産業用データ連携基盤

基本設計書

コネクタ提供機能

別紙2 コンフィグパラメーター一覧

第1.0版

変更来歴

版数	発行年月日	変更内容
0.9	2023/11/14	CADDEを活用して、産業データ連携基盤を開発するにあたり版数0.9として作成
		データ交換時に原本来歴を参照しないため、カタログから交換実績記録用リソースIDを検索する・しないを管理するコンフィグを追加
1.0	2024/3/21	用語の見直し（「データ利用者」を「データ受領者」へ修正等）を反映

コンフィグファイル一覧

提供者

#	シート名	サブシステム名	コンフィグファイル名	コンフィグ配置コンテナ	説明
1	提供者_カタログ検索IF	カタログ検索I/F	provider_ckan.json	コネクタメイン	提供者側カタログ検索に関するコンフィグファイル 提供者コネクタメインコンテナ内に配置
2	提供者_コネクタ情報	コネクタメイン	connector.json	コネクタメイン	提供者コネクタに関するコンフィグファイル 提供者コネクタメインコンテナ内に配置
3	提供者_データ提供IF(HTTPS)	データ提供I/F(HTTPS)	http.json	コネクタメイン	データ提供I/F(HTTPS)関数が読むコンフィグファイル 提供者コネクタメインコンテナ内に配置
4	提供者_データ提供IF(FTP)	データ提供I/F(FTP)	ftp.json	コネクタメイン	データ提供I/F(FTP)関数が読むコンフィグファイル 提供者コネクタメインコンテナ内に配置
5	提供者_データ提供IF(HTTPS NGSI)	データ提供I/F(NGSI)	ngsi.json	コネクタメイン	データ提供I/F(NGSI)関数が読むコンフィグファイル 提供者コネクタメインコンテナ内に配置
6	認可サーバ情報	認可I/F	authorization.json	認可I/F	認可サーバに関するコンフィグファイル 提供者側認可I/Fコンテナ内に配置
7	来歴管理I/F情報	来歴管理I/F	provenance.json	来歴管理I/F	来歴管理I/Fに関するコンフィグファイル 提供者側来歴管理I/Fコンテナ内に配置
8	提供者環境情報	ー	.env	コネクタメイン	提供者コネクタに関する環境変数設定

受領者

#	シート名	サブシステム名	コンフィグファイル名	コンフィグ配置コンテナ	説明
1	受領者 カタログ検索IF	カタログ検索I/F	public_ckan.json	カタログ検索I/F	受領者側カタログ検索I/Fコンテナに配置するコンフィグファイル
2	コネクタロケーション	コネクタメイン	location.json	コネクタメイン	受領者コネクタメインコンテナに配置するコンフィグファイル
3	受領者 コネクタ情報	コネクタメイン	connector.json	コネクタメイン	受領者コネクタに関するコンフィグファイル 受領者コネクタメインコンテナ内に配置
4	受領者 データ提供IF(HTTPS)	データ提供I/F(HTTPS)	http.json	コネクタメイン	データ提供I/F(HTTPS)関数が読むコンフィグファイル 受領者コネクタメインコンテナ内に配置
5	受領者 データ提供IF(FTP)	データ提供I/F(FTP)	ftp.json	コネクタメイン	データ提供I/F(FTP)関数が読むコンフィグファイル 受領者コネクタメインコンテナ内に配置
6	受領者 データ提供IF(HTTPS NGSI)	データ提供I/F(HTTPS NGSI)	ngsi.json	コネクタメイン	データ提供I/F(NGSI)関数が読むコンフィグファイル 受領者コネクタメインコンテナ内に配置
7	来歴管理I/F情報	来歴管理I/F	provenance.json	来歴管理I/F	来歴管理I/Fに関するコンフィグファイル 受領者側来歴管理I/Fコンテナ内に配置
8	受領者環境情報	ー	.env	コネクタメイン	受領者コネクタに関する環境変数設定

