

DX Suite 標準API V2仕様書

2025年02月14日
第2.0版

この仕様書はwith DX Suite / with AI inside 製品にも共通の内容です。

本書の内容については、無断転載・無断使用を固く禁じます。

改版履歴

版数	発行日	変更内容
第1.0版	2022/11/15	初版
第1.1版	2023/02/21	<ul style="list-style-type: none"> ● 各APIのリクエストパラメータに「必須」カラムを追加しました。 ● 以下のAPIを新規で追加しました。 <ul style="list-style-type: none"> ○ ワークフロー検索API ○ 読取パーツ情報取得API ○ 読取ページ画像取得API
第1.2版	2023/03/22	<ul style="list-style-type: none"> ● 読取パーツ情報取得APIのレスポンス値の「accuracy(読取尤度)」に対する補足説明を更新しました。
第1.3版	2023/04/18	<ul style="list-style-type: none"> ● 以下のAPIを新規で追加しました。 <ul style="list-style-type: none"> ○ 読取ユニット検索API
第1.4版	2023/05/23	<ul style="list-style-type: none"> ● 読取ユニット登録APIの以下パラメータを更新しました。 <ul style="list-style-type: none"> ○ filesのファイル指定可能数を変更 ○ unitNameの指定を任意に変更 ● 以下のAPIを新規で追加しました。 <ul style="list-style-type: none"> ○ 仕分けユニット追加API ○ 仕分け実施API ○ 仕分け結果OCR送信API ○ 仕分け結果CSVエクスポートAPI ○ 仕分けユニット検索API ○ 仕分けユニット削除API
第1.5版	2023/06/21	<ul style="list-style-type: none"> ● 各APIの「リクエストの例」を修正しました。
第1.6版	2023/09/19	<ul style="list-style-type: none"> ● 以下のAPIを新規で追加しました。 <ul style="list-style-type: none"> ○ 読取ユニット内の全パーツ画像取得API ● 以下のAPIのレスポンスに項目「パーツ画像作成ステータス」を追加しました。 ● 以下のAPIのレスポンス項目「dataProcessingStatus」に、新規ステータス(600)を追加しました。 <ul style="list-style-type: none"> ○ 読取ユニット検索API ○ 読取ユニット状態取得API

第1.7版	2023/11/21	<ul style="list-style-type: none"> ● 以下のAPIのレスポンス項目「statusCode」のOCR送信完了(90)を廃止し、コンペア完了(130)に統一しました。 <ul style="list-style-type: none"> ○ 仕分けユニット検索API
第1.8版	2024/01/16	<ul style="list-style-type: none"> ● 以下のAPIのレスポンス項目「partsImageCreationStatus」に新規ステータス(0: パーツ画像生成対象無し)を追加しました。 <ul style="list-style-type: none"> ○ 読取ユニット検索API ○ 読取ユニット状態取得API ● 以下のAPIに補足(・「項目抽出」帳票には対応していません。)を追加しました。 <ul style="list-style-type: none"> ○ ワークフロー検索API ○ ワークフロー設定取得API ○ 読取パーツ情報取得API ○ 読取ユニット内の全パーツ画像取得API ○ 読取ページ画像取得API ○ 読取ユニットID取得API ○ 読取ユニット状態取得API ● 以下の新規エラーコードを追加しました。 <ul style="list-style-type: none"> ○ エラーコード: 120015
第1.9版	2024/04/23	<ul style="list-style-type: none"> ● 以下のAPIについて、補足(・「項目抽出」帳票には対応していません。)を削除しました。 <ul style="list-style-type: none"> ○ 読取パーツ情報取得API ● 読取パーツ情報取得APIのレスポンス値の「x」、「y」、「width」、「height」、「accuracy(読取尤度)」に対する補足事項を更新しました。
第2.0版	2025/02/14	<ul style="list-style-type: none"> ● 読取パーツ情報取得APIのレスポンス項目「columnNo」から、「(0はCSV出力しない)」の記述を削除しました。

目次

1. 概要	5
(1) 標準API V2でできること	5
(2) リソースの識別子	5
(3) 帳票アップロード	5
2. 共通仕様	7
(1) インタフェース仕様	7
(2) エラーレスポンスの形式	7
3. API一覧	9
4. API仕様	10
(1) ワークフロー検索API	10
(2) ワークフロー設定取得API	12
(3) 読取ユニット登録API	14
(4) 読取ユニット検索API	16
(5) CSVダウンロードAPI	19
(6) 読取ユニット削除API	20
(7) 読取パーツ情報取得API	21
(8) 読取ユニット内の全パーツ画像取得API	24
(9) 読取ページ画像取得API	26
(10) 読取ユニットID取得API	28
(11) 読取ユニット状態取得API	30
(12) 仕分けユニット追加API	33
(13) 仕分け実施API	35
(14) 仕分け結果OCR送信API	37
(15) 仕分け結果CSVエクスポートAPI	39
(16) 仕分けユニット検索API	40
(17) 仕分けユニット削除API	42
5. 付録	44
(1) エラーコード・メッセージ一覧	44
(2) 各APIにおける入出力の関係	46
(3) Elastic Sorterの仕分けステータス	47

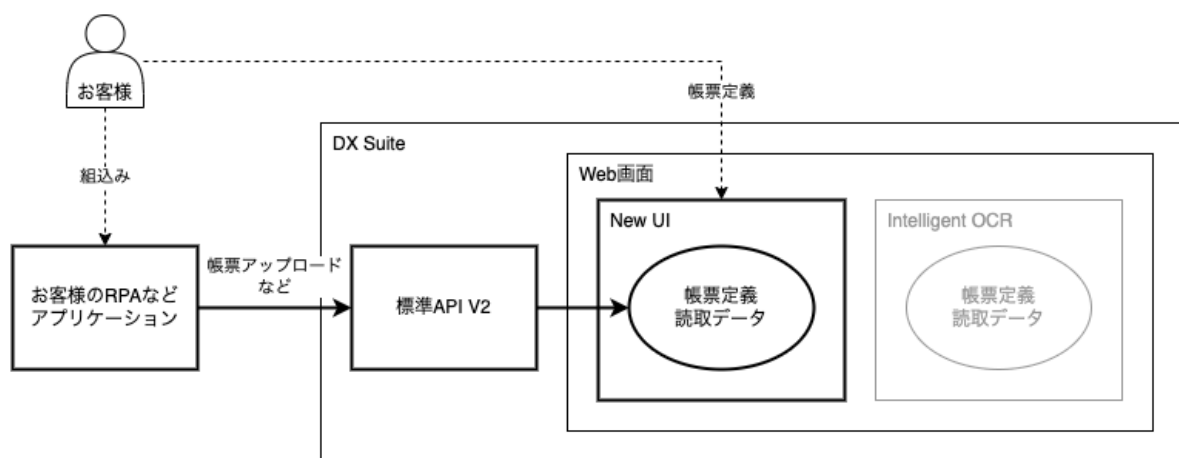
1. 概要

(1) 標準API V2でできること

DX Suite 標準API V2(以下、本API)は、Web画面で操作することなく、DX Suite(New UI)のデータを操作できるWebインターフェースです。お客様が構築するアプリケーションやRPAに本APIの呼出しを組み込むことで、業務処理を自動化できます。

