッセージ一覧(xlsx)」を参照すること。

# データ提供I/F(HTTPS)サブシステム

## 内部仕様

### データフロー

コネクタ

メイン

データ提供

I/F(HTTPS)

HTTPS

サーバ

凡例:

: 対象サブシステム

: その他システム

: 通信

: 関数呼び出し

コネクタメインコンテナ

データ提供

ファイル

HTTPリクエスト

ファイルデータ

データ管理

図 ４．１．１‑１ システム構成

### 公開インタフェース

表 ４．１．２‑１ 公開関数インタフェース

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | 機能 | 関数名 | 概要 | 引数 | 戻り値 |
|  | データ提供 | provide\_data\_http | HTTPSサーバからファイルを取得して返却する。ベーシック認証が必要なドメインの場合は、ベーシック認証を行う。  ※ダイジェスト認証、TLS証明書認証、OAuth等の認証処理は実施しない。 | 1.リソースURL  2. ヘッダ情報 | ファイルデータ |

### 内部データ一覧

コンフィグを読み込み、内部データとして保持する。

表 ４．１．３‑１ 内部データ一覧

|  |  |  |
| --- | --- | --- |
| # | データ名 | 概要 |
|  | basic\_auth | リソースURLのドメインをキーとして、basic\_idとbasic\_passを保持する |
|  | basic\_id | ベーシック認証を行う際に指定するIDを保持する |
|  | basic\_pass | ベーシック認証を行う際に指定するパスワードを保持する |

### コンフィグ定義

コンフィグの一覧を記載する。詳細は「基本設計書\_別紙2\_コンフィグパラメータ一覧(xlsx)」を参照すること。

## 機能詳細

### 処理フロー

#### データ提供

1. パラメータとして、リソースURL、ヘッダ情報を取得する。リソースURLが取得できない場合は、独自Exceptionを発生させる。
2. リソースURL、ヘッダ情報をログ出力する。
3. リソースURLからドメイン(ポート番号を含む)を取得する。ドメインが取得できない場合は、独自Exceptionを発生させる。
4. コンフィグファイルからBasic認証情報(basic\_auth)を取得する。取得できない場合は、処理7.に遷移する。
5. 取得したBasic認証情報から、ドメインをキーとしてベーシック認証のIDとベーシック認証のパスワードを取得する。ドメインが取得できない場合は処理7.に遷移する。

ドメインが取得でき、ベーシック認証のIDとベーシック認証のパスワードが取得できない場合は、独自Exceptionを発生させる。

1. ベーシック認証のIDとベーシック認証のパスワードをファイル取得時の設定に追加する。
2. リソースURLからファイルを取得する。この際、コネクトタイムアウト設定(サーバとの接続を確立するまでのタイムアウト設定)は10秒、リードタイムアウト(サーバがレスポンスを返してくるまでのタイムアウト)は60秒に設定する。
3. ファイル取得結果を確認し、失敗している場合は、エラー内容を確認する。

検索結果のステータスコードが「200」以外の場合は、独自Exceptionを発生させる。

検索結果のステータスコードが「200」の場合は、ファイルデータを呼び出し元に返却する。

### 出力ログ

独自Exceptionに付与するエラーメッセージおよび出力ログの情報は「詳細設計書\_別紙1\_メッセージ一覧(xlsx)」を参照すること。

# データ提供I/F(FTP)サブシステム

## 内部仕様

### データフロー

コネクタ

メイン

データ提供

I/F(FTP)

FTPサーバ

凡例:

: 対象サブシステム

: その他システム

: 通信

: 関数呼び出し

コネクタメインコンテナ

データ提供要求

ファイル

FTPリクエスト

ファイルデータ

データ管理

図 ５．１．１‑１ システム構成

### 公開インタフェース

表 ５．１．２‑１ 公開関数インタフェース

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | 機能 | 関数名 | 概要 | 引数 | 戻り値 |
|  | データ提供 | provide\_data\_ftp | FTPサーバからファイルを取得して返却する。  ※リソースURLに対して、ファイル取得を行う。接続方法はFTPのみとし、SFTPを用いた認証は行わない。 | 1.リソースURL | ファイルデータ |

### 内部データ一覧

コンフィグを読み込み、内部データとして保持する。

表 ５．１．３‑１ 内部データ一覧

|  |  |  |
| --- | --- | --- |
| # | データ名 | 概要 |
|  | ftp\_id | ftp接続の際に指定するIDを保持する |
|  | ftp\_pass | ftp接続の際に指定するパスワードを保持する |

### コンフィグ定義

コンフィグの一覧を記載する。詳細は「基本設計書\_別紙2\_コンフィグパラメータ一覧(xlsx)」を参照すること。

## 機能詳細

### 処理フロー

#### データ提供

1. パラメータとして、リソースURLを取得する。リソースURLが取得できない場合は、独自Exceptionを発生させる。
2. リソースURLをログ出力する。
3. リソースURLからドメイン(ポート番号を含む)を取得する。ドメインが取得できない場合は、独自Exceptionを発生させる。
4. コンフィグファイルからFTP認証情報(ftp\_auth)を取得する。取得できない場合は、処理7.に遷移する。
5. 取得したFTP認証情報から、ドメインをキーとして認証のIDとパスワードを取得する。ドメインに合致する情報が取得できない場合は処理7.に遷移する。
6. ドメインが取得でき、IDとパスワードが取得できない場合は、HTTPステータスコードを「500」に設定し、独自Exceptionを発生させる。
7. リソースURLからFTP接続のID、FTP接続のパスワードを利用してファイルを取得する(コンフィグから該当ドメインの情報が取得できない場合はIDとパスワードに"anonymous"を使用する)。この際、コネクトタイムアウト設定(サーバとの接続を確立するまでのタイムアウト設定)は10秒、リードタイムアウト(サーバがレスポンスを返してくるまでのタイムアウト)は60秒に設定する。FTP通信をする際にリードタイムアウトを設定できないため、本処理自体にタイムアウトを設定し、60秒以内に処理が完了しない場合はタイムアウトとなるように設定する。
8. ファイル取得に失敗している場合は、エラー内容を確認し、独自Exception発生させる。
9. ファイルデータを呼び出し元に返却する。

