

DX Suite クラウド API連携説明書

2025年4月17日
第 1.24 版

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

改版履歴

版数	DX Suite Ver	発行日	変更内容
第1.0版	1.1.19	2018/06/01	初版
第1.1版	1.2.3	2018/07/20	<ul style="list-style-type: none"> 各APIのエラーコードを修正しました。 サンプルコードを修正しました。
第1.2版	1.2.7	2018/08/28	<ul style="list-style-type: none"> ページ追加APIのエラーコードを追加しました。
第1.3版	1.2.18	2018/12/11	<ul style="list-style-type: none"> ページ追加APIのエラーコードを追加しました。
第1.4版	1.3.0	2018/12/25	<ul style="list-style-type: none"> ページ追加APIのパラメータを追加しました。 ページ追加APIのエラーコードを追加しました。
第1.5版	1.5.0	2019/03/26	<ul style="list-style-type: none"> 読取ユニットの検索APIを追加しました。 読取ユニット内の全パーツ画像の取得APIを追加しました。 仕分けAPIを追加しました。 サンプルコードを見直しました。 その他、軽微な見直しを行いました。
第1.6版	1.5.2	2019/04/09	<ul style="list-style-type: none"> 各アプリケーションのステータス仕様を追加しました。 サンプルコードを修正しました。
第1.7版	1.5.5	2019/05/14	<ul style="list-style-type: none"> ドキュメント検索APIを追加しました。 読取ユニット検索APIのパラメータを追加しました。 読取ページの画像取得APIを追加しました。 読取パーツの情報取得APIを追加しました。 その他、軽微な見直しを行いました。
第1.8版	1.5.7	2019/06/11	<ul style="list-style-type: none"> その他、軽微な見直しを行いました。
第1.9版	1.7.0	2019/09/25	<ul style="list-style-type: none"> 読取ユニット削除APIを追加しました。 仕分けユニット削除APIを追加しました。 その他、軽微な見直しを行いました。
第1.10版	1.7.6	2019/12/24	<ul style="list-style-type: none"> 各API個別の仕様にHTTPリクエストとHTTPレスポンスの例を追加しました。 ステータスコードを把握しやすくするためのワークフロー図を追加しました。 仕分け処理の実施API・仕分け結果処理をOCRに送信APIの、リクエストパラメータが不正時のエラーコードをAPI全体で統一しました。(199→103) 仕分け結果CSVエクスポートAPIのエラーコード記述誤りを修正。(199→103) Elastic Sorter の仕分けステータスから発生しないエラーコード(10,15,105,125)を削除しました。 その他、軽微な見直しを行いました。

版数	DX Suite Ver	発行日	変更内容
第1.11版	1.8	2020/02/25	<ul style="list-style-type: none"> 仕分けユニット追加のHTTPリクエストサンプルの誤りを修正しました。 読取ページ追加APIのHTTPリクエストサンプルの表記を修正しました。 各アプリケーションのステータス仕様のIntelligent OCR の読み取りステータスに、6(自動画像補正・パーツ生成 前処理)を追加しました。 読取パーツの情報の取得のレスポンスの誤りを修正しました。
第1.12版	1.10	2020/04/28	<ul style="list-style-type: none"> ドキュメントの検索API パラメータ内の説明誤りを修正しました。 読取ユニット内の全パーツ画像取得APIにて、ダウンロードZIPファイル 構造を追記しました。 読取パーツの情報取得API エラーコード(103)の意味の誤りを修正しました。 読取ページ追加APIに、パラメータとエラーコードを追記しました。 仕分けユニット追加APIに、パラメータとエラーコードを追記しました。 仕分け処理の実施APIに、パラメータとエラーコードを追記しました。 仕分け結果をOCRに送信APIに、パラメータとエラーコードを追記しました。
第1.13版	1.10	2020/05/26	<ul style="list-style-type: none"> リクエスト数の集計が部署毎に可能になったため、3 API共通の仕様を更新しました。
第1.14版	1.12	2020/06/30	<ul style="list-style-type: none"> 仕分けユニット検索のレスポンスの誤りを修正しました。
第1.15版	1.15	2020/10/27	<ul style="list-style-type: none"> 読取ユニット内の全パーツ画像取得APIのサンプルコードを修正しました。 読取ページの画像の取得APIのサンプルコードを修正しました。 読取ページの画像の取得APIのダウンロード命名規則を追記しました。 読取パーツの情報の取得APIの文言を修正しました。
第1.16版	1.16	2020/11/24	<ul style="list-style-type: none"> 読取ページ追加APIのhttpリクエスト例を修正しました。 仕分けユニット追加APIのhttpリクエスト例、レスポンス例を修正しました。 5.各アプリケーションのステータス仕様 (1)Intelligent OCRの読み取りステータスに「111:非定型読取中」「112:非定型読取完了」を追加しました。 5.各アプリケーションのステータス仕様(3)各ステータス別ワークフロー図で読み取りステータス「4:仕分け処理中、5:仕分け完了、6:仕分けエラー」を削除、「111:非定型読取中」「112:非定型読取完了」を追加しました。
第1.17版	1.17	2020/12/22	<ul style="list-style-type: none"> 仕分けユニット検索APIのstatusList配列内のstatusCodeのレスポンスと参照先のステータス一覧表を修正しました。