一覧

#	要素	論理名	デフォルト値	型	値域	M(必須) /O(任意)	説明
1	release_ckan_url	カタログサイト(公開)アクセスURL	https://example.com	文字列	-	O	提供者公開CKANサイトのURLを記載する 提供者公開CKANサイトが存在しない場合は、空文字""を設定する
2	detail_ckan_url	カタログサイト(詳細)アクセスURL	https://example.com	文字列	-	M	提供者詳細CKANサイトのURLを記載する 提供者公開CKANと共用する場合は、カタログサイト(公開)アクセスURLを設定する
3	authorization	カタログサイト認可設定	true	真偽値	-	M	カタログサイト認可設定の有効・無効を管理する値を記載する true:有効 false:無効
4	packages_search_for_data_exchange	データ交換時のリソース検索設定	true	真偽値	-	O	カタログから交換実績記録用リソースIDを検索する・しないを管理する値を記載する true:有効 false:無効

記述例

```
{
   "release_ckan_url" : "https://example.com",
   "detail_ckan_url" : "https://example.com",
   "authorization" : true,
   "packages_search_for_data_exchange" : true
}
```

一覧

#	要素	論理名	デフォルト値	型	値域	M(必須) /O(任意)	説明
1	provider_id	DATA-EXユーザID（提供者）	test_provider_id	文字列	-	M	提供者IDを記載する
2	provider_connector_id	提供者コネクタID	test_provider_connector_id	文字列	-	M	提供者コネクタのID(keycloak)を記載する
3	provider_connector_secret	提供者コネクタのシークレット	test_provider_connector_secret	文字列	-	M	提供者コネクタのシークレット(keycloak)を記載する
4	provider_connector_url	提供者コネクタのURL	https://example.com/provider_1	文字列	-	M	提供者コネクタのURLを記載する
5	location_service_url	ロケーションサービスのURL	https://example.com	文字列	-	M	ロケーションサービスのURLを記載する
6	trace_log_enable	トレースログ出力可否	true	真偽値	-	M	トレースログ出力可否の設定を記載する true:有効 false:無効

記述例

```
{
    "provider_id" : "test_provider_id",
    "provider_connector_id" : "test_provider_connector_id",
    "provider_connector_secret" : "test_provider_connector_secret",
    "provider_connector_url": "http://example.com/provider_1",
    "location_service_url" : "https://testexample.com",
    "trace_log_enable" : true
}
```

一覧

#	要素	論理名	デフォルト値	型	値域	M(必須) /O(任意)	説明
1	basic_auth	ベーシック認証情報	-	配列	-	M	次のようにドメイン名をキーとして情報を格納する "basic_auth:"[{ ・・・ }, {}・・・・]
2	ドメイン情報		-	オブジェクト	-	O	BASIC認証対象のドメインを記載する
3	domain	ドメイン	sample.co.jp:8080	文字列	-	M	ベーシック認証を行うドメインを記載する デフォルトのポートを使用しない場合は:ポート番号の指定も必要
4	basic_id	ベーシック認証のID	anonymous	文字列	-	M	ベーシック認証時に利用するIDを記載する 利用者側で設定する場合は固定値"anonymous"を設定
5	basic_pass	ベーシック認証の パスワード	anonymous	文字列	-	M	ベーシック認証時に利用するパスワード 利用者側で設定する場合は固定値"anonymous"を設定
6	authorization	認可登録設定情報	-	配列	-	M	次のように来歴登録識別URLをキーとして情報を格納する "authorization:"[{ ・・・ }, {}・・・・]
7	url	認可登録対象URL	https://example.com:8080/open	文字列	-	M	リソースURLに含まれている、認可登録確認を適用する対象のURL部分を記載する。
8	enable	認可登録設定	false	真偽値	-	M	認可登録の有効・無効を管理する値を記載する true:有効 false:無効

9	contract_management_service	取引市場利用	-	オブジェクト	-	O	次のように取引市場利用を識別URLをキーとして情報を格納する "contract_management_service:"[{ ・・・ }, {}・・・・]
10	url	取引市場利用対象URL	https://example.com:8080/open	文字列	-	M	リソースURLに含まれている、取引市場を利用する対象のURL部分を記載する 使用可能文字は半角英数、一部URLとして使用可能な記号（ハイフン、アンダーバーなど）、最大文字数は255字
11	enable	取引市場利用設定	false	真偽値	-	M	取引市場利用の有効・無効を管理する値を記載する true:有効 false:無効
12	register_provenance	来歴登録設定情報	-	配列	-	M	次のように来歴登録識別URLをキーとして情報を格納する "register_provenance:"[{ ・・・ }, {}・・・・]
13	url	来歴登録対象URL	https://example.com:8080/open	文字列	-	M	リソースURLに含まれている、来歴登録確認を適用する対象のURL部分を記載する 使用可能文字は半角英数、一部URLとして使用可能な記号（ハイフン、アンダーバーなど）、最大文字数は255字
14	enable	来歴登録設定	false	真偽値	-	M	来歴登録の有効・無効を管理する値を記載する true:有効 false:無効