本APIはDX Suite(New UI)のデータを操作できます。DX Suite(Intelligent OCR)とはデータを共有しませんのでご注意ください。

本APIは帳票の読取からCSV出力までの操作、帳票の仕分け機能を提供します。Web画面を利用してNew UI上に必要な帳票定義、仕分けルールを作成してください。本APIで帳票定義、仕分けルールを作成することはできません。



(2) リソースの識別子

本APIでDX Suite(New UI)のデータを操作するには、ワークフロー(帳票定義)や読取ユニット(読取データ)等リソースの識別子、仕分けルールや仕分けユニットの識別子をパラメータに指定します。各識別子(ID)はDX Suite(New UI)のWeb画面から取得できます。また、読取ユニットのIDや仕分けユニットのIDは本APIの返却データとしても取得できます。

読取ユニット登録時・仕分けユニット登録時等で、OCRリクエスト数集計の仕分け先として、リクエストボディに部署IDを指定できます。部署IDはDX Suite(Intelligent OCR)のWeb画面「設定メニュー／組織と権限の設定」にて管理する部署の識別子です。

(3) 帳票アップロード

1回の読取ユニット登録リクエストについて、複数の画像ファイルや複数のページを含むPDFをアップロードできます。

- 1リクエストあたりの推奨ページ数 (上限) : 3,000
- 1ファイルあたりの推奨ページ数 (上限) : 500
- 読取ユニット内の全パーツ画像取得APIを利用する場合は、1ユニットの合計パーツ数(1ページ内の項目数 x ページ数)が1,500以下となるように読取ユニット登録時に調整してください。

その他、入力画像の制限事項については別資料『DX Suite クラウドサービス仕様書』の「2(2) 入力画像仕様」を参照してください。

2. 共通仕様

(1) インタフェース仕様

本APIの共通仕様は以下です。

項目	説明
リクエストホスト	https://{組織固有}.dx-suite.com/wf/api/standard/v2/ URIは各API仕様のURIを参照ください。
プロトコル	HTTPS
レスポンスデータ形式	application/json または text/csv
文字コード	リクエスト・レスポンスともに UTF-8 です。
レスポンス ステータスコード	サーバで処理が行われた場合、200 で応答します。 処理前のエラーの場合、4xx/5xx で返ることもあります。
認証	呼出しには DX Suite の「設定」>「APIキー設定」で発行したAPI キー が必要です。 APIキーはお客様を特定し課金等に必要のため、第三者に知ら れないようご注意ください。 APIキーは、apikey ヘッダで付与します。

※Windows環境でcurlコマンドをご利用になる場合、ファイル名等にマルチバイト文字(全角文字等)が含まれていると文字コードの影響で正常に処理ができません。

(2) エラーレスポンスの形式

以下のレスポンス構造のJSON形式の文字列で返されます。

JSONオブジェクト		
	errors	エラーリスト エラーの情報を格納した配列
	errorCode	エラーコード
	message	エラーメッセージ

※エラーコードやメッセージについては「エラーコード・メッセージ一覧」を参照

エラーレスポンスの例: 存在しないIDを指定した場合

```
{  
  "errors": [  
    {  
      "errorCode": 120001,  
      "message": "specified resource does not exist"  
    }  
  ]  
}
```


3. API一覧

現在、提供している API は以下です。

	カテゴリ	処理	概要
1	Intelligent OCR	ワークフロー検索	登録済みのワークフロー情報を取得します。
2		ワークフロー設定取得	ワークフローの設定を取得します。
3		読取ユニット登録	ワークフローに読取ユニットを登録します。
4		読取ユニット検索	読取ユニット情報を取得します。
5		CSVダウンロード	読取ユニットに含まれる読取結果をCSVフォーマットで取得します。
6		読取ユニット削除	読取ユニットを削除します。
7		読取パーツ情報取得	読取ユニットの読取パーツ情報を取得します。
8		読取ユニット内の全パーツ画像取得API	読取ユニットのパーツ画像を全て取得します。
9		読取ページ画像取得	読取ユニットのページ画像を取得します。
10		読取ユニットID取得	ワークフローに登録されている読取ユニットのIDを取得します。 * 読取ユニット検索で代用できます
11		読取ユニット状態取得	読取ユニットの状態を取得します。 * 読取ユニット検索で代用できます
12	Elastic Sorter	仕分けユニット追加	仕分けユニットを登録します。
13		仕分け実施	自動仕分けを行います。
14		仕分け結果OCR送信	仕分け結果をOCRに送信します。
15		仕分け結果CSVエクスポート	仕分け結果をCSVフォーマットで取得します。
16		仕分けユニット検索	仕分けユニット情報を取得します。
17		仕分けユニット削除	仕分けユニットを削除します。

各APIのパラメータと返却データはリソースの識別子で関連しています。各APIとリソースとの関係については付録の「[各APIにおける入出力の関係](#)」を参照してください。

API利用参考記事: <https://help-dxsuite.inside.ai/hc/ja/articles/24264809201177>

4. API仕様

(1) ワークフロー検索API

ワークフロー情報を取得するAPI

URI	/workflows		
メソッド	GET (application/json)		
認証	apikey		
URL パラメータ (クエリ)	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	folderId		フォルダID (UUID形式※1) 指定したフォルダ内のワークフローを検索します。 フォルダIDを複数指定する場合、カンマ区切りで指定可能です。
	workflowName		ワークフロー名 指定した文字列に完全一致するワークフローを検索します。 1～128文字まで指定できます。 日本語で設定する場合は、URLエンコードして設定します。 例) テスト → %e3%83%86%e3%82%b9%e3%83%88
※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx 形式の36文字のID ・「項目抽出」帳票には対応していません。			