版数	DX Suite Ver	発行日	変更内容
第1.18版	1.21	2021/06/17	<ul style="list-style-type: none"> 読取ページ追加APIのエラーコード[112]の内容を見直しました。 読取ページ追加APIのエラーコード[117]を削除しました。
第1.19版	1.27	2021/12/14	<ul style="list-style-type: none"> 6.サンプルコード (4) 読取結果CSVのエクスポートのタイトル文言を修正しました。
第1.20版	1.28	2022/01/18	<ul style="list-style-type: none"> 仕分けユニット追加APIのエラーコード[406]に記載のあるパラメータ名を修正しました。
第1.21版	1.29	2022/02/15	<ul style="list-style-type: none"> (1)ドキュメントの検索のパラメータの必須項目を修正しました。 6.サンプルコード (1)ドキュメントの検索のサンプルコードを修正しました。
第1.22版	1.100	2022/10/5	<ul style="list-style-type: none"> 表紙に「この仕様書はwith DX Suite / with AI inside 製品にも共通の内容です」と記載しました。 最終ページに発行元情報を追加しました。 各ページのフッターを削除しました。
第1.23版	1.102	2022/11/18	<ul style="list-style-type: none"> 改ページ位置を調整しました。
第1.24版	1.118	2025/04/17	<ul style="list-style-type: none"> 「errCode」の表記を「errorCode」に修正しました。

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。

目次

1. はじめに	6
2. APIの一覧	7
3. API共通の仕様	8
(1) APIの概要	8
(2) リクエストURI	8
(3) 認証	8
(4) 処理パラメータの指定	9
(5) 正常、エラーの判断	9
(5-1) 正常時の応答	9
(5-2) エラー発生時の応答	9
(6) リクエスト数	10
4. 各API個別の仕様	11
(1) ドキュメントの検索	11
a. HTTPリクエスト例	12
b. HTTPレスポンス ボディ例	12
(2) 読取ユニット検索	13
a. HTTPリクエスト例	15
b. HTTPレスポンス ボディ例	15
(3) 読取ユニット内の全パーツ画像取得	16
a. HTTPリクエスト例	17
b. HTTPレスポンス ボディ例	17
c. ダウンロードZIPファイル 構造	17
(4) 読取結果CSVのエクスポート	18
a. HTTPリクエスト例	19
b. HTTPレスポンス ボディ例	19
(5) 読取ユニット削除	20
a. HTTPリクエスト例	21
b. HTTPレスポンス ボディ例	21
(6) 読取ページ追加	22
a. HTTPリクエスト例	24
b. HTTPレスポンス ボディ例	24
(7) 読取ページの画像の取得	25
a. HTTPリクエスト例	26
b. HTTPレスポンス ボディ例	26
c. ダウンロードZIPファイル 名	26
(8) 読取パーツの情報の取得	27
a. HTTPリクエスト例	29
b. HTTPレスポンス ボディ例	29
(9) 仕分けユニット追加	31
a. HTTPリクエスト例	33