記述例

```
{
   "basic_auth": [
      {
         "domain" : "example.com:8080",
         "basic_id" : "anonymous",
         "basic_pass" : "anonymous"
      }
      {
         .....
      }
   ],
   "authorization": [
      {
         "url" : "https://example.com:8080/auth/",
         "enable" : true
      },
      {
         "url": "https://example.com:8080/open",
         "enable" : false
      }
   ],
   "contract_management_service": [
      {
         "url" : "https://example.com:8080/auth/",
         "enable" : true
      },
      {
         "url": "https://example.com:8080/open",
         "enable" : false
      }
   ],
   "register_provenance": [
      {
         "url" : "https://example.com:8080/auth/",
         "enable" : true
      },
      {
         "url": "https://example.com:8080/open",
         "enable" : false
      }
   ]
}
```

[一覧]

#	要素	論理名	デフォルト値	型	値域	M(必須)/O(任意)	説明

1	ftp_auth	FTP認証情報	-	オブジェクト	-	M	次のようにFTP認証情報を記載する
							合致するドメインが存在しない場合は
							"anonymous:anonymous"でログインする
							{
							"ftp_auth": [
							{
							....
							},
							{
							....
							}
							]

2	ドメイン情報						
3	domain	ドメイン	sample.co.jp:443	文字列	-	M	FTP接続するドメインを記載する デフォルトのポートを使用しない場合は:ポート番号の指定も必要
4	ftp_id	FTP接続のID	anonymous	文字列	-	M	FTP接続時に利用するIDを記載する 利用者側で設定する場合は固定値"anonymous"を設定
5	ftp_pass	FTP接続のパスワード	anonymous	文字列	-	M	FTP接続時に利用するパスワードを記載する 利用者側で設定する場合は固定値"anonymous"を設定
6	authorization	認可登録設定情報	-	配列	-	M	次のように認可登録識別URLをキーとして情報を格納する "authorization:"[{ ... }, {}....]
7	url	認可登録対象URL	ftp://example.com:8080/open	文字列	-	M	リソースURLに含まれている、認可登録確認を適用する対象のURL部分を記載する 使用可能文字は半角英数、一部URLとして使用可能な記号（ハイフン、アンダーバーなど）、最大文字数は255字
8	enable	認可登録設定	false	真偽値	-	M	認可登録の有効・無効を管理する値を記載する true:有効 false:無効

9	contract_management_service	取引市場利用	-	オブジェクト	-	O	次のように取引市場利用を識別URLをキーとして情報を格納する "contract_management_service:"[{ ・・・ }, {}・・・・]
10	url	取引市場利用対象URL	ftp://example.com:8080/open	文字列	-	M	リソースURLに含まれている、取引市場を利用する対象のURL部分を記載する 使用可能文字は半角英数、一部URLとして使用可能な記号（ハイフン、アンダーバーなど）、最大文字数は255字
11	enable	取引市場利用設定	false	真偽値	-	M	取引市場利用の有効・無効を管理する値を記載する true:有効 false:無効
12	register_provenance	来歴登録設定情報	-	配列	-	M	次のように来歴登録識別URLをキーとして情報を格納する "register_provenance:"[{ ・・・ }, {}・・・・]
13	url	来歴登録対象URL	ftp://example.com:8080/open	文字列	-	M	リソースURLに含まれている、来歴登録確認を適用する対象のURL部分を記載する 使用可能文字は半角英数、一部URLとして使用可能な記号（ハイフン、アンダーバーなど）、最大文字数は255字
14	enable	来歴登録設定	false	真偽値	-	M	来歴登録の有効・無効を管理する値を記載する true:有効 false:無効

記述例

```
{
  "ftp_auth": [
    {
    "domain" : "example.com:443",
    "ftp_id" : "anonymous",
    "ftp_pass" : "anonymous"
    },
    {
      .....
    }
  ],
  "authorization": [
    {
```

```
            "url" : "ftp://example.com:8080/auth/",
            "enable" : true
        },
        {
            "url": "ftp://example.com:8080/open",
            "enable" : false
        }
    ],
    "contract_management_service": [
        {
            "url" : "ftp://example.com:8080/auth/",
            "enable" : true
        },
        {
            "url": "ftp://example.com:8080/open",
            "enable" : false
        }
    ],
    "register_provenance": [
        {
            "url" : "ftp://example.com:8080/auth/",
            "enable" : true
        },
        {
            "url": "ftp://example.com:8080/open",
            "enable" : false
        }
    ]
}
```

