



# RDEデータ構造化と データセットテンプレート解説

リリース1.0.2

国立研究開発法人 物質・材料研究機構

Copyright © 2024 National Institute for Materials Science. All rights Reserved.

## 目次

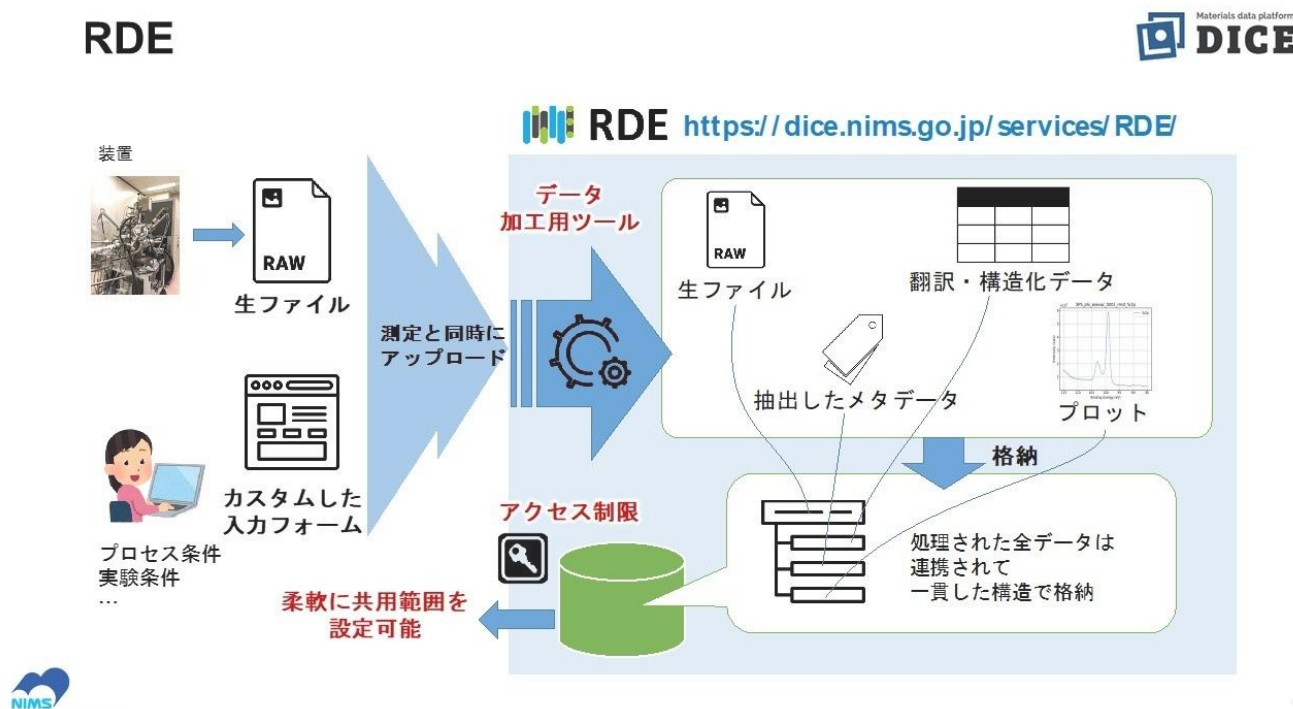
---

1. RDEのデータ登録の概要	3
2. RDEデータセットテンプレート開発環境	4
3. 付録	7
4. RDEを構成するオブジェクト	8
5. RDEのデータ登録とデータ構造化処理のしくみ	10
5.1 データセットとデータセットテンプレート	12
5.2 テンプレートファイルの役割	13
5.3 データのダウンロード	18
6. RDEにおけるデータ構造化処理について	21
6.1 データ構造化処理の入力と出力	22
6.2 構造化処理で取り扱う各種ファイルについての説明	26
7. データセットテンプレート詳細	38
7.1 テンプレートファイルと役割	38
7.2 メタデータ定義[metadata-def.json]	64
7.3 データセットカタログテンプレート[catalog.schema.json]	67
7.4 バッチ定義[jobs.template.yaml,batch.yaml]	69
7.5 tasksupport	70
7.6 dockerコンテナイメージ	70
8. データセットテンプレートのバリデーション	71
8.1 RDEToolKitにおけるJSONファイルのバリデーション	71
9. 付録ファイル	79
9.1 ファイルの提供	79
10. 次のステップ	80

# 1. RDEのデータ登録の概要

RDEの目的は構造化されたデータの収集であり、データを構造化することで得られたメタデータは機械学習などの特徴量項目などに利用することを想定しています。それを実現する方法として**データセットテンプレート**を用いたデータ種類ごとの処理をシステムに随時追加できるしくみを採用しました。実験結果などから得られた装置出力のデータファイルは装置や実験手法などにより様々な種類があり、それぞれに対応した構造化処理プログラムが必要となります。RDEが様々な登録データを受け入れるためには多様な**データセットテンプレート**が必要となり、特に構造化処理プログラムがその要となります。

下図はRDEにおけるデータ登録の概要を示したものです。



本書ではデータ登録の要となる**データセットテンプレート**とそれを構成する各種ファイル、構造化処理プログラムで出力すべきファイルなどについて説明します。なお構造化処理プログラムの開発手順、詳細については開発者向け資料を参照してください。

## 2. RDEデータセットテンプレート開発環境

---

RDEデータセットテンプレートの実行環境および技能要件は以下のとおりです。

### **RDEデータセットテンプレート開発環境**

2024/8/16 時点

**実行環境などは更新される場合があるため、提供資料がどの時点のシステムを対象にしたものか確認をしてください**

## 1. RDE構造化処理用pythonパッケージ

- RDEToolKit v.1.0.0
- 対応pythonバージョン v.3.11
- <https://pydigger.com/pypi/rdetoolkit>
- install: pip install rdetoolkit

## 2. 主たる開発言語

- python v.3.11
- RDEToolKitの利用を推奨

## 3. Dockerおよびコンテナ

- RDEシステム上での処理実行はDockerコンテナによる
- RDEシステムの構造化処理ホストのdocker環境は以下の通り
  - ホストOS : ubuntu server 20.04-lts:latest(updateはAzureが随時実行)
  - Docker version 20.10.24, build 297e128
  - CPUアーキテクチャ x86-64 (linux/amd64, linux/i386)
- pythonを用いるコンテナを作る場合のbaseはpython:3.11-bookwormを基本とする

## 4. データセットテンプレート作成ツール

- テンプレート生成ツール
- 対象エクセルバージョン Microsoft Excel for Microsoft 365動作検証済み
- 生成ツールはpythonで作成、WindowsおよびMacで実行可
- テンプレート編集ツール VScode Extensionで提供

## 5. 構造化処理ホスト

- vcpu : 4
- vmem : 16GB
- 構造化処理使用可能ディスク総量 : 10GB
- 上記仕様のVMを標準で2台構成

## 6. 構造化処理バッチジョブ管理ツール

- Azure Batch Shipyard

現状のAzureリソースなど情報反映済み(5/27)  
2025/4までにホストOSはUbuntu 22.\*に変更予定

## 3. 付録

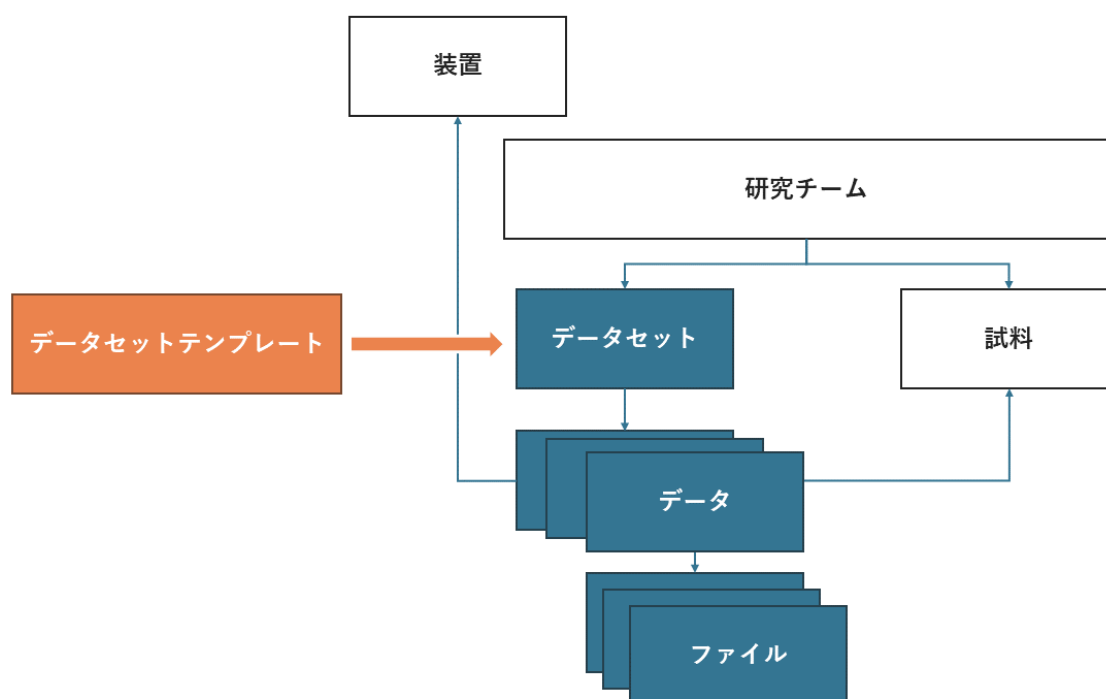
---

- 付録の章参照
- 本資料で掲載したテンプレートファイルなど、下記のを提供しています
- rde-datasettemplate-instructions\_appendix.zipファイルを取得、展開してご利用ください

## 4. RDEを構成するオブジェクト

RDEは研究活動における測定装置出力のデータファイルなど登録するために開発されたデータベースシステムです。特徴としてはデータ登録の過程でデータ固有の処理を実行し機械学習などで利用できるメタデータを抽出・保存することができる仕組みが用意されています。

研究チームが測定装置から得られたデータを、データ種類ごとにデータを登録する過程を想定したオブジェクトで構成されています。データはある装置である試料に対する測定をした結果得られるものとしてモデル化されています。以下の図はRDEを構成するオブジェクトの関係を示したものです。



登録データのアクセス権を管理する単位として**研究チーム**が用意されています。データの登録先となる**データセット**は研究チームの下に複数開設することができます。登録したデータは**データセット**内に**データ**として登録されます(データを一般的用語と区別するため**データ**と呼ぶことがあります)。**データ**には測定した**装置**、**試料**を連携することができます。**試料**は**研究チーム**の元で管理されるため、研究チーム内の複数のデータに割り当てることができます。**データ**は複数のファイルで構成されます。

RDEの特徴としてデータを登録した時点で**データ構造化処理**と呼ばれるプログラムを実行してメタデータの抽出などを行える機能があります。また、データセットごとに入力項目を変えることができます。取り扱い項目の定義ファイルやDockerコンテナイメージで構成された**データセットテンプレート**を用意することで多様なデータ種別に対応したデータセットを