### 出力ログ

独自Exceptionに付与するエラーメッセージおよび出力ログの情報は「詳細設計書\_別紙1\_メッセージ一覧(xlsx)」を参照すること。

# データ提供IF(HTTPS NGSI)サブシステム

## 内部仕様

### データフロー

コネクタ

メイン

データ提供

IF(HTTPS NGSI)

OrionContext

Broker

凡例:

: 対象サブシステム

: その他システム

: 通信

: 関数呼び出し

コネクタメインコンテナ

データ取得要求

コンテキスト情報

HTTPリクエスト

コンテキスト情報

データ管理

図 ６．１．１‑１ システム構成

### 公開インタフェース

表 ６．１．２‑１ 公開インタフェース

|  |  |  |
| --- | --- | --- |
| # | 機能 | 概要 |
|  | データ提供  (provide\_data\_ngsi) | OrionContextBrokerからコンテキスト情報を取得して返却する。  ※リソースURLに対して、コンテキスト情報取得を行う。 |

表 ６．１．２‑２ 関数仕様

|  |  |  |
| --- | --- | --- |
| 書式,実行形式 | def provide\_data\_ngsi(resource\_url, user\_id, options={}) | |
| 引数 | string | 入力：resource\_url リソースURL  コンテキスト情報取得先のOrionContextBrokerのURLを指定する。 |
| string | 入力：user\_id　CADDEユーザID（利用者）  メイン制御 認証・認可処理で取得するCADDEユーザID（利用者）を設定 |
| dict（オプション） | 入力：options オプション（コネクタメインのオプション引数をそのままInputとする）  コンテキスト情報を取得するためのその他の設定値。  引数の値はNGSI API発行時のリクエストヘッダに使用する。リクエストヘッダへ設定する内容はdict型のキーをヘッダキー、値をそのヘッダの値に使用する。  Default：{}  例  {  "Fiware-Service" : " a",  "Fiware-ServicePath"： "/abc"  } |
| 戻り値 | BytesIO | 取得データ（コンテキスト情報） |
| dict | レスポンスヘッダ情報 key:ヘッダ名 value:パラメータ  レスポンスヘッダがない場合は空のdictを返す |
| 例外 | urllib.request.urlopenで異常終了した場合は、独自Exceptionを発生させる。 | |
| 説明 | リソースURLに対してHTTPリクエストしコンテキスト情報を取得する。 | |
| 備考 | - | |

### 内部データ一覧

コンフィグを読み込み、内部データとして保持する。

表 ６．１．３‑１ 内部データ一覧

|  |  |  |
| --- | --- | --- |
| # | データ名 | 概要 |
|  | ngsi\_auth | OrionContextBrokerに接続するための認証情報を保持する |

### コンフィグ定義

コンフィグの一覧を記載する。詳細は「基本設計書\_別紙2\_コンフィグパラメータ一覧(xlsx)」を参照すること。

## 機能詳細

### 処理フロー

#### データ提供

1. パラメータとして、リソースURL、CADDEユーザID（利用者）、オプションを取得する。リソースURLが取得できない場合は、独自Exceptionを発生させる。

オプションは必須ではないためチェックしない。

独自Exceptionに関しては、２．共通内部仕様を参照。

1. コンフィグファイルからCADDEユーザID（利用者）（user\_idは"TEST\_ID"固定）をキーにアクセストークンを取得する。

対象のデータ管理サーバがアクセストークン不要のケースを考慮し、コンフィグファイルの読み込みに失敗した場合、コンフィグファイルからアクセストークンが取得できない場合はエラーにせず、HTTPリクエストのヘッダに設定しない。

1. リソースURLをログ出力する。
2. リソースURLに対してHTTPリクエストをGETメソッドで実行する。

アクセストークンが取得できた場合は、Authorizationヘッダを設定する。オプションにデータが存在する場合(Fiware-Service、FiwareServiceなど)は、その内容をリクエストヘッダに設定する。

1. HTTPリクエストでエラーした場合は、エラー内容を確認する。

①パラメータ不正のエラー（HTTPステータスコード400）が発生した場合は、独自Exceptionを発生させる。

②認証情報が不正のエラー（HTTPステータスコード401）が発生した場合は、独自Exceptionを発生させる。

③指定したリソースが見つからなかったエラー（HTTPステータスコード404）が発生した場合は、独自Exceptionを発生させる。

④指定したリソースが複数存在するエラー（HTTPステータスコード409）が発生した場合は、独自Exceptionを発生させる。

⑤上記以外のエラーが発生した場合は、独自Exceptionを発生させる。

1. HTTPリクエストが正常に終了した場合は、レスポンスデータ（コンテキスト情報）とレスポンスヘッダを返却する。

HTTPリクエストの結果にレスポンスヘッダがなければ、レスポンスヘッダは空のdictで返却する。

### 出力ログ

独自Exceptionに付与するエラーメッセージおよび出力ログの情報は「詳細設計書\_別紙1\_メッセージ一覧(xlsx)」を参照すること。