リクエストの例

```
GET /workflows?folderId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx&workflowName=My+workflow
HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxxxx
Host: <request-host>
```

curlコマンドの例

```
curl
"<request-host>/workflows?folderId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx&workflowName=My+workflow" -X GET -H "apikey:xxxxxxxxxxxxxxxxxxxxx"
```

レスポンスの形式

以下のレスポンス構造のJSON形式の文字列で返されます。

JSONオブジェクト		
	workflows	ワークフローの配列
	id	ワークフローID
	folderId	フォルダID
	name	ワークフロー名

レスポンスの例

```
{
  "workflows": [
    {
      "workflowId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
      "folderId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
      "name": "My workflow"
    },...
  ]
}
```

(2) ワークフロー設定取得API

ワークフローの設定を取得するAPI

URI	/workflows/{workflowId}/revisions/{revision}/configuration		
メソッド	GET (application/json)		
認証	apikey		
URL パラメータ (パス)	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	workflowId	○	ワークフローID (UUID形式※1)
	revision	○	ワークフローのリビジョン ワークフローのリビジョンは1以上の整数です。ワークフローのリビジョンは「 (11) 読取ユニット状態取得API 」にて取得できます。
※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx 形式の36文字のID ・「項目抽出」帳票には対応していません。			

リクエストの例

```
GET /workflows/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/revisions/x/configuration HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxx
Host: <request-host>
```

curlコマンドの例

```
curl "<request-host>/workflows/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/revisions/x/configuration"
-X GET -H "apikey:xxxxxxxxxxxxxxxxxxxx"
```

レスポンスの形式

以下のレスポンス構造のJSON形式の文字列で返されます。

JSONオブジェクト	
workflowId	ワークフローID
revision	ワークフローのリビジョン
applicationType	アプリケーション 1: DX Suite (固定)
ocrKindType	OCR種類 1: 定形

	2: 非定型 3: 全文OCR
atypicalModelName	非定型帳票モデル OCR種類が「非定型」の場合のみ返却されます。
dataCheck	データチェック true : ON false: OFF
dataProcessing	データ加工 true : ON false: OFF
outputcharCode	出力ファイル文字コード 1: Shift-JIS 2: UTF-8

レスポンスの例

```
{
  "workflowId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx",
  "revision": 1,
  "applicationType": 1,
  "ocrKindType": 2,
  "atypicalModelName": "xxxxxx",
  "dataCheck": false,
  "dataProcessing": false,
  "outputCharCode": "2"
}
```

(3) 読取ユニット登録API

ワークフローに読取ユニットを登録するAPI

URI	/workflows/{workflowId}/units		
メソッド	POST (application/json)		
認証	apikey		
リクエスト データ形式	multipart/form-data		
URL パラ メータ (パス)	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	workflowId	○	ワークフローID (UUID形式※1)
	※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx 形式の36文字のID ・全文読取には対応しておりません。		
リクエスト ボディ	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	files	○	追加する画像データファイル サポートしている拡張子は png、jpg、jpeg、pdf、tiff です。 ユニットに登録するファイルを1～1000件まで指定できます。ファイルを複数指定する場合、ファイルの数だけ -F "files=@{ファイル名}" を追加します。
	unitName		登録するユニット名 ユニット名を1～128文字で指定します。半角全角英数記号かなカナ漢字を使用できますが、\ / : * ? " < > は使用不可文字です。 省略時は登録した年月日時分秒で設定されます。
	departmentId		部署ID OCRリクエスト数集計の仕分け先を指定します。形式は数字です。



リクエストの例

```
POST /workflows/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx/units HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxx
Host: <request-host>
Content-Type: multipart/form-data; boundary=-----680b5c02be1293c4

-----680b5c02be1293c4
Content-Disposition: form-data; name="files"; filename="test.png"
Content-Type: image/png
-----680b5c02be1293c4
Content-Disposition: form-data; name="unitName"
test
```

curlコマンドの例

```
curl "<request-host>/workflows/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx/units" -X POST -H
"apikey:xxxxxxxxxxxxxxxxxxxx" -F "files=@test.png" -F "unitName=test"
```

レスポンスの形式

以下のレスポンス構造のJSON形式の文字列で返されます。

JSONオブジェクト

unitId	ユニットID
unitName	ユニット名

レスポンスの例

```
{
  "unitId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx",
  "unitName": "test"
}
```

(4) 読取ユニット検索API

読取ユニット情報を取得するAPI

URI	/units		
メソッド	GET (application/json)		
認証	apikey		
URL パラメータ (クエリ)	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	folderId	※2	フォルダID (UUID形式※1) 指定したフォルダ内のユニットを検索します。 フォルダIDを複数指定する場合、カンマ区切りで指定可能です。
	workflowId	※2	ワークフローID (UUID形式※1) 指定したワークフロー内のユニットを検索します。 ワークフローIDを複数指定する場合、カンマ区切りで指定可能です。
	unitId	※2	ユニットID (UUID形式※1) 指定したユニットIDのユニットを検索します。 ユニットIDを複数指定する場合、カンマ区切りで指定可能です。
	unitName		ユニット名 指定した文字列に完全一致するユニットを検索します。 1~128文字まで指定できます。 日本語で設定する場合は、URLエンコードして設定します。 例) テスト → %e3%83%86%e3%82%b9%e3%83%88
	status		ユニットのステータス 指定したユニットステータスのユニットを検索します。※3 ステータスを複数指定する場合、カンマ区切りで指定可能です。 2: データ化中 13: データチェック中 15: コンペア中 16: コンペア完了

		22: CSVダウンロード済
	createdFrom	ユニット作成日時(UTC) 指定したユニット作成日以降のユニットを検索します。
	createdTo	ユニット作成日時(UTC) 指定したユニット作成日以前のユニットを検索します。

※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx 形式の36文字のID

※2: {folderId}、{workflowId}、{unitId} はいずれか1つのみ必須となります。複数指定した場合はパラメータエラーとなります。

※3: 読取ユニット検索APIがリリースされた後に作成された読取ユニットに対して適切なステータスで検索が可能です。

リクエストの例

```
GET /units?folderId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx&unitName=My+unit
HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxx
Host: <request-host>
```

curlコマンドの例

```
curl "<request-host>/units?folderId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx&unitName=My+unit"
-X GET -H "apikey:xxxxxxxxxxxxxxxxxxxx"
```