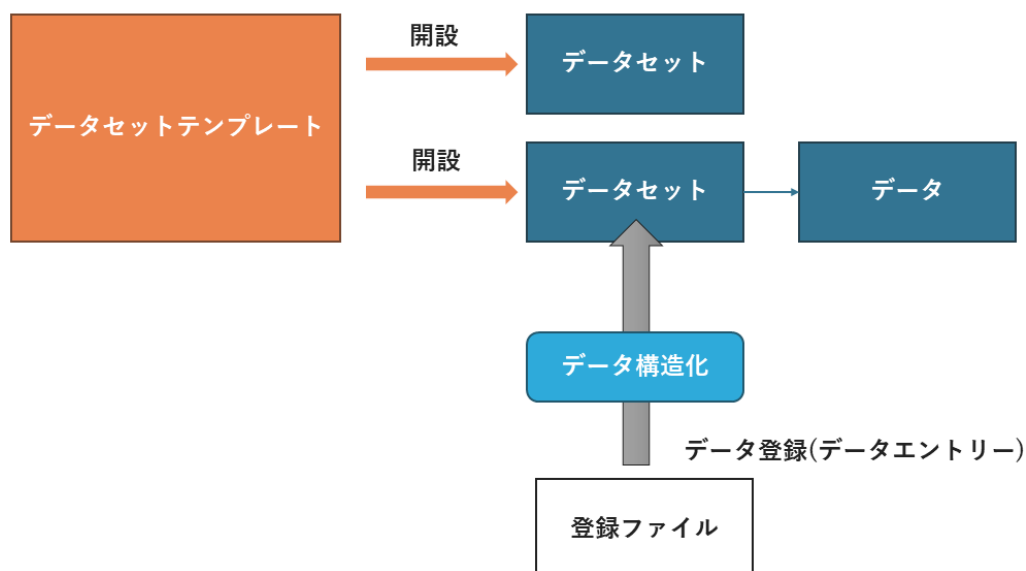
開設することができるようになっています。

本書では、**データセットテンプレート**を開発するために必要なRDEシステムの用語、機能などを説明します。

## 5. RDEのデータ登録とデータ構造化処理のしくみ

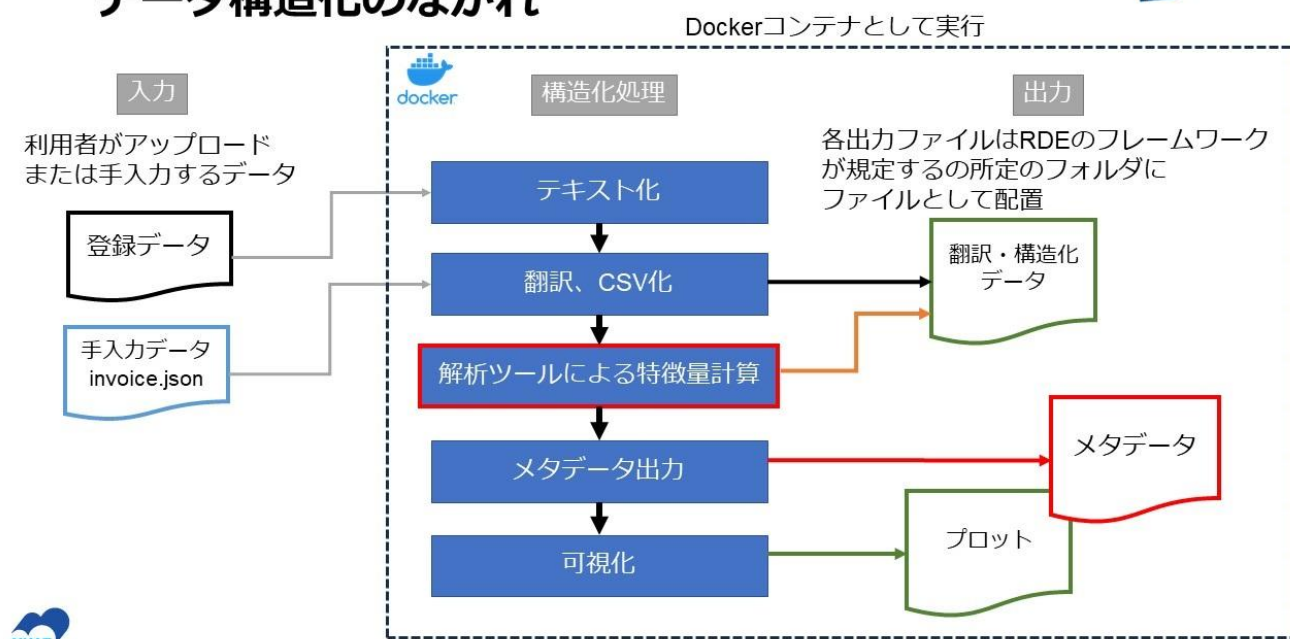
RDEは予め用意したデータセットにデータを登録するという方式を採用しています。データセットはデータセットテンプレートを元に作られます(これをデータセットの開設と呼んでいます)。

下図はRDEにおけるデータ登録の概要を示した図です。データセットテンプレートからデータセットを開設、その後データ登録時にデータ構造化を経てデータセットにデータとして登録されることを示しています。



下図はデータ構造化の部分をもう少し詳細に示しています。典型的な構造化処理はこの図のような処理を行っています。

## データ構造化のながれ

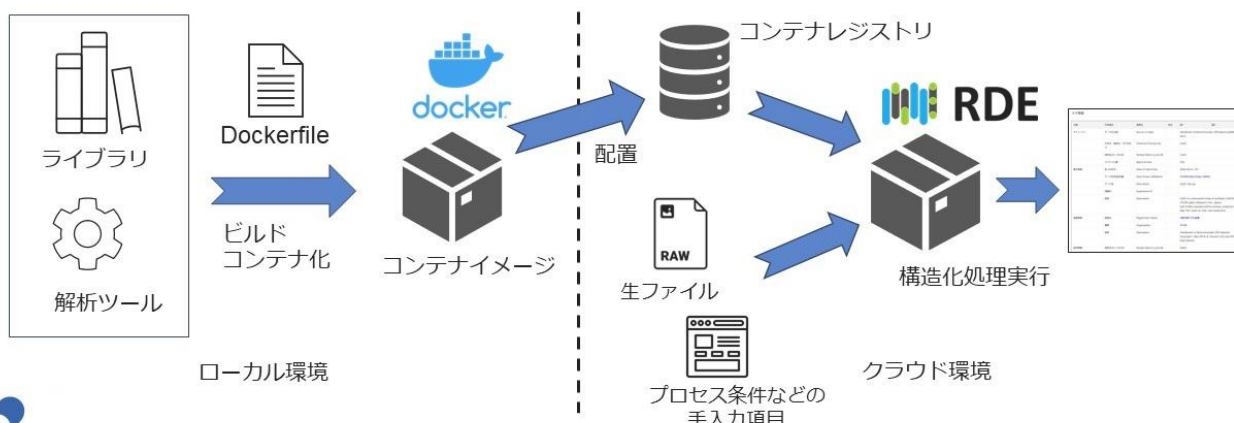


また、本書では詳細は説明RDEの構造化処理はコンテナ技術を利用しており、構造化処理プログラムはDockerコンテナとしてシステムに登録しておき、随時利用する方式としています。この方式のため、システムに随時構造化処理を登録、更新することが可能となっています。なお、DockerコンテナはRDEではデータセットテンプレートの構成要素となります。

## 構造化処理: Dockerを用いた解析ツールの共用



- Docker = コンテナ仮想化を用いてアプリケーションを開発・配置・実行するためのオープンプラットフォーム
- 手元の環境をコンテナ化してRDEの構造化処理として利用可能

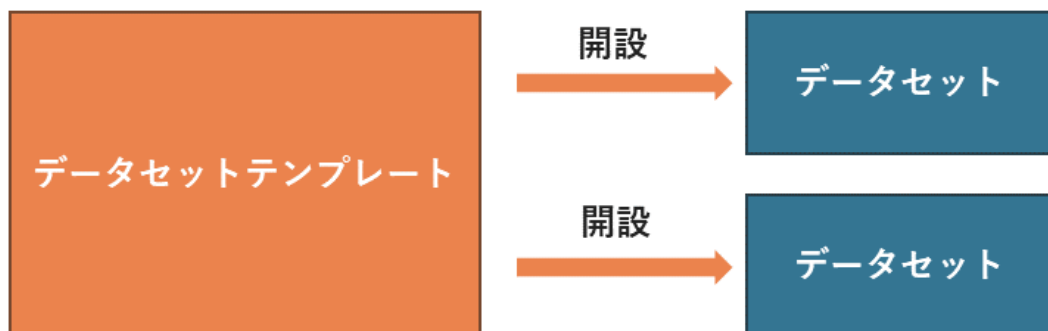


**データセット**は**データセットテンプレート**というひな形(テンプレート)を使って開設されます。データセットテンプレートは、装置出力のデータ型式などに合わせて作成されたもので、登録するデータファイルを構造化処理プログラムやその他登録する情報を定義したファイルで構成されています。登録するデータファイルは様々な形式が想定されますが、登録する対象に合わせてデータセットテンプレートを用意しておくことで様々なデータを取り込むことができるようになっています。RDEに取り込めるデータファイルの多様性はデータセットテンプレートの種類を増やすことで実現することができます。

また、RDEでは登録データから抽出、または手入力の情報をメタデータとして登録する機能があります。メタデータは機械学習のための特徴量や検索のために用いられることを想定しています。この登録データからメタデータや数値データを取り出す処理をRDEでは**データ構造化処理**と呼んでいます。データ構造化処理はデータセットテンプレートを構成する要素として重要な役割を担うものです。

## 5.1 データセットとデータセットテンプレート

---



データセットは1つのデータセットテンプレートから開設されるため、そのデータセットテンプレートがデータ構造化処理可能な同種のデータを登録する入れ物となります。例えば、ある装置からの出力結果を研究プロジェクトごとに登録したい場合は、装置出力に対応したデータセットテンプレートを使って研究プロジェクトごとにデータセットを開設して利用します。

なお、データセットテンプレートは以下のファイル(テンプレートファイルと呼ぶ)などで構成されます。 - json型式の種々の定義ファイル

- yaml形式の構造化処理バッチ処理の記述ファイル - データ構造化処理プログラムと動作環境を含むdocker container image

データセットテンプレートの開発は、これらの構成要素を作成することになります。

## 5.2 テンプレートファイルの役割

テンプレートファイルと用途などは次の通りです。

テンプレートファイル	必須	生成ファイル	用途など
invoice.schema.json	必須	invoice.json	送状スキーマ。データ受入の生成に利用される。また、invoice.jsonのバリデーションに利用される。ただし、invoice.schema.jsonはinvoice.jsonの全項目を網羅していない。構造化処理プログラムでも利用。
metadata-def.json	必須	metadata.json	メタデータ定義。metadata.jsonの項目ごとの定義に利用される。JSONスキーマではないためバリデーションには利用できない。構造化処理プログラムでも利用。RDEToolKitはmetadata-def.jsonからmetadata.jsonを生成することができる。
catalog.schema.json	必須	catalog.json	カタログスキーマ。データセット詳細のカタログ画面に利用される
jobs.template.yaml	必須	-	バッチ定義。構造化処理バッチの処理内容を記述する
batch.yaml	-	-	データ構造化定義(既定のpool以外を利用する場合などに利用)
タスク補助ファイル (tasksupport)	-	-	構造化処理で利用する上記以外のファイル
Dockerコンテナイメージ	-	-	構造化処理プログラムと実行環境

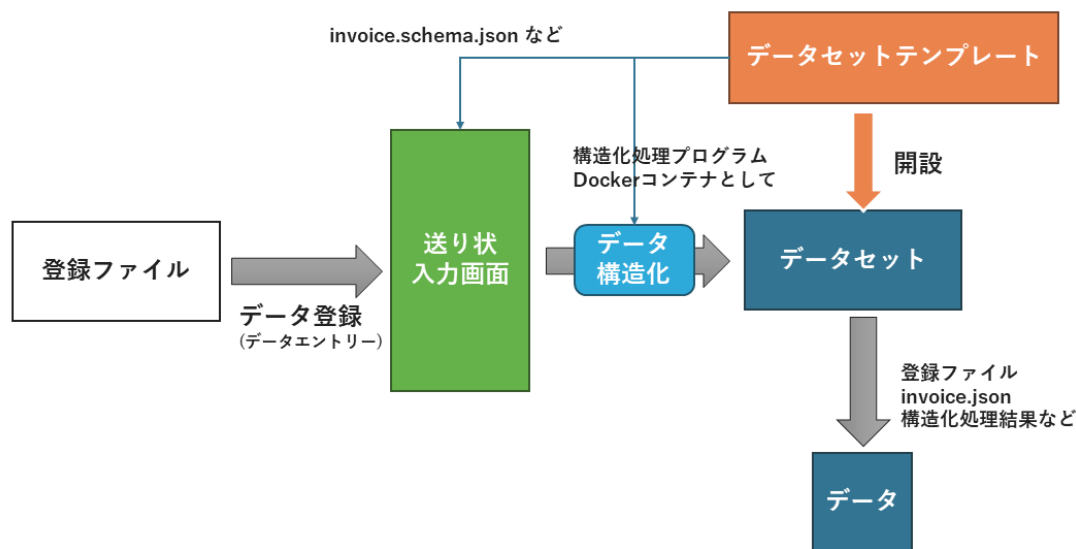
上記で一覧したテンプレートファイルはファイル名が固定されているため名前は変更することができません。ただし、tasksupportフォルダ以下に置く補助的ファイルのファイル名は任意です。

また、これらのデータセットテンプレートを開発する場合は開発しようとしているものと類似のものをサンプルとして提供する予定です。

### 5.2.1 データセットへのデータ登録と送状(invoice)

データセットにデータを登録すると、そのデータセットの開設に利用されたデータセットテンプレートによる構造化処理結果がRDEに保存されます。この保存することを**永続化**と呼ぶことがあります。

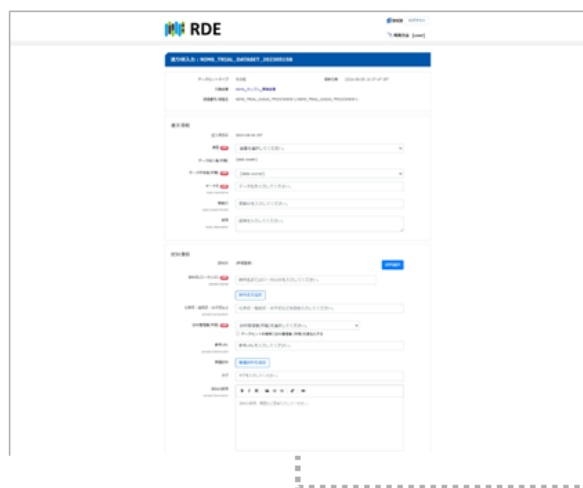
データを登録する際はそのデータに関する諸情報を入力するための**送状入力画面**を利用します。送状入力画面は登録先データセットの元となったデータセットテンプレートで定義されたものが利用されます。そのためデータセットごとに(正確にはデータセットテンプレートごとに)に入力項目を変えることができます。



送状入力項目は、システムが用意したデータの基本情報、試料情報(任意)に加え個別に定義した手入力情報を定義することができます。これは登録データから抽出できないメタデータやラボノートの項目を手入力で登録する場合を想定した機能です。なお、RDEでは、送状を**invoice**と呼ぶこともあります。

### 送状入力画面全体

RDEデータ受入アプリにおける送状入力画面は下図のような入力画面です。この事例は送状項目の基本、試料(基本+カスタム)、固有情報のすべてを定義したものです。このような画面はinvoice.schema.jsonで定義することができます。



## RDEデータ受入アプリ/送状入力画面

[illegible]

以下に送状入力画面を情報項目別に紹介します。

## 基本情報

- データ名、データ所有者などの入力欄
- システムが追加する必須項目
- invoice.schema.jsonで定義せずに出力される項目
- invoice.jsonではbasic要素として出力される

## 基本情報の表示例

送り状入力 : [dataset name]

データセットタイプ	その他	更新日時	2024-08-05 16:37:47 JST
対象装置	NIMS_サンプル_実験装置		
課題番号/課題名	[課題番号]		