b. HTTPレスポンス ボディ例	33
(10) 仕分け処理の実施	34
a. HTTPリクエスト例	35
b. HTTPレスポンス ボディ例	35
(11) 仕分け結果をOCRに送信	36
a. HTTPリクエスト例	37
b. HTTPレスポンス ボディ例	37
(12) 仕分け結果 CSV エクスポート	38
a. HTTPリクエスト例	39
b. HTTPレスポンス ボディ例	39
(13) 仕分けユニット検索	40
a. HTTPリクエスト例	41
b. HTTPレスポンス ボディ例	41
(14) 仕分けユニット削除	43
a. HTTPリクエスト例	44
b. HTTPレスポンス ボディ例	44
5. 各アプリケーションのステータス仕様	45
(1) Intelligent OCR の読み取りステータス	45
(2) Elastic Sorter の仕分けステータス	47
(3) ステータス別ワークフロー図	48
6. サンプルコード	49
(1) ドキュメント検索	49
(2) 読取ユニット検索	49
(3) 読取ユニット内の全パーツ画像取得	49
(4) 読取結果CSVのエクスポート	50
(5) 読取ユニット削除	50
(6) 読取ページ追加	50
(7) 読取ページの画像の取得	50
(8) 読取パーツの情報の取得	51
(9) 仕分けユニット追加	51
(10) 仕分け処理の実施	51
(11) 仕分け結果をOCRに送信	51
(12) 仕分け結果 CSV エクスポート	51
(13) 仕分けユニット検索	51
(14) 仕分けユニット削除	52

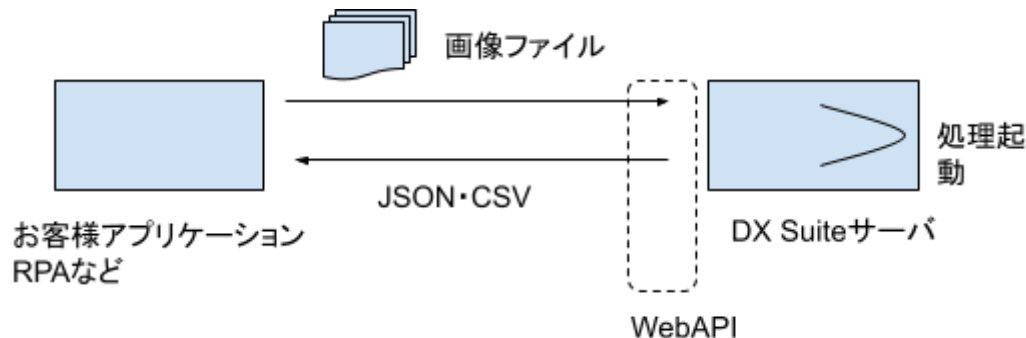
1. はじめに

DX Suite API 連携は、お客様が構築された他のアプリケーションや、RPA と DX Suite を Web API を使って自動連携させる機能です。これによりWeb画面のユーザー操作を行わずに DX Suite のシステム処理を自動化することが可能になります。

DX Suite API で操作ができるのは、帳票の読み取りからCSV出力までです。マスタ操作、帳票定義の作成などはできませんので、これらはWeb画面を使ってください。

DX Suite API を使うと、以下のようなことができます。

- RPA を使って帳票のアップロードや結果のダウンロードを自動化する
 - RPA が HTTP による Web API との連携をサポートする場合、DX Suite の操作をロボット化することができます。
- 自社のイメージ基盤などのソリューションと自動連携
 - 自社にあるイメージシステムが Web API との連携をサポートする場合、スキャニングとDX Suite のアップロードを自動化することができます。
- 大量の画像をアップロードする
 - DX Suite クラウドへ大量の画像をアップロードしたい場合、Web API を呼び出すアプリケーションを用意すればタイムアウトを気にすることなくアップロードができます。
- タイマーによるアップロード・ダウンロード
 - 画像をアップロードしたり、CSV をダウンロードしたりするアプリケーションをタイマー起動するように設定すれば、日々の定例業務を忘れることなく実施することができます。



2. APIの一覧

現在、提供している API は以下です。

	カテゴリ	処理	概要	提供開始時期
1	ドキュメント	検索	ドキュメントの検索	2019年5月
2	読取ユニット	検索	読取ユニットの検索	2019年3月
3		パーツ画像取得	読取ユニット内の全パーツ画像の取得	2019年3月
4		CSVエクスポート	読取結果CSVのエクスポート	2018年5月
5		ユニットの削除	読取ユニットの削除	2019年9月
6	読取ページ	追加	読取ページの追加(ユニットの追加)	2018年5月
7		ページ画像取得	読取ページの画像の取得	2019年5月
8	読取パーツ	パーツ情報取得	読取パーツの情報を取得	2019年5月
9	仕分けユニット	追加	仕分けユニットの追加	2019年3月
10		仕分け処理	仕分け実施	2019年3月
11		OCR送信	仕分け結果をOCRに送信	2019年3月
12		CSVエクスポート	仕分け結果CSVのエクスポート	2019年3月
13		検索	仕分けユニットの検索	2019年3月
14		ユニットの削除	仕分けユニットの削除	2019年9月