レスポンスの形式

以下のレスポンス構造のJSON形式の文字列で返されます。

JSONオブジェクト	
unitId	ユニットID
unitName	ユニット名
status	ユニットのステータス 2: データ化中 13: データチェック中 15: コンペア中 16: コンペア完了 22: CSVダウンロード済
dataProcessingStatus	データ化のステータス 0: ページアップロード中(ページ分割中を含む) 200: データ化中 400: データ化完了 600: 一部データ化失敗
dataCheckStatus	データチェックのステータス 0: データチェック設定無し、 またはデータ化未完了 200: データチェック中 400: データチェック完了

dataCompareStatus	データコンペアのステータス 0: データ化中、または データチェック中、または データチェック完了かつ コンペア対象無し 200: コンペア中 400: コンペア完了
csvDownloadStatus	CSVダウンロードのステータス 0: データ化中、または データチェック未完了、または コンペア未完了、または 100: CSV未ダウンロード 400: CSVダウンロード完了
csvFileName	CSV出力のファイル名
partsImageCreationStatus	パーツ画像作成ステータス 0: パーツ画像生成対象無し 1: パーツ画像生成中 2: パーツ画像生成完了 3: パーツ画像生成一部失敗 4: パーツ画像生成エラー
folderId	フォルダーID
workflowId	ワークフローID
workflowName	ワークフロー名
createdAt	ユニット作成日時(UTC)

レスポンスの例

```
{
  "units": [
    {
      "unitId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
      "unitName": "My unit",
      "status": 22,
      "dataProcessingStatus": 400,
      "dataCheckStatus": 0,
      "dataCompareStatus": 0,
      "csvDownloadStatus": 400,
      "csvFileName": "test",
      "partsImageCreationStatus": 1,
      "folderId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
      "workflowId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
      "workflowName": "My Workflow",
      "createdAt": "2023-04-13T01:22:43.319337Z"
    },...
  ]
}
```

(5)CSVダウンロードAPI

読取ユニットに含まれる読取結果をCSVフォーマットで取得するAPI

URI	/units/{unitId}/csv		
メソッド	GET (text/csv)		
認証	apikey		
URL パラ メータ (パス)	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	unitId	○	ユニットID (UUID形式※1)
※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx 形式の36文字のID			

リクエストの例

```
GET /units/xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/csv HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxx
Host: <request-host>
```

curlコマンドの例

```
curl "<request-host>/units/xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/csv" -X GET -H
"apikey:xxxxxxxxxxxxxxxxxxxx" >C:\API\sample.csv
```

レスポンスの例

CSVの値(ヘッダー、ボディ)がレスポンスとして戻ります。
(上記サンプルのようにCSVファイルの保存先を指定すると、指定したフォルダにCSVファイルがダウンロードされます)

(6) 読取ユニット削除API

読取ユニットの削除をするAPI

URI	/units/{unitId}/delete		
メソッド	POST (application/json)		
認証	apikey		
URL パラ メータ (パス)	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	unitId	○	ユニットID (UUID形式※1)
	※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx 形式の36文字のID		

リクエストの例

```
POST /units/xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx/
delete HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxx
Host: <request-host>
```

curlコマンドの例

```
curl "<request-host>/units/xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx/delete" -X POST -H
"apikey:xxxxxxxxxxxxxxxxxxxx"
```

レスポンスの形式

以下のレスポンス構造のJSON形式の文字列で返されます。

JSONオブジェクト		
	unitId	ユニットID
	unitName	ユニット名

レスポンスの例

```
{
  "unitId": "xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx",
  "unitName": "test"
}
```

(7) 読取パーツ情報取得API

読取パーツ情報を取得するAPI

URI	/units/dataItems		
メソッド	GET (application/json)		
認証	apikey		
URL パラメータ (クエリ)	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	unitId	※2	ユニットID (UUID形式※1) 1つのユニットのみ指定可能です。 削除済みのユニットは検索の対象外となります。 指定ユニットのOCR処理がまだ完了していない場合は、検索できない(エラーが返却される)場合があります。
	pageId	※2	ページID (UUID形式※1) ページIDはカンマ区切りで複数指定が可能です。但し、複数ユニットに跨るページIDの指定はできません。 指定ページのOCR処理がまだ完了していない場合は、検索できない(エラーが返却される)場合があります。
	dataItemId	※2	読取パーツID (UUID形式※1) 読取パーツIDはカンマ区切りで複数指定が可能です。但し、複数ユニットに跨る読取パーツIDの指定はできません。
	detailDataItemId	※2	分割読取パーツID (UUID形式※1) 分割読取パーツIDはカンマ区切りで複数指定が可能です。但し、複数ユニットに跨る分割読取パーツIDの指定はできません。
※1 : xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx 形式の36文字のID ※2 : {unitId}、{pageId}、{dataItemId}、{detailDataItemId} はいずれか1つのみ必須となります。複数指定した場合はパラメータエラーとなります。			

リクエストの例

```
GET /units/dataItems?unitId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxx
Host: <request-host>
```

curlコマンドの例

```
curl "<request-host>/units/dataItems?unitId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" -X GET -H
"apikey:xxxxxxxxxxxxxxxxxxxx"
```

レスポンスの形式

以下のレスポンス構造のJSON形式の文字列で返されます。

JSONオブジェクト		
	dataItems	読取パーツ配列
	dataItemId	読取パーツID (UUID形式※1)
	detailDataItemId	分割読取パーツID (UUID形式※1)
	status	読取パーツのOCRステータス 1: 読取完了 2: 位置検出済み (OCR未実施) 3: 位置未検出 4: 読取失敗
	folderId	フォルダID (UUID形式※1)
	workflowId	ワークフローID (UUID形式※1)
	unitId	ユニットID (UUID形式※1)
	pageId	ページID (UUID形式※1)
	fileName	ファイル名
	columnNo	CSV出力のカラム番号 数字形式で、1がA列、2がB列...となります。
	columnName	CSV出力のヘッダ名
	x	ページ画像に対する座標値 (絶対値) ※2
	y	ページ画像に対する座標値 (絶対値) ※2
	width	読取パーツの幅 ※2
	height	読取パーツの高さ ※2
	result	読取文字列
	accuracy	読取尤度 ※現在、読取尤度は定型帳票 (読取範囲の指定) にのみ提供しております。非定型帳票 (旧モデル)、項目抽出帳票の場合は、常に0.0がセットされます。

		accuracyRank	<p>エンジンが指定した読取文字列の正確性に関する補助値</p> <p>0：情報なし(デフォルト)</p> <p>1：低い精度で正確に読めている</p> <p>2：少し低い精度で正確に読めている</p> <p>3：少し高い精度で正確に読めている</p> <p>4：高い精度で正確に読めている</p> <p>※データチェック画面の「精度」では、1が赤、4が緑の表示となります。</p>
		createdAt	読取パーツ作成日付(UTC)