基本情報

記入年月日

2024-08-07 JST

装置 必須

装置を選択してください。

データ投入者(所属)

[data loader]

データ所有者(所属) 必須

[data owner]

データ名 必須

データ名を入力してください。

実験ID

実験IDを入力してください。

説明

説明を入力してください。

## 試料情報

- 試料情報の入力欄
- 選択で利用できる項目
- 新規の試料入力、登録済み試料の選択を行う
- invoice.schema.jsonではsample要素で定義される項目
- invoice.jsonではsample要素として出力される
- 表示例の青線から上が試料基本項目、下側が追加項目

## 試料情報の表示例



試料情報

試料ID (新規登録)

試料選択

試料名(ローカルID) 必須

試料名またはローカルIDを入力してください。

試料名を追加

化学式・組成式・分子式など

化学式・組成式・分子式などを自由入力してください。

試料管理者(所属) 必須

試料管理者(所属)を選択してください。

☐ データセット共有時に試料管理者 (所属)を匿名化する

参考URL

参考URLを入力してください。

関連試料

関連試料を追加

タグ

タグを入力してください。

試料の説明

B

I

H

🔗

📄

📌

🔍

🔖

試料の説明、履歴など自由入力してください。

---

一般項目

一般名称

一般名称を入力してください

化学組成

試料分類

試料分類を入力してください

CAS番号

CAS番号を入力してください

分類別項目

半導体

名称

名称を入力してください

セラミツ

名称

名称を入力してください

有機材料

分子量

分子量を入力してください

## 固有情報

- 固有情報の入力欄
- 選択で利用できる項目
- 手入力のメタデータを入力するときに利用
- invoice.schema.jsonではcustom要素で定義される項目
- invoice.jsonではcustom要素として出力される

## 固有情報の表示例

固有情報

測定日時 必須

:JST

custom/measurement\_date

送状状数値入力値整数1

custom/invoice\_integer1

送状状数値入力値2

custom/invoice\_number2

送状文字入力値1 必須

custom/invoice\_string1

送状文字入力値2

custom/invoice\_string2

送状状選択値1

custom/invoice\_list1

権数データ登録の有無 必須

custom/is\_divided

RAWデータの共有の可否 必須

custom/is\_private\_raw

## 登録・添付ファイル

- 登録ファイル、添付ファイルを選択して仮登録する
- ファイルを仮登録後、[内容確認]、[登録開始]を実行することでデータ登録が開始される

### ファイル選択の表示例

登録ファイル

登録ファイルをドラッグアンドドロップしてください。

ファイル選択

添付ファイル

添付ファイルをドラッグアンドドロップしてください。

ファイル選択

内容確認

キャンセル

## 5.3 データのダウンロード

ここでデータをダウンロードしたときに取得できるファイルについて説明をしておきます。ダウンロード結果のファイル構造はデータ構造化処理の出力を反映したものとなっています。

RDEのデータセットに登録されたデータは、データセットまたはデータ単位でダウンロードすることができます(ユーザの権限により閲覧などの範囲は異なります)。ダウンロードはzipファイル形式のファイルで、登録データやデータ構造化処理の結果などのファイルが含まれています。

以下にデータをダウンロードしたときの例を示します。ダウンロードで得られたzipファイルを解凍しtreeコマンドで内包するフォルダ、ファイルを示します。

```
$ unzip data_20240709113208.zip
出力は略
$ tree dataset_415dd76f-82a9-4a5e-94d0-198928e3f112/
dataset_415dd76f-82a9-4a5e-94d0-198928e3f112/
├── catalog.json
├── catalog.schema.json
├── data_0011
│   ├── data.json
│   ├── filemeta.json
│   ├── invoice.json
│   ├── main_image
│   │   └── all_series.png
│   ├── meta
│   │   └── metadata.json
│   ├── other_image
│   │   ├── series1.png
│   │   └── series2.png
│   ├── raw
│   │   └── sample.data
│   └── structured
│       ├── series1.csv
│       └── series2.csv
├── invoice.schema.json
└── metadata-def.json

6 directories, 14 files
```

- RDEのデータはデータセットに紐づいているため、ダウンロード結果にはデータセットの情報と個々のデータのファイルなどが含まれています
- また、出力されたjsonファイルを読み解くための情報としてデータセットテンプレートの構成ファイルの一部が出力されます
- ダウンロードしたzipファイルを解凍するとdataset\_[データセットID]のフォルダがあり、その下にデータのファイルが出力されます(このファイル階層はデータセット一括ダウンロードとデータダウンロードで同一です)
- データはdata\_[データ番号(4桁数字)]のフォルダに出力されます
- RDEformat(後述)にはthumbnailフォルダおよびサムネイルファイルが含まれますがダウンロード結果には出力されません
- この出力例では表示されていませんが、添付ファイルがある場合はattachmentフォルダに出力されます

以下に、ダウンロード時に必ず出力される固有のファイルを示します。

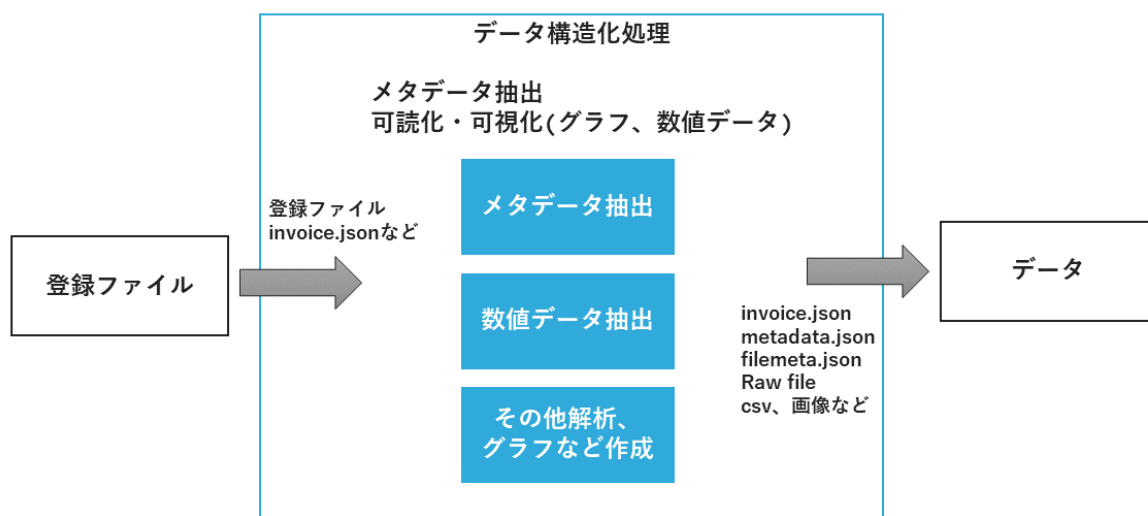
ファイル	フォルダ	説明
catalog.json	[top]	データセットのカタログ情報の出力(システムが生成)
catalog.schema.json	[top]	catalog.jsonの定義ファイル(テンプレートから取得)
invoice.schema.json	[top]	invoice.jsonの定義ファイル(テンプレートから取得)
metadata-def.json	[top]	メタデータ定義ファイル(テンプレートから取得)
data.json	data_[データ番号]	データセットおよびデータに関する基本情報(システムが生成) データごとに出力
filemeta.json	data_[データ番号]	構造化処理で作成したfilemeta.jsonにシステムが生成した情報を付加したもの。構造化処理で作成していない場合はシステムが生成する情報のみとなる データごとに出力

## 6. RDEにおけるデータ構造化処理について

データ構造化は一般的には事前に定めた構造に整形されたデータのことを示します。RDEにおけるデータ構造化は、RDEで定めたファイル保存形式、メタデータ型式などを生成することを示します。ただし、RDEで「データ構造化処理」と言う場合、データを**可読化・可視化**して利用しやすい特定のデータ形式(データの構造化)にする処理を意味します。そのためデータ構造化処理の一般的機能は、データからメタデータ一覧、グラフ、画像、数値データ(csv)などに変換する処理となっています。

データ構造化処理の結果はRDEの規定のファイル保存形式で出力することでシステムに登録することができます。このファイル保存形式を**RDEformat**と呼びます。このRDEformat型式の出力をすることがRDEでのデータ構造化処理プログラムの目的となります。

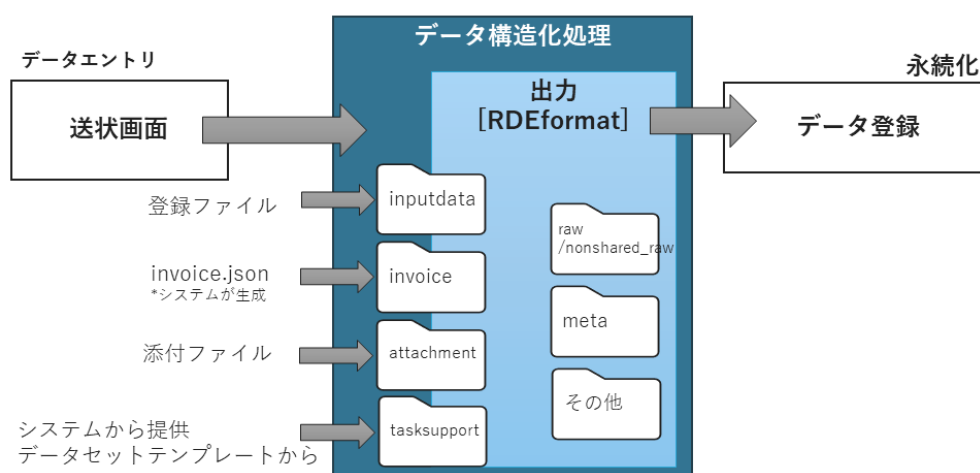
要点 RDEのデータ構造化 = データ構造化(メタデータ) + データの可読化・可視化(グラフ、数値データ) + 解析・計算



以降でデータセットテンプレートとデータ構造化処理に関わる各種ファイルについて説明します。

## 6.1 データ構造化処理の入力と出力

RDEシステム上でデータ構造化処理が実行される場合、入力情報(各種ファイル含む)はシステムが用意し、それを用いて構造化処理が実行(データセットテンプレートとして登録されたもの)されてRDEformat型式のファイルが作成され、その出力内容が登録(永続化)されるという流れとなっています。



### 6.1.1 入力

ユーザが送状画面からデータを登録した後、システム上で構造化処理を実行するためにシステムが準備する入力ファイルは以下の表の通りです。

入力種類	保存先フォルダ	ファイル
登録ファイル	inputdata	ユーザが送信したファイル一式
送状情報	invoice	invoice.json
添付ファイル	attachment	ユーザが送状画面で添付したファイル一式
テンプレートファイル	tasksupport	テンプレート登録時にtasksupportに保存したファイル

これらのファイルはコンテナ上で構造化処理プログラムが実行される際にワーキングフォルダのdataフォルダ以下に保存されます。データ構造化処理プログラムはこれらを入力ファイルとして処理を実行します。

## 6.1.2 出力

構造化処理の結果として出力するファイルおよびフォルダの構成(RDEformat)は以下の通りです。

なお、構造化処理の実行時には入力と出力は同じフォルダとなり。入力と出力のファイルのうち、次の表の種類のファイルが登録(永続化)対象となります。

種類	出力フォルダ	ファイル種別	内容
共有rawデータ	raw	rawデータファイル	inputdataのファイルから共有可能なファイルを保存
非共有rawデータ	nonshared_raw	非共有rawデータファイル	inputdataのファイルから非共有とするファイルを保存。研究チーム以外の表示、ダウンロードが不可となる。
メタデータ	metadata	主要パラメータ情報ファイル	metadata.jsonを配置
ファイルメタ	[top]	--	filemeta.json(任意) ファイル一覧に非表示
送状	invoice	--	invoice.json(入力からの引継ぎ) ファイル一覧に非表示
代表画像	main_image	画像ファイル	画像、グラフなど 複数ある場合はファイル名でソートした1番目のファイルを採用する。
画像	other_image	画像ファイル	画像、グラフなど(画像は区分けをしているがアプリケーションではすべて表示対象となるためシステム上の制約はありません)
thumbnail	thumbnail	--	データー一覧で利用されるサムネイル画像。複数あった場合は名前の順でソートした一番目が採用される。画像形式 jpg、png ファイル一覧非表示、ダウンロードなし
構造化ファイル	structured	構造化ファイル	構造化処理で生成されたその他ファイル
添付ファイル	attachment	添付ファイル	入力で用意された添付ファイル(入力からの引継ぎ)
(複数登録時に利用)	divided	--	1エントリで複数データ登録をする場合に利用。このフォルダの下に0001など4桁の数字のフォルダを作り、RDEformat型式のフォルダを作成する

### 画像に関する留意点

- RDEのアプリケーション(ブラウザで利用)ではプレビュー可能な画像の形式が限定されています
- 指定拡張子以外の場合は存在しないことを示す画像が表示されます
- 画像ファイルの指定拡張子は以下の表の通りです。

## 画像ファイルの指定拡張子

指定拡張子	備考
gif	
jpeg	
jpg	
png	
bmp	Webコンテンツとして推奨されている画像形式対象外であるため、Webブラウザの種類によって表示できない可能性がある。
ds_store	Apple OSで自動作成されるファイル
tif	

### 6.1.3 構造化処理の出力結果とデータ詳細画面での扱いについて

構造化処理の出力結果とデータ登録後のアプリケーションでの扱いについて以下の通りです。

#### データ一覧画面

thumbnailを表示  
画像がない場合は「No Image」と表示

The screenshot displays a data list interface with three entries. Each entry consists of a thumbnail image (a line graph) and a block of text containing metadata. The first entry is highlighted with a red box. A green callout box points to the thumbnails with the text 'thumbnailを表示 画像がない場合は「No Image」と表示'.