3. API共通の仕様

(1) APIの概要

本APIの概要は以下です。

項目	説明
プロトコル	HTTPS GET/POSTメソッドにより行います。なお、PUT や DELETE などのメソッドは使用しません。いずれのメソッドを使うかはAPI個別の仕様を参照してください。
リクエストデータ形式	ファイルのアップロードを行う API は、multipart/form-data 形式でアップロードします。そのほかのリクエストは、application/x-www-form-urlencoded 形式です。
レスポンスデータ形式	JSONフォーマットです。画像やCSVなどをダウンロードする API はバイトデータがダウンロードされます。
文字コード	JSONの文字コードは UTF-8 です。 CSVファイルの文字コードは Shift_JIS (MS932) または UTF-8 です。 (帳票定義の設定に従います)

(2) リクエストURI

リクエストURIは以下ようになります。

https://(サブドメイン名).dx-suite.com/(サービス名)/api/v1/(機能名)

項目	説明
サブドメイン名	お客様毎に割り当てられた値です。
サービス名	ConsoleWeb または Sorter になります。
機能名	API毎に異なります。

(3) 認証

本APIを実行するには認証パラメータが必要になります。

認証パラメータにはAPIキーを使用します。

- ・APIキーは、APIキー設定画面より作成・削除することができます。
- ・APIキーは、一意の文字列で組織に紐付けられています。
- ・API実行時にAPIキーを指定するには、HTTPヘッダ「X-ConsoleWeb-APIKey」を使用します。

例: APIキーが **your-api-key...** の場合

X-ConsoleWeb-APIKey: **your-api-key...**

(4) 処理パラメータの指定

処理に必要なパラメータは、URIもしくはクエリ文字列として指定します。それぞれのAPI個別の仕様を参照してください。

(5) 正常、エラーの判断

正常時は HTTPステータスコード 200 を応答します。異常時は異常内容により HTTPステータスコード 4xx/5xx を応答します。

(5-1) 正常時の応答

正常時は HTTPステータスコード 200 を応答します。正常に処理された結果は、コンテンツボディとして応答されます。コンテンツは、application/json もしくは、画像のMIMEタイプとなります。

正常時のコンテンツボディの形式はAPI毎の仕様を参照してください。

(5-2) エラー発生時の応答

エラー時は HTTPステータスコード 4xx/5xx を応答します。エラーの詳細はコンテンツボディにJSON形式で返されます。エラー発生時のJSONは以下の形式となります。

キー	値
status	エラー
errorCode	エラー状態を示すID
message	エラーメッセージ

HTTPステータスコードの一覧

ステータスコード	意味
400	Bad Request リクエストが正しく行われなかった
401	Unauthorized 認証パラメータがないか、認証されていない
403	Forbidden 現在の認証では指定されたリソースにアクセスできない
404	Not Found 現在の認証では指定されたリソースが存在しない
405	Method Not Allowed 指定されたメソッドは使うことができない
406	Not Acceptable 指定されたリクエストは受付できない
500	Internal Server Error サーバ側でエラーが発生した

(6) リクエスト数

APIのリクエスト数は、DX Suite のリクエスト数照会画面で参照することができます。

部署毎にリクエスト数を集計したい場合は、API実行時にユーザーIDを指定してください。

- ・ユーザーIDを指定しない場合は、部署は「未設定」となります。
- ・ユーザーIDを指定可能なAPIや指定方法は、「各API個別の仕様」を確認ください。

4. 各API個別の仕様

各APIの個別の仕様は以下になります。

(1) ドキュメントの検索

ドキュメントを検索します。

URI	/ConsoleWeb/api/v1/documents		
メソッド	GET		
認証パラメータ	APIキー (X-ConsoleWeb-ApiKeyヘッダ)		
パラメータ	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
	{docsetId}		フォルダID 複数をカンマ区切りで指定可能 ※ 存在するIDを設定してください。
	{documentName}		ドキュメント名 ※ 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
レスポンス	正常の場合は、以下の構造のJSON。		
	フィールド	型	説明
	status	文字列	success : 正常
	errorCode	数値	正常を示すID(0)
	message	文字列	メッセージ
	documents	配列	ドキュメントの配列
	id	数値	ドキュメントのID
	docsetId	数値	フォルダのID
	name	文字列	ドキュメントの名前
	エラーの場合は、エラーコードを含む以下の構造のJSON。		
	フィールド	型	説明
	status	文字列	error : エラー