※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx 形式の36文字のID

※2: 明細項目が含まれる項目抽出ユニットについては、明細の座標値(x, y)及び幅・高さ(width, height)は行単位での値となります。

レスポンスの例

```
{
  "dataItems": [
    {
      "dataItemId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
      "detailDataItemId": null,
      "status": 1,
      "folderId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
      "workflowId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
      "unitId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
      "pageId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
      "fileName": "sample.png",
      "columnNo": 1,
      "columnName": "銀行名",
      "x": 100,
      "y": 200,
      "width": 120,
      "height": 30,
      "result": "テスト銀行",
      "accuracy": 0.9976,
      "accuracyRank": 4,
      "createdAt": "2023-02-21 11:43:14.0"
    }, ...
  ]
}
```

(8) 読取ユニット内の全パーツ画像取得API

読取ユニットのパーツ画像を全て取得するAPI

URI	/units/{unitId}/downloadPartsImage		
メソッド	GET (application/json)		
認証	apikey		
URL パラ メータ (パス)	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	unitId	○	ユニットID (UUID形式※1)
	※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx 形式の36文字のID		
	※合計パーツ数(1ページ内の項目数 x ページ数)が3,000以下のユニットで利用できます。		
・「項目抽出」帳票には対応しておりません。			

リクエストの例

```
GET /units/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/downloadPartsImage HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxx
Host: <request-host>
```

curlコマンドの例

```
curl "<request-host>/units/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/downloadPartsImage" -X GET -H
"apikey:xxxxxxxxxxxxxxxxxxxx" -o "sample.zip"
```

レスポンスの形式

ZIPファイルがバイナリデータで返されます。

* 上記curlコマンド例にあるように、保存オプション(--output または -o)によりファイルとして保存もできます

ダウンロードZIPファイル構造

フォルダ	フォルダ内容
/Org/	パーツ画像
/Col/	CSVヘッダー名を入れたtxtファイル
/File/	ページのファイル名と行列情報を入れたtxtファイル
/Txt/	読み取り結果を入れたtxtファイル

また、各フォルダ内に生成されるファイル名は、以下の命名規則で作成されます。

通常項目の場合：

CSV出力先カラム名 + “_” + 読取ページID + “_” + 読取パーツID

分割項目の場合：

CSV出力先カラム名 + “_” + 読取ページID + “_” + 読取パーツID + “_” + 読取分割パーツID

(9) 読取ページ画像取得API

読取ページ画像を取得するAPI

URI	/units/images		
メソッド	GET (application/json)		
認証	apikey		
URL パラメータ (クエリ)	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	unitId	※2	ユニットID (UUID形式※1) 1つのユニットのみ指定可能です。 削除済みのユニットは検索の対象外となります。
	pageId	※2	ページID (UUID形式※1) ページIDはカンマ区切りで複数指定が可能です。但し、複数ユニットに跨るページIDの指定はできません。
	filenameType		ZIPファイルに含まれる画像のファイル名タイプ 1: ページID 2: アップロード時のファイル名 未指定の場合はページIDとなります。
※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx 形式の36文字のID ※2: {unitId}、{pageId} はいずれか1つのみ必須となります。複数指定した場合はパラメータエラーとなります。 ・「項目抽出」帳票には対応しておりません。			

リクエストの例

```
GET /units/images?pageId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx&filenameType=2 HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxxxx
Host: <request-host>
```

curlコマンドの例

```
curl
"<request-host>/units/images?pageId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx&filenameType=2" -X
GET -H "apikey:xxxxxxxxxxxxxxxxxxxx" -o "sample.zip"
```

レスポンスの形式

ZIPファイルがバイナリデータで返されます。

* 上記curlコマンド例にあるように、保存オプション(--output または -o)によりファイルとして保存もできます

(10) 読取ユニットID取得API

ワークフローに登録された読取ユニットのIDを取得するAPI

* 本APIは(4)読取ユニット検索APIで代用できます。標準API V1からの移行の際は(4)読取ユニット検索APIのご利用をお薦めします。

URI	/workflows/{workflowId}/units/ids		
メソッド	GET (application/json)		
認証	apikey		
URL パラメータ (パス)	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	workflowId	○	ワークフローID (UUID形式※1)
※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx 形式の36文字のID			
URL パラメータ (クエリ)	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	offset		取得開始位置 何件目(数値)から取得するかを設定します。 任意項目のため、パラメータを設定しない場合、デフォルト値0を設定します。
	limit		取得件数 1回あたりの取得件数(数値)を設定します。 任意項目のため、パラメータを設定しない場合と10000以上が設定された場合、デフォルト値10000を設定します。 ※2
※2: ユニットID取得APIは返却件数が最大10000件になります。			
・「項目抽出」帳票には対応していません。			

リクエストの例

```
GET /workflows/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx/units/ids?offset=0&limit=10000 HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxx
Host: <request-host>
```

curlコマンドの例

```
curl "<request-host>/workflows/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx/units/ids" -X GET -H  
"apikey:xxxxxxxxxxxxxxxxxxxxx"
```

レスポンスの形式

以下のレスポンス構造のJSON形式の文字列で返されます。

JSONオブジェクト

unitIds	ユニットIDの配列
offset	取得開始位置
limit	取得件数
totalCount	全件数

レスポンスの例

```
{  
  "unitIds": [  
    "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx",  
    "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx"  
  ],  
  "offset": 0,  
  "limit": 10000,  
  "totalCount": 20000  
}
```

(11) 読取ユニット状態取得API

読取ユニットの状態を取得するAPI

* 本APIは(4)読取ユニット検索APIで代用できます。標準API V1からの移行の際は(4)読取ユニット検索APIのご利用をお薦めします。

URI	/units/status?unitId={unitId}		
メソッド	GET (application/json)		
認証	apikey		
URL パラメータ (クエリ)	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	unitId	○	ユニットID (UUID形式※1) 1～100件まで指定できます。ユニットIDを複数指定する場合、ユニットIDの数だけパラメータに &unitId={ユニットID} を追加します。 (100件以上のIDを指定した場合は、エラーとはならず1～100件分のレスポンスとなり、101件以上のレスポンスは含まれません)

※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx 形式の36文字のID
・「項目抽出」帳票には対応していません。

リクエストの例

```
GET /units/status?unitId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxxxx
Host: <request-host>
```

curlコマンドの例

```
curl "<request-host>/units/status?unitId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx" -X GET -H
"apikey:xxxxxxxxxxxxxxxxxxxxx"
```