データ所有者(所属) [data owner] (NIMS)	試料名(ローカルID) NIMS_TRIAL_試料	説明	タグ/ソノミー [CAS番号]	登録日時	データID
[data owner] (NIMS)	NIMS_TRIAL_試料		[CAS番号]	2024-07-26 14:55:15 JST	06d75a5-5d68-4740-9695-0561aaa0905
[data owner] (NIMS)	NIMS_TRIAL_試料		[CAS番号]	2024-07-26 14:55:15 JST	06d75a5-5d68-4740-9695-0561aaa0905
[data owner] (NIMS)	NIMS_TRIAL_試料		[CAS番号]	2024-07-25 14:02:22 JST	2ec8b7d5-8086-4c78-936d-8f95769017b



## データ詳細画面/概要

代表画像、画像が表示される



invoice.json、metadata.jsonの内容の一覧表示  
日本語名などの情報は  
invoice.schema.json、metadata-  
def.jsonから取得

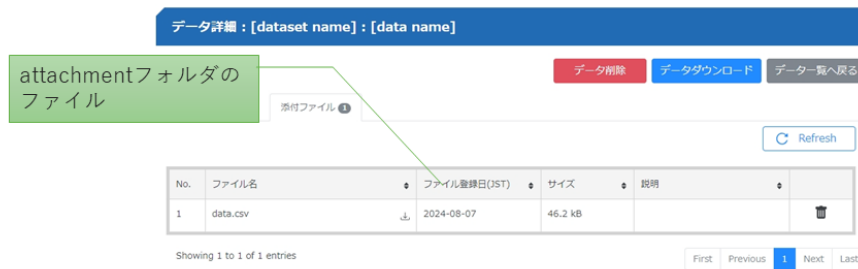
## データ詳細画面/ファイル

RDEformatの種類ごとに  
一覧される

No.	ファイル種類	ファイル名	ファイル登録日(JST)	サイズ
1	rawデータファイル	sample.data	2024-07-26	413 B
2	主要パラメータ情報ファイル	metadata.json	2024-07-26	2.31 kB
3	構造化ファイル	series1.csv	2024-07-26	170 B
4	構造化ファイル	series2.csv	2024-07-26	76 B
5	構造化ファイル	series3.csv	2024-07-26	130 B
6	日本語画像ファイル	rl_series.png	2024-07-26	25.9 kB
7	画像ファイル	series1.png	2024-07-26	19.9 kB
8	画像ファイル	series3.png	2024-07-26	21.3 kB
9	画像ファイル	series2.png	2024-07-26	21.2 kB

プレビュー可能な  
画像形式の場合のマーク

## データ詳細画面/添付ファイル



## 6.2 構造化処理で取り扱う各種ファイルについての説明

以降で構造化処理で取り扱う各種ファイルの詳細について説明します。

### 6.2.1 メタデータ

メタデータ(metadata)、一般的には、あるデータの属性や関連する情報を記述したデータのことを指します。データ自身が内包している情報を構造化処理によってメタデータファイルとして外に出すことでデータを検索しやすくなることができます。RDEでは一般的なメタデータの使い方に加えて、測定データに含まれる、または解析して得られたメタデータを機械学習の特徴量や教師データとして利用することを想定したものを含めてメタデータと呼んでいます。

RDEが取り扱うメタデータは大別すると以下の3種類が用意されています。なお、RDEのメタデータはjson型式で出力されます。

種類	説明
invoice	手入力情報から得られるメタデータ。invoice.jsonから取得する。データの基本情報、装置、試料、手入力情報を扱う
metadata	装置出力データなど登録データファイルから構造化処理により得られたメタデータ
filemeta	登録ファイルに関するメタデータ

以下に各種類ごとに説明します。

## invoice

- invoiceメタデータはinvoice.jsonから取得する
- invoice.jsonというjson型式のファイルとして出力されます
- invoice.jsonは、データの基本情報、試料、装置、手入力情報(固有情報)を取り扱います
- invoice.jsonはテンプレートファイルの1つinvoice.schema.jsonというJSONスキーマファイルで定義されます
- invoice.schema.jsonは送状画面の生成やinvoice.jsonの形式確認に利用されます
- invoice.schema.jsonはinvoice.jsonのJSONスキーマとして利用できるが、basic要素の定義がないため検証ができない
- データ登録する際にシステムがinvoice.jsonを生成し、構造化処理時にシステムから提供されデータ登録時にシステムに保存されます
- データの永続化登録時にinvoice.jsonの内容はinvoice.schema.jsonによるバリデーションが行われ、不整合があれば失敗します。
- なお、invoice.jsonは構造化処理において変更可能
- 例えば、データ名を構造化処理プログラムで指定し直すことができます

invoiceメタデータは以下の3つの情報の集合から構成されます。

1. basic
2. データの基本情報を記述する
3. データ名、データ所有者、装置の情報
4. システムが生成します
5. 必須項目です
6. sample
7. 試料に関する情報
8. 定義された情報項目の記述
9. sampleは任意要素です
10. custom
11. 固有メタデータとも呼ばれる
12. 手入力メタデータがある場合に記述する
13. customは任意要素です

以下はinvoice.jsonの事例です。

```
{
  "datasetId" : "[データセットID]",
  "basic" : {
    "dateSubmitted" : "[受付年月日YYYY-MM-DD]",
    "dataOwnerId" : "[データ所有者のDICE ID]",
    "dataName" : "[データ名]",
    "instrumentId" : "[装置ID]",
    "experimentId" : "[実験ID]",
    "description" : "[説明]"
  },
  "custom" : {
    "key1" : "value1",
    "key2" : 999
  },
  "sample" : {
    "sampleId" : "[試料ID]",
    "names" : [ "[試料名]" ],
    "composition" : null,
    "referenceUrl" : null,
    "description" : "[説明]",
    "generalAttributes" : [ {
      "termId" : "[用語ID]",
      "value" : "[値]"
    } ],
    "ownerId" : "[試料所有者のDICE ID]"
  }
}
```

```
}
}
```

invoice.jsonのスキーマ定義については「データセットテンプレートのバリデーション」を参照

上記のcustom項目のinvoice.schema.jsonは以下の通り。

invoice.schema.jsonはinvoice.jsonのJSONスキーマであり、JSONのバリデーションに利用できます。

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://rde.nims.go.jp/rde/dataset-templates/simple-data-registration_datasettemplate/invoice.schema.json",
  "description": "データセットテンプレート",
  "type": "object",
  "required": [
    "sample",
    "custom"
  ],
  "properties": {
    "custom": {
      "type": "object",
      "label": {
        "ja": "固有情報",
        "en": "Custom Information"
      },
      "required": [],
      "properties": {
        "key1": {
          "label": {
            "ja": "キー1",
            "en": "key1"
          },
          "type": "string"
        },
        "key2": {
          "label": {
            "ja": "キー2",
            "en": "key2"
          },
          "type": "number"
        }
      }
    }
  }
}
```

## 6.2.2 metadata

- 登録データなどから構造化処理で抽出したメタデータ
- JSON型式で作成する
- システムではファイル名metadata.jsonとして出力する

- metadata.jsonはテンプレートファイルの1つmetadata-def.json(メタデータ定義ファイルと呼ぶ)で出力項目を定義する
- ただしmetadata-def.jsonはJSONスキーマではないためmetadata.jsonのバリデーションには利用できない
- 単一出力項目はconstant要素以下に出力する
- 複数出力項目はvariable要素(配列)以下に出力する(データ系列ごとにメタデータの値が異なる場合に利用)

## メタデータの項目の定義について

メタデータ(metadata.json)の定義はメタデータ定義(metadata-def.json)に記述します。メタデータ定義はJSONスキーマではないためmetadata.jsonのバリデーションに用いることはできません。ただし、RDEtoolKitの機能を利用してメタデータを操作するとメタデータ定義による評価を行うことができます。

メタデータの実例からメタデータとメタデータ定義の関係を説明します。

メタデータ定義でのメタ項目の定義 - メタ項目の項目名が「XrayBeamDiameter」。データ由来の項目名としている。 - データ型はnumber型 - 共通単位があるため"um"として設定

```
"XrayBeamDiameter": {
  "name": {
    "ja": "X線ビーム径",
    "en": "Xray Beam Diameter"
  },
  "schema": {
    "type": "number"
  },
  "unit": "um"
},
```

上記で定義したメタ項目の出力例 - 値はvalueで指定 - unitはメタデータ定義に設定されているが、個別でも設定可。この例では同じ単位だが異なる単位とする場合に利用する。

```
"XrayBeamDiameter": {
  "value": 100.0,
  "unit": "um"
},
```

以下にmetadata.jsonの出力事例を示します。

```
{
  "constant": {
```

```

"common.data_origin": {"value": "experiments"},
"common.technical_category": {"value": "measurement"},
"measurement.method_category": {"value": "分光法"},
"measurement.method_sub_category": {"value": "オージェ電子分光法"},
"measurement.analysis_field": {"value": "化学状態, 電子の性質, 不純物分析, 定性分析"},
"measurement.measurement_environment": {"value": "真空中"},
"measurement.measured_date": {"value": "2009-07-13"},
"operation_date_time_year": {"value": 2009},
"operation_date_time_month": {"value": 7},
"operation_date_time_day": {"value": 13},
"operation_date_time_hour": {"value": 10},
"operation_date_time_minute": {"value": 41},
"operation_date_time_second": {"value": 29},
"probe_energy": {"value": "10.00", "unit": "keV"},
"probe_current": {"value": "3.030x10-8", "unit": "A"},
省略
},
"variable": [
  {
    "species_label_transitions": {"value": "AlLVV"},
    "abscissa_start": {"value": "42.00", "unit": "eV"},
    "abscissa_end": {"value": "76.00", "unit": "eV"},
    "abscissa_increment": {"value": "1.00", "unit": "eV"},
    "collection_time": {"value": "50", "unit": "ms"},
    "total_acquisition_number": {"value": "7"}
  },
  {
    "species_label_transitions": {"value": "AlKLL"},
    "abscissa_start": {"value": "1365.00", "unit": "eV"},
    "abscissa_end": {"value": "1408.00", "unit": "eV"},
    "abscissa_increment": {"value": "1.00", "unit": "eV"},
    "collection_time": {"value": "50", "unit": "ms"},
    "total_acquisition_number": {"value": "6"}
  },
  省略
]
}

```

## 6.2.3 filemeta

- filemetaは、登録する個々のファイルに対してメタデータを追加するときに作成します
- 構造化処理で作成する場合はfilemeta.jsonとして作成します
- ファイル名filemeta.jsonとしてダウンロード時に出力されます
- なお、データ構造化処理で作成しない場合は、システムが自動的に作成します
- データ構造化処理で作成したfilemeta.jsonが存在する場合は、システムが自動生成したものとマージされて出力されます

以下は、構造化処理でfilemeta.jsonを作成しなかった場合にシステムが自動生成したfilemeta.jsonの事例です。

```
{
  "filemeta": [ {
    "filename": "meta/metadata.json",
    "metadata": { }
  }, {
    "filename": "main_image/id.png",
    "metadata": { }
  }, {
    "filename": "other_image/id_AlKLL.png",
    "metadata": { }
  }, {
    "filename": "other_image/id_AlLVV.png",
    "metadata": { }
  }, {
    "filename": "other_image/id_AsLMM.png",
    "metadata": { }
  }, {
    "filename": "other_image/id_C.png",
    "metadata": { }
  }, {
    "filename": "other_image/id_GaLMM.png",
    "metadata": { }
  }, {
    "filename": "other_image/id_O.png",
    "metadata": { }
  }, {
    "filename": "raw/data",
    "metadata": { }
  }, {
    "filename": "raw/id",
    "metadata": { }
  }, {
    "filename": "raw/para",
    "metadata": { }
  }, {
    "filename": "structured/id.csv",
    "metadata": { }
  } ]
}
```

以下はfilemeta.jsonを構造化処理で作成しメタデータをファイルに付加した事例です。

```
{
  "filemetadata": [
    {
      "filename": "structured/20100130115018(wide).csv",
      "metadata": {
        "format": {
          "itemtype": "format",
          "name": {
            "ja": "ファイルフォーマット",
            "en": "file format"
          },
          "value": "csvw",
          "schema": {
            "type": "string"
          }
        }
      }
    },
  ],
}
```