一覧

#	要素	論理名	デフォルト値	型	値域	M(必須)/O(任意)	説明
1	ngsi_auth	NGSI アクセストークン	-	オブジェクト	-	M	NGSIアクセストークンを配列で格納する "ngsi_auth" :[{ ・・・ }, {・・・・]
2	ドメイン情報						
3	domain	ドメイン	example.com:8080	文字列	-	O	ベーシック認証を行うドメインを記載する デフォルトのポートを使用しない場合は:ポート番号の指定も必要
4	auth	アクセストークン	input API key.	文字列	-	M	データ管理サーバ(NGSI)へAPIアクセスするためのアクセストークン（Authorization）を記載する
5	authorization	認可登録設定情報	-	配列	-	M	次のように来歴登録識別URLをキーとして情報を格納する "authorization:"[{ ・・・ }, {}・・・・]
6	url	認可登録対象URL	https://example.com:8080/open	文字列	-	M	リソースURLに含まれている、認可登録確認を適用する対象のURL部分を記載する。サービスタイプを記載する場合、クエリストリングとして記載する 使用可能文字は半角英数、一部URLとして使用可能な記号（ハイフン、アンダーバーなど）、最大文字数は255字
7	tenant	テナント	open	文字列	-	O	テナントの指定が必要である場合、記載する
8	servicepath	サービスパス	/open	文字列	-	O	サービスパスの指定が必要である場合、記載する
9	enable	認可登録設定	false	真偽値	-	M	認可登録の有効・無効を管理する値を記載する true:有効 false:無効
10	contract_management_service	取引市場利用	-	オブジェクト	-	O	次のように取引市場利用を識別URLをキーとして情報を格納する "contract_management_service:"[{ ・・・ }, {}・・・・]

11			url	取引市場利用対象URL	https://example.com:8080/v2/entities?type=open	文字列	-	M	リソースURLに含まれている、取引市場を利用する対象のURL部分を記載する。サービスタイプを記載する場合、クエリストリングとして記載する 使用可能文字は半角英数、一部URLとして使用可能な記号（ハイフン、アンダーバーなど）、最大文字数は255字
12			tenant	テナント	open	文字列	-	O	テナントの指定が必要である場合、記載する
13			servicepath	サービスパス	/open	文字列	-	O	サービスパスの指定が必要である場合、記載する
14			enable	取引市場利用設定	false	真偽値	-	M	取引市場利用の有効・無効を管理する値を記載する true:有効 false:無効
15	register_provenance			来歴登録設定情報	-	配列	-	M	次のように来歴登録識別URLをキーとして情報を格納する "register_provenance:"[{ ・・・ }, {}・・・・]
16			url	来歴登録対象URL	ftp://example.com:8080/open	文字列	-	M	リソースURLに含まれている、来歴登録確認を適用する対象のURL部分を記載する 使用可能文字は半角英数、一部URLとして使用可能な記号（ハイフン、アンダーバーなど）、最大文字数は255字
17			tenant	テナント	open	文字列	-	O	テナントの指定が必要である場合、記載する
18			servicepath	サービスパス	/open	文字列	-	O	サービスパスの指定が必要である場合、記載する
19			enable	来歴登録設定	false	真偽値	-	M	来歴登録の有効・無効を管理する値を記載する true:有効 false:無効

記述例

```
{
   "ngsi_auth" : [
      {
         "domain" : "example.com:PORT",
         "auth" : "input API key."
      }
   ],
   "authorization": [
      {
         "url" : "https://example.com:8080/v2/entities?type=auth",
         "tenant" : "auth",
         "servicepath" : "/auth",
         "enable" : true
      },
      {
         "url": "https://example.com:8080/open",
         "tenant" : "open",
```

```
            "servicepath" : "/open",
            "enable" : false
        }
    ],
    "contract_management_service": [
        {
            "url" : "https://example.com:8080/v2/entities?type=auth",
            "tenant" : "auth",
            "servicepath" : "/auth",
            "enable" : true
        },
        {
            "url": "https://example.com:8080/v2/entities?type=open",
            "tenant" : "open",
            "servicepath" : "/open",
            "enable" : false
        }
    ],
    "register_provenance": [
        {
            "url" : "https://example.com:8080/v2/entities?type=auth",
            "tenant" : "auth",
            "servicepath" : "/auth",
            "enable" : true
        },
        {
            "url": "https://example.com:8080/open",
            "tenant" : "open",
            "servicepath" : "/open",
            "enable" : false
        }
    ]
}
```