レスポンスの形式

以下のレスポンス構造のJSON形式の文字列で返されます。

JSONオブジェクト	
unitId	ユニットID
unitName	ユニット名
dataProcessingStatus	データ化のステータス 0: ページアップロード中(ページ分割中を含む)

	200: データ化中 400: データ化完了 600: 一部データ化失敗
dataCheckStatus	データチェックのステータス 0: データチェック設定無し、 またはデータ化未完了 200: データチェック中 400: データチェック完了
dataCompareStatus	データコンペアのステータス 0: データ化中、または データチェック中、または データチェック完了かつ コンペア対象無し 200: コンペア中 400: コンペア完了
csvDownloadStatus	CSVダウンロードのステータス 0: データ化中、または データチェック未完了、または コンペア未完了、または 100: CSV未ダウンロード 400: CSVダウンロード完了
partsImageCreationStatus	パーツ画像作成ステータス 0: パーツ画像生成対象無し 1: パーツ画像生成中 2: パーツ画像生成完了 3: パーツ画像生成一部失敗 4: パーツ画像生成エラー
workflowId	ワークフローID
revision	ワークフローのリビジョン
createdAt	ユニット作成日時(UTC)
updatedAt	ユニット更新日時(UTC)

レスポンスの例

```
[
  {
    "unitId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "unitName": "abc",
    "dataProcessingStatus": 400,
    "dataCheckStatus": 0,
    "dataCompareStatus": 0,
    "csvDownloadStatus": 400,
    "partsImageCreationStatus": 1,
    "workflowId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "revision": x
    "createdAt": "2021-04-21T08:20:45Z",
    "updatedAt": "2021-04-22T09:44:50Z"
  }, ...
]
```

}

(12) 仕分けユニット追加API

仕分けユニットを登録するAPI

URI	/sorter/add		
メソッド	POST (application/json)		
認証	apikey		
リクエスト データ形式	multipart/form-data		
リクエスト ボディ	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	sortConfigId	○	仕分けルールID (UUID形式※1)
	files	○	追加するファイル サポートしている拡張子は png、jpg、jpeg、pdf、tiff です。 ユニットに登録するファイルを1～100件まで指定できます。ファイルを複数指定する場合、ファイルの数だけ -F "files=@{ファイル名}" を追加します。
	sortUnitName		登録する仕分けユニット名 ※ ユニット名を1～128文字で指定します。半角全角英数記号かなカナ漢字を使用できますが、\ / : * ? " < > は使用不可文字です。 ※ 未指定の場合は仕分けルール名 + システム日付で設定されます。
	departmentId		部署ID OCRリクエスト数集計の仕分け先を指定します。形式は数字です。
	runSorting		継続して仕分け処理を開始する true: 送信する, false: 送信しない ※ デフォルト値はfalse
	sendOcr		継続してOCRに処理結果を送信する true: 送信する, false: 送信しない

		※ デフォルト値はfalse ※ このパラメータはrunSortingがtrueでないと作用しません。
※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx 形式の36文字のID		

リクエストの例

```
POST wf/api/standard/sorter/add HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxx
Host: <request-host>
Content-Type: multipart/form-data; boundary=-----680b5c02be1293c4

-----680b5c02be1293c4
Content-Disposition: form-data; name="files"; filename="test.png"
Content-Type: image/png
-----680b5c02be1293c4
Content-Disposition: form-data; name="sortUnitName"
test
-----680b5c02be1293c4
Content-Disposition: form-data; name="sortConfigId"
xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx
```

curlコマンドの例

```
curl "<request-host>/sorter/add" -X POST -H "apikey:xxxxxxxxxxxxxxxxxxxx" -F
"sortConfigId=xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx"
-F "files=@test.png" -F "sortUnitName=test"
```

レスポンスの形式

以下のレスポンス構造のJSON形式の文字列で返されます。

JSONオブジェクト		
	sortUnitId	仕分けユニットID
	sortUnitName	仕分けユニット名

レスポンスの例

```
{
  "sortUnitId": "xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx"
  "sortUnitName": "test"
}
```

(13) 仕分け実施API

仕分けユニットIDを元に仕分けを実施するAPI

URI	/sorter/sorting		
メソッド	POST (application/json)		
認証	apikey		
リクエスト データ形式	multipart/form-data		
リクエスト ボディ	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	sortUnitId	○	仕分けユニットID (UUID形式※1)
	departmentId		部署ID OCRリクエスト数集計の仕分け先を指定 します。形式は数字です。
	sendOcr		継続してOCRに処理結果を送信する true: 送信する, false: 送信しない ※ デフォルト値はfalse
※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx 形式の36文字のID			

リクエストの例

```
POST wf/api/standard/sorter/sorting HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxx
Host: <request-host>

Content-Type: multipart/form-data; boundary=-----680b5c02be1293c4

-----680b5c02be1293c4
Content-Disposition: form-data; name="sortUnitId"
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx
```

curlコマンドの例

```
curl "<request-host>/sorter/sorting" -X POST -H "apikey:xxxxxxxxxxxxxxxxxxxx" -d "sortUnitId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx"
```

レスポンスの形式

以下のレスポンス構造のJSON形式の文字列で返されます。

JSONオブジェクト

sortUnitId	仕分けユニットID
sortUnitName	仕分けユニット名

レスポンスの例

```
{  
  "sortUnitId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx"  
  "sortUnitName": "test"  
}
```

(14) 仕分け結果OCR送信API

仕分けユニットIDで仕分けした結果をOCRに転送します。

URI	/sorter/sendOcr		
メソッド	POST (application/json)		
認証	apikey		
リクエスト データ形式	multipart/form-data		
リクエスト ボディ	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	sortUnitId	○	仕分けユニットID (UUID形式※1)
	departmentId		部署ID OCRリクエスト数集計の仕分け先を指定します。形式は数字です。
※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx 形式の36文字のID			

リクエストの例

```
POST wf/api/standard/sorter/sendOcr HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxx
Host: <request-host>

Content-Type: multipart/form-data; boundary=-----680b5c02be1293c4

-----680b5c02be1293c4
Content-Disposition: form-data; name="sortUnitId"
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

curlコマンドの例

```
curl "<request-host>/sorter/sendOcr" -X POST -H "apikey:xxxxxxxxxxxxxxxxxxxx" -d "sortUnitId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
```

レスポンスの形式

以下のレスポンス構造のJSON形式の文字列で返されます。

JSONオブジェクト		
	sortUnitId	仕分けユニットID

	sortUnitName	仕分けユニット名
--	--------------	----------

レスポンスの例