```

    "column0": {
      "itemtype": "descriptor",
      "datatype": "number",
      "name": {
        "ja": "列名",
        "en": "column"
      },
      "value": "Kinetic Energy",
      "unit": "eV",
      "url": "https://en.wikipedia.org/wiki/Kinetic_energy",
      "schema": {
        "type": "string"
      }
    }
  }
}
]
}

```

## 6.2.4 タクソノミー(taxonomy)

登録したメタデータのうちデータの分類に利用する項目をタクソノミーと呼びます。タクソノミーはデータセット単位で任意に設定することができ、指定することでデータの詳細検索の条件やデータ一覧画面でのツリー表示に利用されます。タクソノミーはテンプレートに設定されるものではなく、個々のデータセットセットで随時設定できます。

タクソノミーを設定した場合のツリー表示の例は以下の通りです。

この例ではタクソノミーとして - invoice.basic.date\_submitted - invoice.basic.data\_name - invoice.basic.experiment\_id

の3項目を指定しています。

データセット編集画面でタクソノミキーを複数している場合は、各項目を半角スペース区切りで入力します。

ギャラリー表示ツリー表示

▼ CAS番号

test / data\_title

test\_20240726\_1-1 / data\_title 2

test\_20240726\_1-2 / data\_title 2

testdata20240614 / data\_title 2

test\_20240726\_1-1 / data\_title 2

14

データ所有者(所属) [data owner] (NIMS)

試料名(ローカルID) NIMS\_TRIAL\_試料

説明

登録日時 2024-07-26 14:55:15 JST

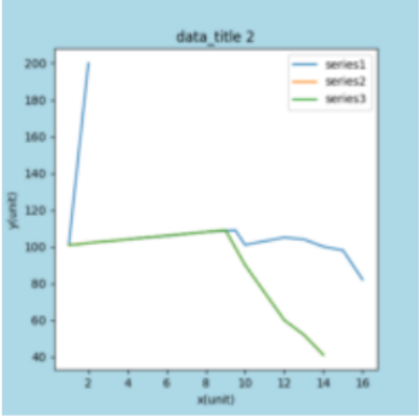
データID 0b8288c5-c2c5-492b-a18d-5cbfb0e12639

data\_title 2

series1

series2

series3



タクソノミーとして指定できるメタデータは各データセットテンプレートの invoice.schema.json、metadata-def.jsonで定義された項目および装置マスタの一部項目です。タクソノミーとして利用できる項目と記述方法を以下の表に一覧します。

基本情報(invoiceより)

ソース	項目	キー名	説明
invoice	basicのdateSubmitted	invoice.basic.date_submitted	記入年月日
invoice	basicのdata_owner	invoice.basic.data_owner	データの所有者と所属
invoice	basicのdataName	invoice.basic.data_name	データ名
invoice	basicのexperimentId	invoice.basic.experiment_id	実験ID
invoice	basicのdescription	invoice.basic.description	データの説明

- 34/80 - Copyright © 2024 National Institute for Materials Science. All rights Reserved.

## 装置情報

ソース	項目	キー名	説明
-	-	instrument.name	装置登録名
-	-	instrument.organization	装置の管理機関
-	-	instrument.description	装置の説明

## 試料情報(invoiceなどより)

ソース	項目	キー名	説明
invoice	sampleのnames[]	sample.name	試料のローカルIDの最初の一つ
invoice	sampleのcomposition	sample.composition	化学式・組成式・分子式など
invoice	sampleのdescription	sample.description	試料の説明
-	sampleのgeneralのキー名	sample.general.<キー名>	一般試料用語のキー名
-	sampleのspecificのキー名	sample.specific.<キー名>	特定試料用語のキー名

## 固有情報(invoice、metadataより)

ソース	項目	キー名	説明
invoice	customのキー名	invoice.custom.<キー名>	送状のcustom項目。定義した項目がキャメルケースの場合はケバブ形式で変換して指定する。 例: CamelCase → camel_case
metadata	constantのキー名	<キー名>	metadataのconstantの項目。定義した項目名をそのまま記述
metadata	variableのキー名	<キー名>	metadataのvariableの項目。定義した項目名をそのまま記述

## 6.2.5 送状画面から登録された測定データファイルなど取扱いについて

送状画面から登録された測定データファイルなどは、構造化処理においてinputdataフォルダに保存されています。

登録されたファイルは装置から出力された未加工のデータファイルや計算結果などが想定されます。これらをRDEではRAWファイルと呼んでいます。このRAWファイルを特定のユーザにのみダウンロード可(これを非共有と呼ぶ)とするか制限しないようにするかを構造化処

理で決定することができます(登録後は変更ができません)。以下のように所定のフォルダにコピーすることで共有の可否を変更することができます。

RAWデータファイルの共有設定	処理
共有可	rawフォルダに対象RAWファイルを保存する
共有不可	nonshared_rawフォルダに対象RAWファイルを保存する

RAWデータの共有の可否は構造化処理プログラムの作り方しだいでデータ単位に設定することもできます。通常はデータセットテンプレート単位で共有可・不可は一律としています。

### 6.2.6 画像の取扱い

RDEのアプリケーションにて、画像として表示したいファイルはmain\_image、other\_imageのいずれかのフォルダに保存します。main\_imageは代表画像、other\_imageは画像とシステム上の区分けは用意されていますが、区分けによるアプリケーションでの表示の差はわずかです。システム運用上も明確な区分けを設けていませんので、各データセットテンプレートで適宜利用してください。また、thumbnailフォルダにサムネイル画像を保存するとデータ一覧でサムネイル表示されます。

なお、サムネイル、代表画像、画像が登録されていない場合は、画像を表示するページでは"No Image"と表示される画像が利用されます。

代表画像、画像、サムネイルのアプリケーション上での動作は以下の通りです。 - 代表画像、画像は、データ詳細画面で表示される。その際、代表画像、画像でそれぞれファイル名でソートした順に表示される。 - 代表画像を一つ選ぶ場合は、名前の順でソートした一番先頭のファイルが利用される - 代表画像、画像は、データ詳細画面のファイル一覧に表示される - サムネイル画像はデータ一覧表示で利用されます - サムネイル画像はファイル一覧に表示されず、ダウンロードに含まれません

Hint: 画像の区分けについて。データの内容を画像として表示する場合は代表画像(main\_image)に、グラフなどの作成物を画像(other\_image)として区分けするという方法も考えられます。 Hint: 顕微鏡写真のtif型式などのファイルは画像とRAWデータという二つの用途を兼ねている場合があります。このようなファイルを画像としてプレビューできるようにしたい場合は、元ファイルからプレビュー可能な画像形式にしたものをmain\_imageなどに保存し、元ファイルをrawまたはnonshared\_rawに保存するという方法があります。

#### データ一覧画面(ギャラリー)の表示例(NoImageあり)

ギャラリー表示

ツリー表示

検索条件

登録日時

☐昇順

☒降順

test\_20240726\_1-2 / data\_title 2

15

No  
Image

データ所有者(所属)  
[data owner] (NIMS)

試料名(ローカルID)  
NIMS\_TRIAL\_試料

説明

タクソノミー

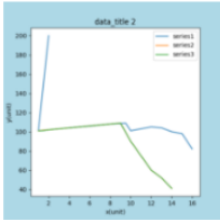
CAS番号

登録日時 2024-07-26 14:55:15 JST

データID 9be875a5-bda8-4740-9b95-05fe1aea0b95

test\_20240726\_1-1 / data\_title 2

14



データ所有者(所属)  
[data owner] (NIMS)

試料名(ローカルID)  
NIMS\_TRIAL\_試料

説明

タクソノミー

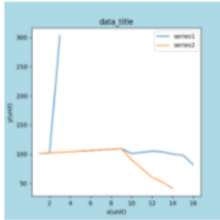
CAS番号

登録日時 2024-07-26 14:55:15 JST

データID 0b8288c5-c2c5-492b-a18d-5cbfb0e12639

test\_data

13



データ所有者(所属)  
[data owner] (NIMS)

試料名(ローカルID)  
NIMS\_TRIAL\_試料

説明

タクソノミー

CAS番号

登録日時 2024-07-25 16:02:22 JST

データID 2ec8b7c6-8086-4c78-936d-bff95769017b

## 7. データセットテンプレート詳細

### 7.1 テンプレートファイルと役割

テンプレートファイル名と用途などは次の通りです(再掲)。

テンプレートファイル	必須	対応ファイル	用途など
invoice.schema.json	必須	invoice.json	送状スキーマ。データ受入の生成に利用される。また、invoice.jsonのバリデーションに利用される。ただし、invoice.schema.jsonはinvoice.jsonの全項目を網羅していない。構造化処理プログラムでも利用。
metadata-def.json	必須	metadata.json	メタデータ定義。metadata.jsonの項目ごとの定義に利用される。JSONスキーマではないためバリデーションには利用できない。構造化処理プログラムでも利用。RDEToolKitはmetadata-def.jsonからmetadata.jsonを生成することができる。
catalog.schema.json	必須	catalog.json	カタログスキーマ。データセット詳細のカタログ画面に利用される
jobs.template.yaml	必須	-	バッチ定義。構造化処理バッチの処理内容を記述する
batch.yaml	-	-	データ構造化定義(既定のpool以外を利用する場合などに利用)
タスク補助ファイル (tasksupport)	-	-	構造化処理で利用する上記以外のファイル
Dockerコンテナイメージ	-	-	構造化処理プログラムと実行環境

上記で一覧したテンプレートファイルはファイル名が固定されているため名前は変更することができません。ただし、tasksupportフォルダ以下に置く補助的ファイルのファイル名は任意です。

以降で各テンプレートファイルの詳細を説明をします。

#### 7.1.1 送状テンプレート[invoice.schema.json]

参考:02\_永続化ファイル設計.pdf

invoice.schema.jsonについて説明します。

- 送状スキーマは送状画面の生成に利用されます
- ただし、基本情報は送状スキーマに記述しなくても画面に反映されます
- データの登録処理においてinvoice.schema.jsonの定義に合致しない場合は登録エラーとなります
- 送状スキーマの書式については別紙「」を参照してください

固有情報の定義

- 送状を用いて手入力のメタデータをと登録する場合、固有情報(custom)要素を定義します。  
- 固有情報の項目を送状テンプレートで定義すると、送状画面に欄が用意されます。  
- 以下に、定義できるメタデータの型と画面上の欄の表示の例を示します。

送状テンプレートで定義できるデータの型は以下の通りです。

type	format	enum	説明
number	-	-	実数
integer	-	-	整数
string	-	-	文字列
string	-	-	文字列 widgeでtextareaを指定すると入力がtextareaとなる
string	date	-	日付。RFC 3339のfull-date 例:2003-07-07T00:00:00+09:00、 2003-07-07T00:00:00Z
string	time	-	時刻。RFC 3339のfull-time 例:00:00:00+09:00、 00:00:00Z
string		[選択肢列挙]	文字列。入力は選択肢から選択
string	uri	-	文字列。uriを使う場合
string	uuid	-	文字列。uuidを使う場合

以上の型の定義に加え、送状画面での操作などに関わるoptionsを指定することができる。

以降に送状テンプレートのtypeとアプリケーション画面の事例を列挙します。

## string

文字列を入力する項目に利用

送状文字入力値1

文字列

custom/invoice\_string1

```
"invoice_string1": {  
  "label": {  
    "ja": "送状文字入力値1",  
    "en": "invoice_string1"  
  },  
  "type": "string"  
},
```

アプリではinput(html)として表示されます。

複数行の入力をするときの定義例

UIの設定はoptions要素で定義します。

ただし、登録時には改行コード付きの文字列として登録されます

送状文字入力値2

1行目

2行目

3行目

custom/invoice\_string2

```
"invoice_string2": {  
  "label": {  
    "ja": "送状文字入力値2",  
    "en": "invoice_string2"  
  },  
  "type": "string",  
  "options": {  
    "widget": "textarea",  
    "rows": 3  
  }  
},
```

アプリではtextarea(html)として表示されます。



## 選択項目

複数データ登録の有無  
custom/is\_divided

有

有

無

```
"is_devided": {
  "label": {
    "ja": "複数データ登録の有無",
    "en": "is_devided"
  },
  "type": "string",
  "enum": [
    "有",
    "無"
  ]
}
```

アプリではselect(html)として表示されます。

## number

実数を入力する項目に利用

送状状数値入力値2  
custom/invoice\_number2

1.52

```
"invoice_number1": {
  "label": {
    "ja": "送状状数値入力値1",
    "en": "invoice_number1"
  },
  "type": "number"
},
```

## integer

整数を入力する項目に利用

送状状数値入力値整数1  
custom/invoice\_integer1

```
"invoice_integer1": {
  "label": {
```

```

    "ja": "送状状数値入力値整数1",
    "en": "invoice_integer1"
  },
  "type": "integer"
},

```

整数以外を入力して登録をしようとすると警告が表示され登録が中断されます。  
整数欄に実数を入れた場合の警告表示

送状状数値入力値整数1  

custom/invoice\_integer1

### string(format=date)

年月日を入力する項目に利用  
typeとformat指定を組み合わせて定義します

測定日時 必須  JST

custom/measurement\_date

```

"measurement_date": {
  "label": {
    "ja": "測定日時",
    "en": "measurement date"
  },
  "type": "string",
  "format": "date"
},

```

なお、日付項目はアプリでは日付入力のカジエツツ(html)が適用されます。

### 試料情報項目の定義

- 送状(invoice)における試料情報は基本情報、一般項目、分類別項目に大別されます。
- 基本情報は試料名、試料管理者など基本的情報を保存するためのもので、invoice.schema.jsonで項目を列挙する必要がありません。required要素(配列)に"sample"を追加するだけで利用できるようになります
- 一般項目、分類別項目は、RDEに予め登録された用語(term)と用語分類(class)を指定することで入力枠を生成することができます
- 用語と用語分類は、テンプレート生成様式(エクセルファイル)で提供されます

**試料基本項目** 試料の基本項目はシステムで予め定められた項目が表示されます。

以下のようにrequiredでsampleを指定、propertiesにsample{}を入力しておくとしシステム既定の項目が表示されます。

```
"$schema": "https://json-schema.org/draft/2020-12/schema",
"$id": "https://rde.nims.go.jp/rde/dataset-templates/
dataset_template__material_minimum_sample/invoice.schema.json",
"description": "RDEデータセットテンプレートサンプル試料最小構成invoice",
"type": "object",
"required": [
  "sample"
],
"properties": {
  "sample": {
    "type": "object",
    "label": {
      "ja": "試料情報",
      "en": "Sample Information"
    },
    "properties": {}
  }
}
```

試料情報

試料ID (新規登録)

試料選択

試料名(ローカルID) 必須

sample/names

試料名またはローカルIDを入力してください。

試料名を追加

化学式・組成式・分子式など

sample/composition

化学式・組成式・分子式などを自由入力してください。

試料管理者(所属) 必須

試料管理者(所属)を選択してください。

☐ データセット共有時に試料管理者 (所属)を匿名化する

参考URL

sample/referenceUrl

参考URLを入力してください。

関連試料

関連試料を追加

タグ

タグを入力してください。

試料の説明

sample/description

**B I H** | “ ≡ ≡ | 🔗 | 🗑

試料の説明、履歴など自由入力してください。

### 一般項目の定義

一般項目を定義する場合は以下のようにgeneralAttributesのitemsに定義します。割り当てる項目をtermIdで指定します。