一覧

#	要素	論理名	デフォルト値	型	値域	M(必須) /O(任意)	説明
1	authorization_server_url	認可サーバアクセスURL	https://example.com	文字列	-	M	認可サーバへのアクセスURLを記載する

記述例

```
{
  "authentication_server_url" : "https://example.com"
}
```

一覧

#	要素	論理名	デフォルト値	型	値域	M(必須) /O(任意)	説明
1	LC_CTYPE	カテゴリ情報に使用するロケール	-	文字列	-	M	ファイルのロケールを設定する UTF-８固定
2	CA_BUNDLE_LOCATION	バンドルされたロケーションサーバ用CA証明	-	文字列	-	M	バンドルされているCA証明書の絶対パスを設定する
3	CERT_LOCATION	ロケーションサーバ用クライアント証明書	-	文字列	-	M	クライアント証明書の絶対パスを設定する
4	KEY_LOCATION	ロケーションサーバ用クライアント秘密鍵	-	文字列	-	M	クライアント秘密鍵の絶対パスを設定する
5	CA_BUNDLE_AUTHENTICATION	バンドルされたロケーションサーバ用CA証明	-	文字列	-	M	バンドルされているCA証明書の絶対パスを設定する
6	CERT_AUTHENTICATION	認可I/F用クライアント証明書	-	文字列	-	M	クライアント証明書の絶対パスを設定する
7	KEY_AUTHENTICATION	認可I/F用クライアント秘密鍵	-	文字列	-	M	クライアント秘密鍵の絶対パスを設定する
8	CA_BUNDLE_PROVENANCE_MANAGEMENT	バンドルされたロケーションサーバ用CA証明	-	文字列	-	M	バンドルされているCA証明書の絶対パスを設定する
9	CERT_PROVENANCE_MANAGEMENT	来歴管理I/F用クライアント証明書	-	文字列	-	M	クライアント証明書の絶対パスを設定する
10	KEY_PROVENANCE_MANAGEMENT	来歴管理I/F用クライアント秘密鍵	-	文字列	-	M	クライアント秘密鍵の絶対パスを設定する

記述例

```
CA_BUNDLE_LOCATION=/etc/docker/certs.d/ca.pem
CERT_LOCATION=/etc/docker/certs.d/client.crt
KEY_LOCATION=/etc/docker/certs.d/client.key
CA_BUNDLE_AUTHORIZATION=/etc/docker/certs.d/ca.pem
CERT_AUTHORIZATION=/etc/docker/certs.d/client.crt
KEY_AUTHORIZATION=/etc/docker/certs.d/client.key
CA_BUNDLE_PROVENANCE_MANAGEMENT=/etc/docker/certs.d/ca.pem
CERT_PROVENANCE_MANAGEMENT=/etc/docker/certs.d/client.crt
KEY_PROVENANCE_MANAGEMENT=/etc/docker/certs.d/client.key
```

一覧

#	要素	論理名	デフォルト値	型	値域	M(必須) /O(任意)	説明
1	public_ckan_url	横断検索サイト URL	https://example.com/backend/api/package_search	文字列	-	M	横断検索サイトのアクセスURLを記載する

記述例

```
{
    "ckan_url" : "http://search.ckan.jp/backend/api/package_search"
}
```

※ 横断検索サイト URLは基本的に上記で固定となる。

一覧

#	要素	論理名	デフォルト値	型	値域	M(必須) /O(任意)	説明

1	connector_location	コネクタロケーション	-	オブジェクト	-	M	"connector_location:"{
							(提供者コネクタID1): {
							・・・
							},
							(提供者コネクタID2)・・・・
							}

2	提供者コネクタID	提供者コネクタID	-	文字列	-	M	利用可能な提供者コネクタIDを記載する
3	provider_connector_url	提供者コネクタのアクセスURL	https://example.com	文字列	-	M	利用可能な提供者の提供者コネクタのアクセスURLを記載する