```
{
  "sortUnitId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx"
  "sortUnitName": "test"
}
```

(15) 仕分け結果CSVエクスポートAPI

仕分けユニットIDを元にCSV形式でエクスポートを行います。

URI	/sorter/csvDownload		
メソッド	POST (text/csv)		
認証	apikey		
リクエスト データ形式	multipart/form-data		
リクエストボディ	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	sortUnitId	○	仕分けユニットID (UUID形式※1)
※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx 形式の36文字のID			

リクエストの例

```
post wf/api/standard/sorter/csvDownload HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxx
Host: <request-host>

Content-Type: multipart/form-data; boundary=-----680b5c02be1293c4

-----680b5c02be1293c4
Content-Disposition: form-data; name="sortUnitId"
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

curlコマンドの例

```
curl "<request-host>/sorter/csvDownload" -X POST -H "apikey:xxxxxxxxxxxxxxxxxxxx" -d "sortUnitId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" >C:\API\sample.csv
```

レスポンスの例

CSVの値(ヘッダー、ボディ)がレスポンスとして戻ります。
(上記サンプルのようにCSVファイルの保存先を指定すると、指定したフォルダにCSVファイルがダウンロードされます)

(16)仕分けユニット検索API

仕分けユニットのステータスを返却します

URI	/sorter/status		
メソッド	POST(application/json)		
認証	apikey		
リクエスト データ形式	multipart/form-data		
リクエストボ ディ	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	sortUnitId	○	仕分けユニットID(UUID形式※1)
※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx 形式の36文字のID			

リクエストの例

```
POST wf/api/standard/sorter/status HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxx
Host: <request-host>

Content-Type: multipart/form-data; boundary=-----680b5c02be1293c4

-----680b5c02be1293c4
Content-Disposition: form-data; name="sortUnitId"
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

curlコマンドの例

```
curl "<request-host>/sorter/status" -X POST -H "apikey:xxxxxxxxxxxxxxxxxxxx" -d
"sortUnitId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
```

レスポンスの形式

以下のレスポンス構造のJSON形式の文字列で返されます。

JSONオブジェクト		
	statusCode	仕分けステータスを表すコード値(詳細は後述)
	statusName	仕分けステータス名称
	pageCountAll	仕分け対象の全ページ数
	statusList	仕分け先別のOCR処理一覧

	readingUnitId	仕分け先の読み取りユニットID ※ 仕分けされたページがないか、OCRIに送信していない仕分け先は0が設定されます。
	sendCountSuccess	OCR送信済み件数(データ化完了件数)
	sendCountFailure	OCR送信エラー件数(データ化失敗件数)
	sendCountAll	OCR送信対象件数
	statusCode	仕分けステータスを表すコード値(詳細は後述)
	statusName	ステータス名称

レスポンスの例

```
{
  "statusCode": 130,
  "statusName": "コンペア完了",
  "pageCountAll": 1,
  "statusList": [
    {
      "readingUnitId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
      "sendCountSuccess": 1,
      "sendCountFailure": 0,
      "sendCountAll": 1,
      "statusCode": 130,
      "statusName": "コンペア完了"
    },
    {
      "readingUnitId": "0",
      "sendCountSuccess": 0,
      "sendCountFailure": 0,
      "sendCountAll": 0,
      "statusCode": 0,
      "statusName": ""
    }
  ]
}
```

(17) 仕分けユニット削除API

仕分けユニットの削除をするAPI

URI	/sorter/delete		
メソッド	POST (application/json)		
認証	apikey		
リクエスト データ形式	multipart/form-data		
リクエストボディ	以下のパラメータを設定します。		
	パラメータ名	必須	パラメータ値
	sortUnitId	○	仕分けユニットID (UUID形式※1)
※1: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx 形式の36文字のID			

リクエストの例

```
POST wf/api/standard/sorter/delete HTTP/2
apikey: xxxxxxxxxxxxxxxxxxxx
Host: <request-host>

Content-Type: multipart/form-data; boundary=-----680b5c02be1293c4

-----680b5c02be1293c4
Content-Disposition: form-data; name="sortUnitId"
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx
```

curlコマンドの例

```
curl "<request-host>/sorter/delete" -X POST -H "apikey:xxxxxxxxxxxxxxxxxxxx" -d
"sortUnitId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx"
```

レスポンスの形式

以下のレスポンス構造のJSON形式の文字列で返されます。

JSONオブジェクト		
	sortUnitId	仕分けユニットID
	sortUnitName	仕分けユニット名

レスポンスの例