```
"properties": {
  "generalAttributes": {
    "type": "array",
    "items": [
      {
        "type": "object",
        "required": [
          "termId"
        ],
        "properties": {
          "termId": {
            "const": "3adf9874-7bcb-e5f8-99cb-3d6fd9d7b55e"
          }
        }
      }
    ]
  },
}
```

上記のように一般項目(generalAttributes)を定義すると以下のように画面に入力項目が表示されます。 - 名称は、指定したtermIdで登録されたマスタデータが取得された名称が出力されます - また、名称はブラウザの言語設定に合わせて日本語/英語が自動的に表示されます - 入力はすべてstring扱いです - これらの項目をtaxonomyとして利用する場合のキー名は一般項目一覧を参照してください

一般項目	一般名称 sample/generalAttributes/0/value	<input type="text" value="一般名称を入力してください"/>
	化学組成 sample/generalAttributes/1/value	<input type="text"/>
	試料分類 sample/generalAttributes/2/value	<input type="text" value="試料分類を入力してください"/>
	CAS番号 sample/generalAttributes/3/value	<input type="text" value="CAS番号を入力してください"/>

**分類別項目の定義** 分類別項目を定義する場合は以下のようにspecificAttributesのitemsに定義します。割り当てる項目を分類のclassIdとtermIdの組み合わせで指定します。

```
"properties": {
  "specificAttributes": {
    "type": "array",
    "items": [
      {
        "type": "object",
        "required": [
```

```
      "classId",
      "termId"
    ],
    "properties": {
      "classId": {
        "const": "52148afb-6759-23e8-c8b8-33912ec5bfcf"
      },
      "termId": {
        "const": "70c2c751-5404-19b7-4a5e-981e6cebbb15"
      }
    }
  },
},
```

上記のように分類別項目(specificAttributes)を定義すると以下のように画面に入力項目が表示されます。 - 名称は、指定したclassIdとtermIdで登録されたマスタデータが取得された名称が出力されます - また、名称はブラウザの言語設定に合わせて日本語/英語が自動的に表示されます - 入力はすべてstring扱いです - これらの項目をtaxonomyとして利用する場合のキー名は一般項目一覧を参照してください

分類別項目

半導体

名称

sample/specificAttributes/0/value

セラミッ

名称

sample/specificAttributes/1/value

有機材料

分子量

sample/specificAttributes/2/value

## 7.1.2 invoice.schema.jsonの事例

以下に、invoice.schema.jsonの事例を紹介します。送状(invoice)で取り扱う基本情報、試料情報、固有情報の組み合わせを変えた事例となっています。

### 送状基本情報のみ

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://rde.nims.go.jp/rde/dataset-templates/dataset_template__minimum_sample/invoice.schema.json",
  "description": "RDEデータセットテンプレートサンプル最小構成invoice",
  "type": "object",
  "required": [],
  "properties": {}
}
```

送り状入力 : I

データセットタイプ  
対象装置  
課題番号/課題名

その他  
NIMS\_サンプル\_実験装置  
NIMSPJ20231220B001/NIMSPJ20231220B001

更新日時  
2023-12-20 13:17:46 JST

基本情報

記入年月日  
2024-03-14 JST

装置  
装置を選択してください。

データ投入者(所属)  
TOSAKA,Hiroaki (NIMS)

データ所有者(所属)  
TOSAKA,Hiroaki (NIMS)

データ名  
データ名を入力してください。

実験ID  
実験IDを入力してください。

説明  
説明を入力してください。

登録ファイル

登録ファイルをドラッグアンドドロップしてください。

ファイル選択

添付ファイル

添付ファイルをドラッグアンドドロップしてください。

ファイル選択

内容確認

キャンセル

表示例

## 送状基本情報+試料基本情報

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://rde.nims.go.jp/rde/dataset-templates/dataset_template__material_minimum_sample/invoice.schema.json",
  "description": "RDEデータセットテンプレートサンプル試料最小構成invoice",
  "type": "object",
  "required": [
    "sample"
  ],
  "properties": {
    "sample": {
      "type": "object",
      "label": {
        "ja": "試料情報",
        "en": "Sample Information"
      },
      "properties": {}
    }
  }
}
```

送り状入力：

データセットタイプ  
対象装置  
課題番号/課題名

その他  
NIMS\_サンプル\_実験装置  
NIMSPJ20231220B001/NIMSPJ20231220B001

更新日時  
2023-12-20 13:17:46 JST

基本情報

記入年月日  
2024-03-14 JST

装置  
装置を選択してください。

データ投入者(所属)  
TOSAKA,Hiroaki (NIMS)

データ所有者(所属)  
登坂ph3弘明 (合同会社フェーズ3)

データ名  
データ名を入力してください。

実験ID  
実験IDを入力してください。

説明  
説明を入力してください。

試料情報

試料ID  
(新規登録) 試料選択

試料名(ローカルID)  
試料名またはローカルIDを入力してください。

試料名を追加

化学式・組成式・分子式など  
化学式・組成式・分子式などを自由入力してください。

試料管理者(所属)  
試料管理者(所属)を選択してください。

☐ データセット共有時に試料管理者(所属)を匿名化する

参考URL  
参考URLを入力してください。

関連試料  
関連試料を追加

タグ  
タグを入力してください。

試料の説明  
試料の説明、履歴など自由入力してください。

\*\*表示例\*\*

## 送状基本情報+試料基本情報+試料追加情報



```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://rde.nims.go.jp/rde/dataset-templates/dataset_template_material_option_sample/invoice.schema.json",
  "description": "RDEデータセットテンプレートサンプル試料オプションありinvoice",
  "type": "object",
  "required": [
    "sample"
  ],
  "properties": {
    "sample": {
      "type": "object",
      "label": {
        "ja": "試料情報",
        "en": "Sample Information"
      },
      "properties": {
        "generalAttributes": {
          "type": "array",
          "items": [
            {
              "type": "object",
              "required": [
                "termId"
              ],
              "properties": {
                "termId": {
                  "const": "3adf9874-7bcb-e5f8-99cb-3d6fd9d7b55e"
                }
              }
            },
            {
              "type": "object",
              "required": [
                "termId"
              ],
              "properties": {
                "termId": {
                  "const": "0aadfff2-37de-411f-883a-38b62b2abbce"
                }
              }
            },
            {
              "type": "object",
              "required": [
                "termId"
              ],
              "properties": {
                "termId": {
                  "const": "0444cf53-db47-b208-7b5f-54429291a140"
                }
              }
            },
            {
              "type": "object",
              "required": [
                "termId"
              ],
              "properties": {
                "termId": {
                  "const": "e2d20d02-2e38-2cd3-b1b3-66fdb8a11057"
                }
              }
            }
          ]
        },
        "specificAttributes": {
          "type": "array",
          "items": [
            {
              "type": "object",
              "required": [
                "classId",
                "termId"
              ],
              "properties": {
                "classId": {
                  "const": "52148afb-6759-23e8-c8b8-33912ec5bfcf"
                },
                "termId": {
                  "const": "70c2c751-5404-19b7-4a5e-981e6cebbb15"
                }
              }
            },
            {
              "type": "object",
              "required": [
                "classId",
                "termId"
              ],
              "properties": {
                "classId": {
                  "const": "961c9637-9b83-0e9d-e60e-ffc1e2517afd"
                },
                "termId": {
                  "const": "70c2c751-5404-19b7-4a5e-981e6cebbb15"
                }
              }
            }
          ]
        }
      }
    },
    {
      "type": "object",
      "required": [
        "classId",
        "termId"
      ],
      "properties": {
        "classId": {
          "const": "961c9637-9b83-0e9d-e60e-ffc1e2517afd"
        },
        "termId": {
          "const": "70c2c751-5404-19b7-4a5e-981e6cebbb15"
        }
      }
    }
  ],
  {
    "type": "object",

```

```
    "required": [
      "classId",
      "termId"
    ],
    "properties": {
      "classId": {
        "const": "01cb3c01-37a4-5a43-d8ca-f523ca99a75b"
      },
      "termId": {
        "const": "dc27a956-263e-f920-e574-5beec912a247"
      }
    }
  }
}
}
```

送付状況:

データセットタイプ	その他	更新日時	2023-12-20 13:17:46 JST
対象装置	NIMS_サンプル_実験装置		
課題番号/課題名	NIMSPJ20231220B001/NIMSPJ20231220B001		

基本情報

記入年月日

2024-03-14 JST

装置

装置を選択してください。

データ投入者(所属)

TOSAKA,Hiroaki (NIMS)

データ所有者(所属)

豊坂ph3弘明 (合同会社フーズ3)

データ名

データ名を入力してください。

実験ID

実験IDを入力してください。

説明

説明を入力してください。

試料情報

試料ID

(新規登録)

試料選択

試料名(ローカルID)

試料名またはローカルIDを入力してください。

試料名を追加

化学式・組成式・分子式など

化学式・組成式・分子式などを自由に入力してください。

試料管理者(所属)

試料管理者(所属)を選択してください。

☐ データセット共有時に試料管理者(所属)を匿名化する

参考URL

参考URLを入力してください。

関連試料

関連試料を追加

タグ

タグを入力してください。

試料の説明

B I H “ ” := | = 🔗 🗑

試料の説明、履歴など自由に入力してください。

一般項目

一般名称

一般名称を入力してください

化学組成

試料分類

試料分類を入力してください

CAS番号

CAS番号を入力してください

分類別項目

半導体

名称

名称を入力してください

セラミツ

名称

名称を入力してください

有機材料

分子量

分子量を入力してください

\*\*表示例\*\*

## 送状基本情報+固有情報

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://rde.nims.go.jp/rde/dataset-templates/dataset_template_custom_sample/invoice.schema.json",
  "description": "RDEデータセットテンプレートサンプル固有情報invoice",
  "type": "object",
  "required": [
    "custom"
  ],
  "properties": {
    "custom": {
      "type": "object",
      "label": {
        "ja": "固有情報",
        "en": "Custom Information"
      },
      "required": [
        "measurement_date",
        "invoice_string1"
      ],
      "properties": {
        "measurement_date": {
          "label": {
            "ja": "測定日時",
            "en": "measurement date"
          },
          "type": "string",
          "format": "date"
        },
        "invoice_number1": {
          "label": {
            "ja": "送状状数値入力値1",
            "en": "invoice_number1"
          },
          "type": "number"
        },
        "invoice_number2": {
          "label": {
            "ja": "送状状数値入力値2",
            "en": "invoice_number2"
          },
          "type": "number"
        },
        "invoice_string1": {
          "label": {
            "ja": "送状文字入力値1",
            "en": "invoice_string1"
          },
          "type": "string"
        },
        "invoice_string2": {
          "label": {
            "ja": "送状文字入力値2",
            "en": "invoice_string2"
          },
          "type": "string",
          "options": {
            "widget": "textarea",
            "rows": 3
          }
        },
        "invoice_list1": {
          "label": {
            "ja": "送状状選択値1",
            "en": "invoice_list1"
          },
          "type": "string",
          "enum": [
            "selectable1",
            "selectable2",
            "selectable3"
          ]
        }
      }
    }
  }
}

```

送り状入力：

データセットタイプ

その他

更新日時

2023-12-20 13:17:46 JST

対象装置

NIMS\_サンプル\_実験装置

課題番号/課題名

NIMSPJ20231220B001/NIMSPJ20231220B001

基本情報

記入年月日

2024-03-14 JST

装置

装置を選択してください。

データ投入者(所属)

TOSAKA,Hiroaki (NIMS)

データ所有者(所属)

豊坂ph3弘明 (合同会社フエーズ3)

データ名

データ名を入力してください。

実験ID

実験IDを入力してください。

説明

説明を入力してください。

固有情報

測定日時

YYYY-MM-DD

JST

送状数値入力値1

送状数値入力値2

送状文字入力値1

送状文字入力値2

送状選択値1

\*\*表示例\*\*

### 7.1.3 送状基本情報+試料基本情報+試料追加情報+固有情報

---

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://rde.nims.go.jp/rde/dataset-templates/dataset_template_material_custom_sample/invoice.schema.json",
  "description": "RDEデータセットテンプレートサンプル試料と固有情報invoice",
  "type": "object",
  "required": [
    "custom",
    "sample"
  ],
  "properties": {
    "custom": {
      "type": "object",
      "label": {
        "ja": "固有情報",
        "en": "Custom Information"
      },
      "required": [
        "measurement_date",
        "invoice_string1"
      ],
      "properties": {
        "measurement_date": {
          "label": {
            "ja": "測定日時",
            "en": "measurement date"
          },
          "type": "string"
        },
        "invoice_number1": {
          "label": {
            "ja": "送状状数値入力値1",
            "en": "invoice_number1"
          },
          "type": "number"
        },
        "invoice_number2": {
          "label": {
            "ja": "送状状数値入力値2",
            "en": "invoice_number2"
          },
          "type": "number"
        },
        "invoice_string1": {
          "label": {
            "ja": "送状文字入力値1",
            "en": "invoice_string1"
          },
          "type": "string"
        },
        "invoice_string2": {
          "label": {
            "ja": "送状文字入力値2",
            "en": "invoice_string2"
          },
          "type": "string",
          "options": {
            "widget": "textarea",
            "rows": 3
          }
        },
        "invoice_list1": {
          "label": {
            "ja": "送状状選択値1",
            "en": "invoice_list1"
          },
          "type": "string",
          "enum": [
            "selectable1",
            "selectable2",
            "selectable3"
          ]
        }
      }
    },
    "sample": {
      "type": "object",
      "label": {
        "ja": "試料情報",
        "en": "Sample Information"
      },
      "properties": {
        "generalAttributes": {
          "type": "array",
          "items": [
            {
              "type": "object",
              "required": [
                "termId"
              ],
              "properties": {
                "termId": {
                  "const": "3adf9874-7bcb-e5f8-99cb-3d6fd9d7b55e"
                }
              }
            },
            {
              "type": "object",
              "required": [
                "termId"
              ]
            }
          ]
        }
      }
    }
  }
}

```