記述例

```
{
    "connector_location": {
        "provider_test_id": {
            "provider_connector_url": "http://example.com"
        }
    }
}
```

一覧

#	要素	論理名	デフォルト値	型	値域	M(必須) /O(任意)	説明
1	consumer_connector_id	受領者コネクタID	test_consumer_connector_id	文字列	-	M	受領者コネクタのID(keycloak)を記載する
2	location_service_url	ロケーションサービスのURL	https://example.com	文字列	-	M	ロケーションサービスのURLを記載する
3	trace_log_enable	トレースログ出力可否	true	真偽値	-	M	トレースログ出力可否の設定を記載する true:有効 false:無効

記述例

```
{
   "consumer_connector_id" : "test_consumer_connector_id",
   "consumer_connector_secret" : "test_consumer_connector_secret",
   "location_service_url" : "https://example.com"
   "trace_log_enable" : true
}
```

一覧

#	要素	論理名	デフォルト値	型	値域	M(必須) /O(任意)	説明
1	basic_auth	ベーシック認証情報	-	配列	-	M	次のようにドメイン名をキーとして情報を格納する "basic_auth:"[{ ・・・ }, {}・・・・]
2	ドメイン情報		-	オブジェクト	-	O	BASIC認証対象のドメインを記載する
3	domain	ドメイン	sample.co.jp:8080	文字列	-	M	ベーシック認証を行うドメインを記載する デフォルトのポートを使用しない場合は:ポート番号の指定も必要
4	basic_id	ベーシック認証のID	anonymous	文字列	-	M	ベーシック認証時に利用するIDを記載する 利用者側で設定する場合は固定値"anonymous"を設定
5	basic_pass	ベーシック認証の パスワード	anonymous	文字列	-	M	ベーシック認証時に利用するパスワードを記載する 利用者側で設定する場合は固定値"anonymous"を設定

記述例

```
{
   "basic_auth": [
      {
         "domain" : "sample.co.jp:8080",
         "basic_id" : "anonymous",
         "basic_pass" : "anonymous"
      },
      {
         .....
      }
   ]
}
```

一覧

#	要素	論理名	デフォルト値	型	値域	M(必須) /O(任意)	説明

1	ftp_auth	FTP認証情報	-	オブジェクト	-	M	次のようにFTP認証情報を記載する
							合致するドメインが存在しない場合は
							"anonymous:anonymous"でログインする
							{
							"ftp_auth": [
							{
							....
							},
							{
							....
							}
							]

2	domain	ドメイン	sample.co.jp:443	文字列	-	M	FTP接続するドメインを記載する デフォルトのポートを使用しない場合は:ポート番号の指定も必要
3	ftp_id	FTP接続のID	anonymous	文字列	-	M	FTP接続時に利用するIDを記載する 利用者側で設定する場合は固定値"anonymous"を設定
4	ftp_pass	FTP接続のパスワード	anonymous	文字列	-	M	FTP接続時に利用するパスワードを記載する 利用者側で設定する場合は固定値"anonymous"を設定

記述例

```
{
   "ftp_auth": [
      {
      "domain" : "sample.co.jp:443",
      "ftp_id" : "anonymous",
      "ftp_pass" : "anonymous"
      },
      {
         .....
      }
   ]
}
```

一覧

#	要素	論理名	デフォルト値	型	値域	M(必須) /O(任意)	説明

1	ngsi_auth	NGSI アクセストークン	-	オブジェクト	-	M	NGSIアクセストークンを配列で格納する
							"ngsi_auth" :[
							{
							・・・
							},
							{・・・・
							}

2	domain	ドメイン	example.com:PORT	文字列	-	O	データ管理サーバ(NGSI)のドメインURLを記載する
3	auth	アクセストークン	input API key.	文字列	-	M	データ管理サーバ(NGSI)へAPIアクセスするためのアクセストークン（Authorization）を記載する

記述例

```
{
   "ngsi_auth" : [
      {
         "domain" : "example.com:PORT",
         "auth" : "input API key."
      }
   ]
}
```

一覧

#	要素	論理名	デフォルト値	型	値域	M(必須) /O(任意)	説明
1	provenance_management_api_url	来歴管理I/Fへのアクセスのベースurl	https://example.com/v2	文字列	-	M	来歴管理I/Fへのアクセスurlを記載する 来歴管理エージェントへのurlを設定する