```
{
  "sortUnitId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx"
```

```
"sortUnitName": "test"  
}
```

5. 付録

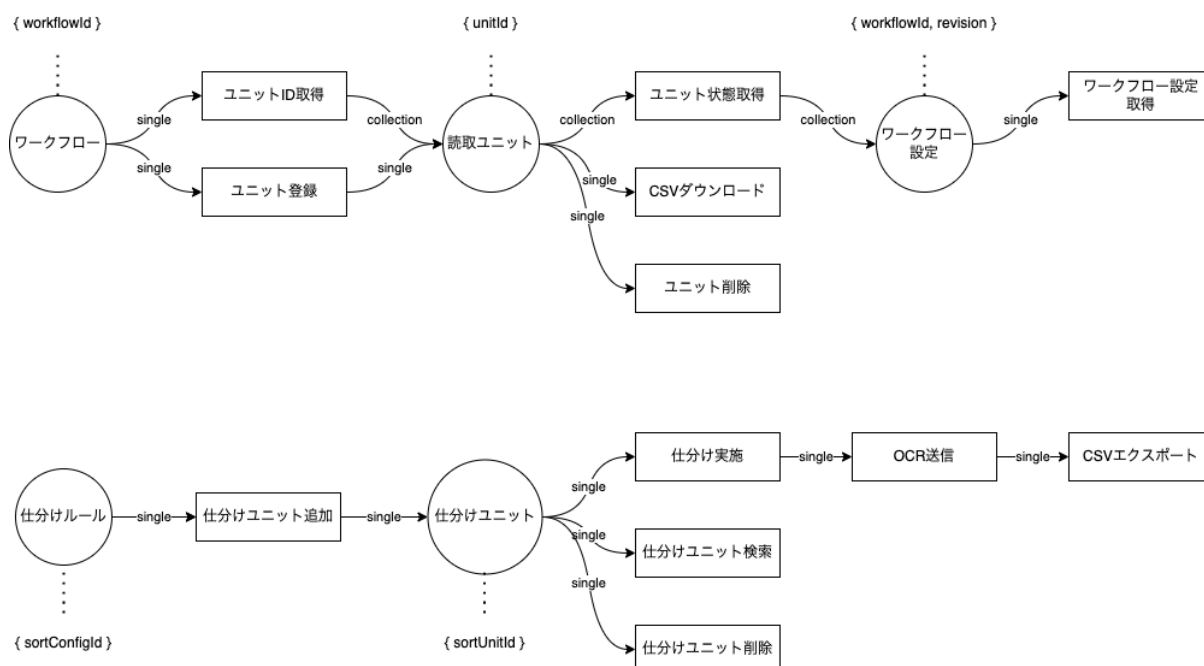
(1) エラーコード・メッセージ一覧

メッセージ	内容	エラーコード
request had invalid authentication credentials	認証情報が正しくありません。APIキーをご確認ください。	1 300001
no handler found	存在しないエンドポイントをリクエストしています。	2
request method is not allowed	サポートしないHTTPメソッドでリクエストしています。	3
request was forbidden	APIにアクセスする権限がありません。接続元のIPアドレスをご確認ください。	300002
{ } is required	必要なパラメータが指定されていません。	10001
size of { } must be less than or equal to { }	一度に指定できるパラメータ数の上限を超えています。	10002
{ } had invalid characters (Please check the API specification.)	利用できない文字がパラメータに指定されています。	10003
{ } was invalid format (Please check the API specification.)	不正な書式の値がパラメータに指定されています。	10005
{ } must not be blank (Remove the field or set null if it is not needed.)	パラメータに空文字が指定されています。不要であればこのパラメータを削除してください。	10006
{ } must not contain blanks	パラメータに空文字が指定されています。不要であればこのパラメータを削除してください。	10007
{ } length must be between {1} and {2}	パラメータの文字数は{1}から{2}の間の文字数で指定してください。	10008
{ } was invalid value (Please check the API specification.)	利用できない値がパラメータに指定されています。	10009
one of { } is required	{ }のうちのいずれか一つのパラメータを指定してください。	10010
{ } was invalid pattern (Please check the API specification.)	利用できない形式の値がパラメータに指定されています。正しい形式で指定してください。	10011
{ } had invalid characters (Please check the API specification.)	利用できない文字がパラメータに指定されています。	110003
file format (extension) is unsupported	サポートしない形式(拡張子)のファイルがアップロードされています。	210001
tiff compression format is unsupported	サポートしないTIFF圧縮形式のファイルがアップロードされています。	210002
encrypted pdf is unsupported	サポートしないパスワードつきPDFファイルがアップロードされています。	210003
specified department does not exist	存在しないか誤った部署IDが指定されています。	20001
specified resource does not exist	指定のリソースは存在していないか、アクセスできません。	120001

メッセージ	内容	エラーコード
workflow was not ready (Please save the workflow and try again.)	指定のワークフローは保存されていません。ワークフローを保存してもう一度やり直してください。	120002
unit process was not completed. (Please try again after the unit process is completed.)	指定の読取ユニットはデータ化が完了していません。データ化が完了したらもう一度やり直してください。	120003
search conditions that retrieve multiple reading units are not allowed.(Please set the search condition within one unit.)	検索条件で複数のユニットに跨るパラメータ指定はできません。検索条件のパラメータは1つのユニット内で指定してください。	120004
page process was not completed. (Please try again after the page process is completed.)	指定の読取ページは処理が完了していません。ページ処理が完了したらもう一度やり直してください。	120005
workflow for recipient was not determined (Please try again after determining workflow for recipient.)	送信先のワークフローが確定されていません。ワークフロー確定後もう一度やり直してください。	120006
sort unit process was not completed (Please try again after the sort unit process is completed.)	指定の仕分けユニットは処理が完了していません。処理が完了したらもう一度やり直してください。	120007
pre-sorted pages were not found (Please try again after uploading pages.)	仕分け前トレイにページが存在しません。ファイルアップロード後もう一度やり直してください。	120008
sorting was not yet done (Please try again after sorting.)	仕分けがされていません。仕分け実施後もう一度やり直してください。	120009
already sent to OCR	すでにOCR送信済みです。	120010
not yet sent to OCR (Please try again after sending to OCR)	OCR未送信です。OCR送信後もう一度やり直してください。	120011
sending to OCR was not completed (Please try again after the reading unit process is completed.)	OCR送信が完了していません。対象の読み取りユニットの処理が完了後もう一度やり直してください。	120012
conflict error has occurred (Please try again after the process is completed.)	競合が発生しました。競合した指定リソースの処理が完了した後、再度やり直してください。	120013
We do not support old units. Please create a new unit and try again.	V1.110以前に作られたユニットです。新しいユニットでもう一度やり直してください。	120014
specified resource is not available from the API	指定のリソースはAPIからご利用できません	120015
server error has occurred (Please contact our Customer Support Team.)	サーバーエラーが発生しました。カスタマーサポートチームに問い合わせてください。	80000 180000 280000 380000

(2) 各APIにおける入出力の関係

各APIの入出力において、リソースのIDは次の図に示す関係にあります。



各リソースは次の表に示すIDで識別できます。

リソース	識別子と説明
ワークフロー	ワークフローIDで識別します。 ワークフローIDはWeb画面から取得できます。
読取ユニット	ユニットIDで識別します。 ユニットIDはWeb画面から取得できます。また、ユニット登録APIの結果としても、ユニットID取得APIの対象としても取得できます。
ワークフロー設定	ワークフローIDとリビジョン番号で識別します。 ワークフローの設定はリビジョン番号によって変更履歴管理されています。読取ユニットはワークフロー設定の特定リビジョンに対して登録されます。
仕分けルール	仕分けルールIDで識別します。 仕分けルールIDはWeb画面から取得できます。
仕分けユニット	仕分けユニットIDで識別します。 仕分けユニットIDはWeb画面から取得できます。また、仕分けユニット追加APIの結果としても、取得できます。

(3)Elastic Sorterの仕分けステータス

仕分けステータスを表すコード値は以下の通りです。

ステータスコード	ステータス名称
30	ページ登録中
35	ページ登録エラー
40	ページ登録完了
50	仕分け中
55	仕分けエラー
60	仕分け完了
990	全て仕分け不可
87	OCR送信中
88	OCR送信エラー
100	データチェック中
120	コンペア中
130	コンペア完了
140	CSV出力中
145	CSV出力エラー
150	CSV出力完了



〈 発行元 〉

AI inside 株式会社

〒150-0002

東京都渋谷区渋谷3-8-12 渋谷第一生命ビルディング4階