```

        "properties": {
          "termId": {
            "const": "0aadfff2-37de-411f-883a-38b62b2abbce"
          }
        },
      },
      {
        "type": "object",
        "required": [
          "termId"
        ],
        "properties": {
          "termId": {
            "const": "0444cf53-db47-b208-7b5f-54429291a140"
          }
        }
      },
      {
        "type": "object",
        "required": [
          "termId"
        ],
        "properties": {
          "termId": {
            "const": "e2d20d02-2e38-2cd3-b1b3-66fdb8a11057"
          }
        }
      }
    ],
    "specificAttributes": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "required": [
            "classId",
            "termId"
          ],
          "properties": {
            "classId": {
              "const": "52148afb-6759-23e8-c8b8-33912ec5bfcf"
            },
            "termId": {
              "const": "70c2c751-5404-19b7-4a5e-981e6cebbb15"
            }
          }
        },
        {
          "type": "object",
          "required": [
            "classId",
            "termId"
          ],
          "properties": {
            "classId": {
              "const": "961c9637-9b83-0e9d-e60e-ffc1e2517afd"
            },
            "termId": {
              "const": "70c2c751-5404-19b7-4a5e-981e6cebbb15"
            }
          }
        },
        {
          "type": "object",
          "required": [
            "classId",
            "termId"
          ],
          "properties": {
            "classId": {
              "const": "01cb3c01-37a4-5a43-d8ca-f523ca99a75b"
            },
            "termId": {
              "const": "dc27a956-263e-f920-e574-5beec912a247"
            }
          }
        }
      ]
    }
  }
}

```

[illegible]

一般名称	一般名称を入力してください
化学組成	
試験法	試験方法を入力してください
CAS番号	CAS番号を入力してください
分類別検索	
半導体	半導体を入力してください
セラミ	セラミを入力してください
有機材料	有機材料を入力してください
分子鎖	分子鎖を入力してください

**\*\*表示例\*\***

## 7.1.4 invoice.schema.jsonに記載されない定義について

invoice.schema.jsonはデータ受入画面を生成するための定義書かつ基本情報、試料、固有情報のメタデータを定義するためのJSONスキーマでありデータセットテンプレートとなっています。

ただし、基本情報、試料情報の一部はシステムが自動生成するためinvoice.schema.jsonに定義を記載する必要がありません。ここでは基本情報などinvoice.schema.jsonでは記載を省略できる項目に対するJSONスキーマを紹介します。

## 7.1.5 送状基本情報、試料のJSONスキーマ(invoice.schema.json)のjsonスキーマ

invoiceのbasic、sample(basic)は、invoice.schema.jsonでは定義されていないためバリデーションを行うことができません。invoice.schema.jsonのJSONスキーマの事例を紹介します。

### invoice\_basic\_and\_sample.schema.json

```
{
  "$schema": "https://json-schema.org/draft-07/schema",
  "description": "invoice_basic_schema/when adding samples",
  "type": "object",
  "required": [
    "basic",
    "datasetId"
  ],
  "properties": {
    "datasetId": {
      "type": "string"
    },
    "basic": {
      "type": "object",
      "label": {
        "ja": "送状基本情報",
        "en": "Invoice Basic Information"
      },
      "required": [
        "dateSubmitted",
        "dataOwnerId",
        "dataName"
      ],
      "properties": {
        "dateSubmitted": {
          "type": "string",
          "format": "date"
        },
        "dataOwnerId": {
```

```

        "type": "string",
        "pattern": "^[0-9a-zA-Z]{56})$"
    },
    "dateName": {
        "type": "string",
        "pattern": "^.*"
    },
    "instrumentId": {
        "type": [
            "string",
            "null"
        ],
        "pattern": "^[^$|^([0-9a-zA-Z]{8}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{12})$"
    },
    "experimentId": {
        "type": [
            "string",
            "null"
        ]
    },
    "description": {
        "type": [
            "string",
            "null"
        ]
    }
},
"sample": {
    "anyOf": [
        {
            "$ref": "#/definitions/sample/sampleWhenAdding"
        },
        {
            "$ref": "#/definitions/sample/sampleWhenRef"
        }
    ]
},
"definitions": {
    "sample": {
        "sampleWhenAdding": {
            "type": "object",
            "required": [
                "sampleId",
                "names",
                "ownerId"
            ],
            "properties": {
                "sampleId": {
                    "type": "string",
                    "pattern": "^[^$"
                },
                "names": {
                    "type": "array"
                },
                "ownerId": {
                    "description": "sample ownere id",
                    "type": "string",
                    "pattern": "^[0-9a-zA-Z]{56})$"
                }
            }
        }
    }
}

```

```

    },
    "composition": {
      "type": [
        "string",
        "null"
      ]
    },
    "referenceUrl": {
      "type": [
        "string",
        "null"
      ]
    },
    "description": {
      "type": [
        "string",
        "null"
      ]
    },
    "generalAttributes": {
      "type": "array",
      "properties": {
        "termId": {
          "type": [
            "string",
            "null"
          ],
          "pattern": "^[0-9a-f]{8}-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{12})$"
        },
        "value": {
          "type": [
            "string",
            "null"
          ]
        }
      }
    },
    "specificAttributes": {
      "type": "array",
      "properties": {
        "classId": {
          "type": [
            "string",
            "null"
          ],
          "pattern": "^[0-9a-f]{8}-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{12})$"
        },
        "termId": {
          "type": [
            "string",
            "null"
          ],
          "pattern": "^[0-9a-f]{8}-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{12})$"
        },
        "value": {
          "type": [
            "string",
            "null"
          ]
        }
      }
    }
  }
}

```

```

    ]
  }
}
}
},
"sampleWhenRef": {
  "type": "object",
  "required": [
    "sampleId"
  ],
  "properties": {
    "sampleId": {
      "type": "string",
      "pattern": "^[0-9a-f]{8}-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{12})$"
    },
    "names": {
      "type": "array"
    },
    "ownerId": {
      "description": "sample ownere id",
      "type": "string",
      "pattern": "^[0-9a-zA-Z]{56})$"
    },
    "composition": {
      "type": [
        "string",
        "null"
      ]
    },
    "referenceUrl": {
      "type": [
        "string",
        "null"
      ]
    },
    "description": {
      "type": [
        "string",
        "null"
      ]
    },
    "generalAttributes": {
      "type": "array",
      "properties": {
        "termId": {
          "type": [
            "string",
            "null"
          ],
          "pattern": "^[0-9a-f]{8}-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{12})$"
        },
        "value": {
          "type": [
            "string",
            "null"
          ]
        }
      }
    }
  }
}

```

```

    },
    "specificAttributes": {
      "type": "array",
      "properties": {
        "classId": {
          "type": [
            "string",
            "null"
          ],
          "pattern": "^[0-9a-f]{8}-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})$"
        },
        "termId": {
          "type": [
            "string",
            "null"
          ],
          "pattern": "^[0-9a-f]{8}-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})$"
        },
        "value": {
          "type": [
            "string",
            "null"
          ]
        }
      }
    }
  }
}

```

## 7.2 メタデータ定義[metadata-def.json]

### 7.2.1 メタデータ定義(metadata-def.json)について

- メタデータ定義はメタデータファイルの生成などに利用されます
- データセット詳細、データ詳細のメタ項目一覧における項目名、単位の出力に用いられます
- また、メタデータ定義は抽出されたメタデータの内容を説明するファイルとして利用できるため、メタデータ定義は組み合わせて使うものと位置付けています
- メタデータ定義は他のテンプレートとは異なりJSONスキーマ形式とはなっていないため、JSONのバリデーションには利用できません。

メタデータ定義の基本構成は以下の通りです。 - item\_nameの箇所に装置出力データから取得した項目名を割り当てます - name/ja、name/enは画面表示の項目名を指定します - scehamaには、メタ項目のデータ型を指定します。invoicec.schema.jsonにおけるtype、formatとほぼ同じ内容です - variableは、その項目が複数個出力される場合に定義します。単一出力の場合は定義する必要はありません。例えば、複数系列のメタデータ項目がある場合に利用します。 - orderは表示順を指定します。画面における一覧時の並び順となります。必須ではありません。

```
{
  "item_name": {
    "name": {
      "ja": "アイテム名",
      "en": "item name"
    },
    "schema": {
      "type": "string"
    },
    "variable": 1,
    "order": 1,
  }
}
```

### 7.2.2 constantとvariable

- メタデータ定義でvariable=1指定されたものはメタデータではvariable要素に配列として出力されます



- variableの指定がないものは、メタデータではconstant要素の子要素として出力されます
- 以下にその事例を示します

### メタデータ定義の事例

```
{
  "measurement_date": {
    "name": {
      "ja": "測定日",
      "en": "measurement date"
    },
    "schema": {
      "type": "string",
      "format": "date"
    },
    "order": 1,
  },
  "series_name": {
    "name": {
      "ja": "系列名",
      "en": "series name"
    },
    "schema": {
      "type": "string"
    },
    "order": 2,
    "variable": 1
  }
}
```

### 上記メタデータ定義に基づき出力されたメタデータの実例

```
{
  "constant": {
    "measurement_date": {
      "value": "2024-08-05"
    }
  },
  "variable": [
    {
      "series_name": {
        "value": "series1"
      }
    },
    {
      "series_name": {
        "value": "series2"
      }
    }
  ]
}
```

### 表示例

この例では系列名がvariable指定された項目であり、列項目名に値1、値2と表示した列に値が出力されています。

メタ情報

分類	日本語名	英語名	単位	値1	値2
固有情報	測定日	measurement date		2024-08-05	
	系列名	series name		series1	series2

### 7.2.3 データ型:arrayの取扱い

データ型はinvoice.schema.jsonとほぼ同じですが、メタデータ定義にはarray型があります。ただし、これの利用には以下の制約があるため、使用に際しては注意をしてください。

- array型のメタデータは登録できますが、RDEアプリの表示上は配列全体が一つの文字列として表示されます
- ただし、metadata.jsonファイルはそのまま保存され、ダウンロード時も保存時の状態で取り出すことができます

```
"meta_array":{
  "name":{
    "ja": "配列値",
    "en": "array_value"
  },
  "schema": "array"
}
```

```
{
  "constant": {
    "meta_array": {
      "value": [
        1, 2, 3, 4, 5, 6, 7, 8
      ]
    }
  }
}
```

	測定日時	measurement date		2024-03-01T00:00:00
	独立変数(X軸ラベル)	x-label		x(unit)
	従属変数(Y軸ラベル)	y-label		y(unit)
	系列数	series number	PCS	3
	送状文字入力値1	invoice_string1		1
	配列値	array_value		[1, 2, 3, 4, 5, 6, 7, 8]
	系列名	series name		series1series2
	系列ごとデータ数	data count by series	PCS	1511
	系列ごと平均値	data mean by series		102.0788.45

## 7.3 データセットカタログテンプレート [catalog.schema.json]

- データセットのカタログの入力様式はcatalog.schema.jsonで定義します
- データセット詳細のデータカタログはcatalog.schema.jsonを元に表示、入力画面が生成されます
- データセット詳細のデータカタログで「編集」することでcatalog.jsonが生成されます
- なお、未作成の場合は、ダウンロードに含まれるcatalog.jsonの内容は以下の通りです

データセットカタログが未編集の場合のcatalog.jsonの内容

```
null
```

catalog.schema.jsonの一例を掲載します。

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://rde.nims.go.jp/rde/dataset-templates/NIMS_TRIAL_DATASETTEMPLATE-1/",
  "type": "object",
  "required": [
    "catalog"
  ],
  "description": "NIMS_TRIAL_DATASETTEMPLATE-1",
  "properties": {
    "catalog": {
      "type": "object",
      "label": {
        "ja": "RDEトライアルデータセットテンプレート-1",
        "en": "RDE trial datasettemplate-1"
      },
      "required": []
    }
  }
}
```

```

"properties": {
  "dataset_title": {
    "label": {
      "ja": "データセット名",
      "en": "Dataset Title"
    },
    "type": "string"
  },
  "abstract": {
    "label": {
      "ja": "概要",
      "en": "Abstract"
    },
    "type": "string"
  },
  "data_creator": {
    "label": {
      "ja": "作成者",
      "en": "Data Creator"
    },
    "type": "string"
  },
  "experimental_apparatus": {
    "label": {
      "ja": "使用装置",
      "en": "Experimental Apparatus"
    },
    "type": "string"
  },
  "data_distribution": {
    "label": {
      "ja": "データの再配布",
      "en": "Data Distribution"
    },
    "type": "string"
  },
  "raw_data_type": {
    "label": {
      "ja": "データの種類",
      "en": "Raw Data Type"
    },
    "type": "string"
  },
  "stored_data": {
    "label": {
      "ja": "格納データ",
      "en": "Stored Data"
    },
    "type": "string"
  },
  "remarks": {
    "label": {
      "ja": "備考",
      "en": "Remarks"
    },
    "type": "string"
  },
  "references": {
    "label": {
      "ja": "参考論文",
      "en": "References"
    }
  }
}