記述例

```
{
   "provenance_management_api_url" : "https://example.com/v2"
}
```

一覧

#	要素	論理名	デフォルト値	型	値域	M(必須) /O(任意)	説明
1	LC_CTYPE	カテゴリ情報に使用するロケール	-	文字列	-	M	ファイルのロケールを設定する UTF-８固定
2	CA_BUNDLE_CATALOG_SEARCH	バンドルされたロケーションサーバ用CA証明書	-	文字列	-	M	バンドルされているCA証明書の絶対パスを設定する
3	CERT_CATALOG_SEARCH	カタログ検索I/F用クライアント証明書	-	文字列	-	M	クライアント証明書の絶対パスを設定する
4	KEY_CATALOG_SEARCH	カタログ検索I/F用クライアント秘密鍵	-	文字列	-	M	クライアント秘密鍵の絶対パスを設定する
5	CA_BUNDLE_LOCATION	バンドルされたロケーションサーバ用CA証明書	-	文字列	-	M	バンドルされているCA証明書の絶対パスを設定する
6	CERT_LOCATION	ロケーションサーバ用クライアント証明書	-	文字列	-	M	クライアント証明書の絶対パスを設定する
7	KEY_LOCATION	ロケーションサーバ用クライアント秘密鍵	-	文字列	-	M	クライアント秘密鍵の絶対パスを設定する
8	CA_BUNDLE_DATA_EXCHANGE	バンドルされたロケーションサーバ用CA証明書	-	文字列	-	M	バンドルされているCA証明書の絶対パスを設定する
9	CERT_DATA_EXCHANGE	データ交換I/F用クライアント証明書	-	文字列	-	M	クライアント証明書の絶対パスを設定する
10	KEY_DATA_EXCHANGE	データ交換I/F用クライアント秘密鍵	-	文字列	-	M	クライアント秘密鍵の絶対パスを設定する
11	CA_BUNDLE_AUTHENTICATION	バンドルされたロケーションサーバ用CA証明書	-	文字列	-	M	バンドルされているCA証明書の絶対パスを設定する
12	CERT_AUTHENTICATION	認証I/F用クライアント証明書	-	文字列	-	M	クライアント証明書の絶対パスを設定する
13	KEY_AUTHENTICATION	認証I/F用クライアント秘密鍵	-	文字列	-	M	クライアント秘密鍵の絶対パスを設定する
14	CA_BUNDLE_PROVENANCE_MANAGEMENT	バンドルされたロケーションサーバ用CA証明書	-	文字列	-	M	バンドルされているCA証明書の絶対パスを設定する
15	CERT_PROVENANCE_MANAGEMENT	来歴管理I/F用クライアント証明書	-	文字列	-	M	クライアント証明書の絶対パスを設定する
16	KEY_PROVENANCE_MANAGEMENT	来歴管理I/F用クライアント秘密鍵	-	文字列	-	M	クライアント秘密鍵の絶対パスを設定する

記述例

```
LC_CTYPE=C.UTF-8
CA_BUNDLE_CATALOG_SEARCH=/etc/docker/certs.d/ca.pem
CERT_CATALOG_SEARCH=/etc/docker/certs.d/client.crt
KEY_CATALOG_SEARCH=/etc/docker/certs.d/client.key
CA_BUNDLE_LOCATION=/etc/docker/certs.d/ca.pem
CERT_LOCATION=/etc/docker/certs.d/client.crt
KEY_LOCATION=/etc/docker/certs.d/client.key
CA_BUNDLE_DATA_EXCHANGE=/etc/docker/certs.d/ca.pem
CERT_DATA_EXCHANGE=/etc/docker/certs.d/client.crt
KEY_DATA_EXCHANGE=/etc/docker/certs.d/client.key
CA_BUNDLE_AUTHENTICATION=/etc/docker/certs.d/ca.pem
CERT_AUTHENTICATION=/etc/docker/certs.d/client.crt
KEY_AUTHENTICATION=/etc/docker/certs.d/client.key
CA_BUNDLE_PROVENANCE_MANAGEMENT=/etc/docker/certs.d/ca.pem
CERT_PROVENANCE_MANAGEMENT=/etc/docker/certs.d/client.crt
KEY_PROVENANCE_MANAGEMENT=/etc/docker/certs.d/client.key
```