```

```

    },
    "type": "string"
  }
}
}
}
}

```

## 7.4 バッチ定義[jobs.template.yaml,batch.yaml]

下記の定義ファイルは実例を示したものです。

この定義ファイルには、プログラムなど実行環境をまとめたDocker image、実行するプログラムなどが記載されています。なお、これらのファイルはNIMSが提供します。

### 7.4.1 jobs.template.yaml

利用するDocker image(docker\_image)の指定、実行するコマンド(command)を指定します。その他の設定は変更はしません。なお、RDEの構造化処理のコマンドは  
 command: python /app/main.py  
 が標準となります。

```

job_specifications:
- id: ${job-id}
  allow_run_on_missing_image: true
  auto_complete: true
  max_task_retries: 0
  tasks:
  - id: task_1
    docker_image: rdecontreg.azurecr.io/nims_trial_datasettemplate_1:v.1.6
    command: python /app/main.py
    input_data:
      azure_storage:
        - storage_account_settings: mystorageaccount
          remote_path: ${job-dir}
          local_path: ${AZ_BATCH_TASK_WORKING_DIR}/data
    output_data:
      azure_storage:
        - storage_account_settings: mystorageaccount
          remote_path: ${job-dir}
          local_path: ${AZ_BATCH_TASK_WORKING_DIR}/data
      condition: taskcompletion

```

## 7.4.2 batch.yaml

---

このファイルはNIMSが提供します。  
構造化処理を実行する環境を指定します。

```
pool: poollinux
```

## 7.5 tasksupport

---

tasksupport(タスクサポート)フォルダは、構造化処理などで利用する設定ファイルなどをテンプレートに予め配置しておくための保存場所となります。例えば、RDEToolKitではメタデータ定義をタスクサポートに配置されたmetadata-def.jsonから読み込むことを想定して作られています。タスクサポートに配置されたファイルは、構造化処理においてtasksupportフォルダにコピーされます。

## 7.6 dockerコンテナイメージ

---

RDEの構造化処理のプログラムはDockerコンテナで実行されます。

データセットテンプレートの登録時には予めbuildされたDocker imageをRDEのレポジトリに登録しておきます。

Dockerコンテナを利用することで、プログラムおよび実行環境、ライブラリ、その他プログラムを実行環境を含めてまとめておくことができます。

Dockerコンテナの登録はNIMSが行います。開発者はローカル環境でテストをした後にDockerfileおよびbuildに必要なもの一式を送付していただくことになります。

## 8. データセットテンプレートのバリデーション

### 8.1 RDEToolKitにおけるJSONファイルのバリデーション

RDEToolKitはinvoice.jsonファイルなどのバリデーションを行い書式や値の不整合を確認します。invoice.jsonはinvoice.schema.jsonを用いてバリデーションできますが、すべてのinvoice項目を網羅していません。例えば、invoice.jsonで指定するdatasetId、basic.dataOwnerId、basic.dateSubmittedなど必須項目がありますが、これらはinvoice.schema.jsonではバリデーションできません。そのため、RDEToolKitでは、ユーザー定義のinvoice項目以外のシステムが生成する項目について網羅したJSONスキーマファイルを用いてバリデーションを行っています。このバリデーションで不具合があるとデータの永続登録時に不具合が発生し登録処理が異常終了となるため、事前の確認が欠かせません。

構造化処理プログラム開発においてはinvoice.jsonファイルは自作する必要があります(テンプレート生成ツールを利用すれば作成できます)、テストにおいてJSONスキーマに則ったjsonファイルを作成することが必要です。そのための情報を下記にまとめました。

#### 8.1.1 invoice.jsonのbasic、sampleなどの定義

以下はinvoice.jsonのbasicとsampleの出力事例です。

datasetId、dataOwnerId(などのユーザID)、試料のclassId、termIdなどはUUIDやランダム文字列と定義されています。それぞれ長さが異なります。試料のclassId、termIdはテンプレート様式ファイルで提供される値を利用すれば良いのですが、datasetId、sampleIdなどはシステムが生成するものであり、またdataOwnerIdなどのuserIdについては本人以外知り得ない情報となり自作する必要があります。

```
{
  "datasetId" : "b84a220a-0be1-40f5-b46b-bde08a318523",
  "basic" : {
    "dateSubmitted" : "2024-03-13",
    "dataOwnerId" : "de17c7b3f0ff5126831c2d519f481055ba466ddb6238666132316439",
    "dataName" : "invoiceSample_TSK20240313-1",
    "instrumentId" : null,
    "experimentId" : null,
    "description" : ""
  },
  "sample" : {
    "sampleId" : "2ddf932e-fa80-40d7-a010-939ab486e309",
  }
}
```

```
"names" : [ "sample_name" ],
"composition" : null,
"referenceUrl" : null,
"description" : null,
"generalAttributes" : [ {
  "termId" : "3adf9874-7bcb-e5f8-99cb-3d6fd9d7b55e",
  "value" : null
},
"specificAttributes" : [ {
  "classId" : "52148afb-6759-23e8-c8b8-33912ec5bfcf",
  "termId" : "70c2c751-5404-19b7-4a5e-981e6cebbb15",
  "value" : null
}],
"ownerId" : "de17c7b3f0ff5126831c2d519f481055ba466ddb6238666132316xxx"
}
```

invoice.jsonにおけるidの自作についてはそれぞれ以下の表のように作成してください。

項目	生成条件	必須
datasetId	UUID形式	必須
basic.dateSubmitted	日付、YYYY-MM-DD	必須
basic.dataOwnerId	ランダム文字列、56文字	必須
basic.dataName	文字列、256文字	必須
basic.instrumentId	UUID形式	任意
basic.experimentId	任意文字列、256文字	任意
basic.description	任意文字列、8192文字	任意
sample.sampleId	UUID形式	利用時必須
sampleのtermId	UUID形式	利用時必須
sampleのclassId	UUID形式	利用時必須

なお、invoice.jsonには含まれてはいない項目ですがdataのIDについては以下の通り

項目	生成条件	必須
dataId	UUID形式	利用時必須

## 8.1.2 RDEToolKitでinvoice.jsonのバリデーションに用いられるJSONスキーマファイル

以下はRDEToolKitでinvoice.jsonのバリデーションに用いられるJSONスキーマファイル  
invoice\_basic\_and\_sample.schema.json



の全出力です。invoice.schema.jsonで開発者が定義できるもの以外は下記のようにスキーマ定義されています。

```
{
  "$schema": "https://json-schema.org/draft-07/schema",
  "description": "invoice_basic_schema/when adding samples",
  "type": "object",
  "required": [
    "basic",
    "datasetId"
  ],
  "properties": {
    "datasetId": {
      "type": "string"
    },
    "basic": {
      "type": "object",
      "label": {
        "ja": "送状基本情報",
        "en": "Invoice Basic Information"
      },
      "required": [
        "dateSubmitted",
        "dataOwnerId",
        "dataName"
      ],
      "properties": {
        "dateSubmitted": {
          "type": "string",
          "format": "date"
        },
        "dataOwnerId": {
          "type": "string",
          "pattern": "^[0-9a-zA-Z]{56})$"
        },
        "dateName": {
          "type": "string",
          "pattern": "^.*)"
        },
        "instrumentId": {
          "type": [
            "string",
            "null"
          ],
          "pattern": "^[^$|^([0-9a-zA-Z]{8})-[0-9a-zA-Z]{4})-[0-9a-zA-Z]{4})-[0-9a-zA-Z]{4})-[0-9a-zA-Z]{12})$)"
        },
        "experimentId": {
          "type": [
            "string",
            "null"
          ]
        },
        "description": {
          "type": [
            "string",
            "null"
          ]
        }
      }
    }
  }
}
```

```

    }
  },
  "sample": {
    "anyOf": [
      {
        "$ref": "#/definitions/sample/sampleWhenAdding"
      },
      {
        "$ref": "#/definitions/sample/sampleWhenRef"
      },
      {
        "$ref": "#/definitions/sample/sampleWhenAddingExcelInvoice"
      }
    ]
  }
},
"definitions": {
  "sample": {
    "sampleWhenAdding": {
      "type": "object",
      "required": [
        "sampleId",
        "names",
        "ownerId"
      ],
      "properties": {
        "sampleId": {
          "type": "string",
          "pattern": "^$"
        },
        "names": {
          "type": "array"
        },
        "ownerId": {
          "description": "sample ownere id",
          "type": "string",
          "pattern": "^[0-9a-zA-Z]{56})$"
        },
        "composition": {
          "type": [
            "string",
            "null"
          ]
        },
        "referenceUrl": {
          "type": [
            "string",
            "null"
          ]
        },
        "description": {
          "type": [
            "string",
            "null"
          ]
        },
        "generalAttributes": {
          "type": "array",
          "properties": {
            "termId": {
              "type": [

```

```

        "string",
        "null"
    ],
    "pattern": "^[0-9a-f]{8}-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{12})$"
},
{
    "value": {
        "type": [
            "string",
            "null"
        ]
    }
},
},
"specificAttributes": {
    "type": "array",
    "properties": {
        "classId": {
            "type": [
                "string",
                "null"
            ],
            "pattern": "^[0-9a-f]{8}-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{12})$"
        },
        "termId": {
            "type": [
                "string",
                "null"
            ],
            "pattern": "^[0-9a-f]{8}-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{12})$"
        },
        "value": {
            "type": [
                "string",
                "null"
            ]
        }
    }
},
},
"sampleWhenRef": {
    "type": "object",
    "required": [
        "sampleId"
    ],
    "properties": {
        "sampleId": {
            "type": "string",
            "pattern": "^[0-9a-f]{8}-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{12})$"
        },
        "names": {
            "type": "array"
        },
        "ownerId": {
            "description": "sample ownere id",
            "type": "string",
            "pattern": "^[0-9a-zA-Z]{56})$"
        }
    }
}

```

```

    },
    "composition": {
      "type": [
        "string",
        "null"
      ]
    },
    "referenceUrl": {
      "type": [
        "string",
        "null"
      ]
    },
    "description": {
      "type": [
        "string",
        "null"
      ]
    },
    "generalAttributes": {
      "type": "array",
      "properties": {
        "termId": {
          "type": [
            "string",
            "null"
          ],
          "pattern": "^[0-9a-f]{8}-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{12})$"
        },
        "value": {
          "type": [
            "string",
            "null"
          ]
        }
      }
    },
    "specificAttributes": {
      "type": "array",
      "properties": {
        "classId": {
          "type": [
            "string",
            "null"
          ],
          "pattern": "^[0-9a-f]{8}-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{12})$"
        },
        "termId": {
          "type": [
            "string",
            "null"
          ],
          "pattern": "^[0-9a-f]{8}-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{12})$"
        },
        "value": {
          "type": [
            "string",
            "null"
          ]
        }
      }
    }
  }
}

```

```

    ]
  }
}
}
},
"sampleWhenAddingExcelInvoice": {
  "type": "object",
  "required": [
    "names",
    "ownerId"
  ],
  "properties": {
    "sampleId": {
      "type": "string",
      "pattern": "^$"
    },
    "names": {
      "type": "array"
    },
    "ownerId": {
      "description": "sample ownere id",
      "type": "string",
      "pattern": "^[0-9a-zA-Z]{56})$"
    },
    "composition": {
      "type": [
        "string",
        "null"
      ]
    },
    "referenceUrl": {
      "type": [
        "string",
        "null"
      ]
    },
    "description": {
      "type": [
        "string",
        "null"
      ]
    },
    "generalAttributes": {
      "type": "array",
      "properties": {
        "termId": {
          "type": [
            "string",
            "null"
          ],
          "pattern": "^[0-9a-f]{8})-(0-9a-f){4})-(0-9a-f){4})-(0-9a-f){4})-(0-9a-f){12})$"
        },
        "value": {
          "type": [
            "string",
            "null"
          ]
        }
      }
    }
  }
}

```

```

    },
    "specificAttributes": {
      "type": "array",
      "properties": {
        "classId": {
          "type": [
            "string",
            "null"
          ],
          "pattern": "^[0-9a-f]{8}-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})$"
        },
        "termId": {
          "type": [
            "string",
            "null"
          ],
          "pattern": "^[0-9a-f]{8}-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})$"
        },
        "value": {
          "type": [
            "string",
            "null"
          ]
        }
      }
    }
  }
}

```

## 9. 付録ファイル

---

### 9.1 ファイルの提供

---

- 本資料で掲載したテンプレートファイルなど、下記のものを提供しています
- rde-datasettemplate-instructions\_appendix.zipファイルを取得、展開してご利用ください

#### 9.1.1 提供ファイル一式

---

##### **invoice.schema.jsonの事例で掲載したinvoice.schema.json**

invoice\_schema\_samplesフォルダ参照

- 送状基本情報のみ
  - dataset\_template\_minimumフォルダ
- 送状基本情報+試料基本情報
  - dataset\_template\_material\_minimumフォルダ
- 送状基本情報+試料基本情報+試料追加情報
  - dataset\_template\_material\_optionフォルダ
- 送状基本情報+固有情報
  - dataset\_template\_customフォルダ
- 送状基本情報+試料基本情報+試料追加情報+固有情報
  - dataset\_template\_material\_customフォルダ

各フォルダに保存されたエクセルファイルは、テンプレート作成ツールで利用可能な様式ファイルです。

##### **データセットカタログテンプレート、バッチ定義などで掲載したもの**

NIMS\_TRIAL\_DATASETTEMPLATE-1フォルダ参照

## データのダウンロード結果のサンプル

data\_download\_sampleフォルダ参照

## JSONスキーマ

- RDEToolKitのバリデーションで利用されているJSONスキーマファイルです。

json\_schemaフォルダ参照

# 10. 次のステップ

---

構造化処理プログラムの開発について知りたい方は  
RDEToolKit(invoiceモード)を利用したシンプルなRDEデータ構造化処理プログラムハンズオン.pdf  
をご覧ください。