	errorCode	数値	エラー状態を示すID
	message	文字列	メッセージ
エラーコード	HTTPステータスコードと、戻りJSONのerrorCodeにセットされるコードは以下です。		
	HTTP Status	errorCode	意味
	200	0	処理は正常に受け付けられた
	404	103	該当のフォルダが存在しないか、ドキュメントが存在しないか、アクセス権限がありません。
	400	104	入力パラメータの形式に誤りがあります
	401	101	認証エラーです。APIキーが誤っているか、APIキーの指定方法が正しくありません。
	500	199	その他の不明なエラーです。リクエストが正しいかを確認いただき、再度発生する場合はお問い合わせください。

a. HTTPリクエスト例

※ 青字の値は実行環境によって異なります。

```
GET /ConsoleWeb/api/v1/documents?docsetId=123 HTTP/1.1
Host: company-name.dx-suite.com
X-ConsoleWeb-ApiKey: your-api-key...
```

b. HTTPレスポンス ボディ例

※ 実際のJSONには改行やインデントはありません。

```
{
  "status": "success",
  "errorCode": 0,
  "message": "ReadingUnit listing.",
  "documents": [
    {
      "id": 456,
      "docsetId": 123,
      "name": "MyTest"
    }
  ]
}
```

(2) 読取ユニット検索

読取ユニットを検索します。

URI	/ConsoleWeb/api/v1/reading/units	
メソッド	GET	
認証パラメータ	APIキー (X-ConsoleWeb-ApiKeyヘッダ)	
パラメータ	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
	{status}	ユニットのステータス値 複数をカンマ区切りで指定可能
	{docsetId}	※1 フォルダID 複数をカンマ区切りで指定可能
	{documentId}	※1 ドキュメントID 複数をカンマ区切りで指定可能
	{readingUnitId}	※1 読取ユニットID 複数をカンマ区切りで指定可能
	{name}	読取ユニット名 複数をカンマ区切りで指定可能
	{createdFrom}	作成日の範囲の開始日時 (yyyy-mm-dd HH:MM:SSで指定) ※ スペースが含まれるため、URLエンコードして設定します。 例) 2019-01-01 10:00:00 → 2019-01-01+10%3A00%3A00
	{createdTo}	作成日の範囲の終了日時 (yyyy-mm-dd HH:MM:SSで指定) ※ スペースが含まれるため、URLエンコードして設定します。
※1 : {docsetId}、{documentId}、{readingUnitId} 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。		
レスポンス	正常の場合は、以下の構造のJSON。	
	フィールド	型 説明
	status	文字列 success : 正常
	errorCode	数値 正常を示すID(0)
	message	文字列 メッセージ
	readingUnits	配列 ユニットの配列

ユニットのステータスは「[5.1 Intelligent OCR の読み取りステータス](#)」を参照
エラーの場合は、エラーコードを含む以下の構造のJSON。

フィールド	型	説明
status	文字列	error : エラー
errorCode	数値	エラー状態を示すID
message	文字列	メッセージ

エラーコード	HTTPステータスコードと、戻りJSONのerrorCodeにセットされるコードは以下です。		
	HTTP Status	errorCode	意味
	200	0	処理は正常に受付られた
	400	104	入力パラメータの形式に誤りがあります
	401	101	認証エラーです。APIキーが誤っているか、APIキーの指定方法が正しくありません。
	404	103	指定した{docsetId}, {documentId}, {readingUnitId}に紐づく情報が存在しない
	500	199	その他の不明なエラーです。リクエストが正しいかを確認いただき、再度発生する場合はお問い合わせください。

a. HTTPリクエスト例

※ 青字の値は実行環境によって異なります。

```
GET /ConsoleWeb/api/v1/reading/units?documentId=123 HTTP/1.1
Host: company-name.dx-suite.com
X-ConsoleWeb-APIKey: your-api-key...
```

b. HTTPレスポンス ボディ例

※ 実際のJSONには改行やインデントはありません。

```
{
  "status": "success",
  "errorCode": 0,
  "message": "ReadingUnit listing.",
  "readingUnits": [
    {
      "id": 135,
      "name": "サンプル",
      "status": 13,
      "csvFileName": "sample.csv",
      "docsetId": 123,
      "documentId": 789,
      "documentName": "申込書",
      "createdAt": "2019-03-03 12:34:56.0"
    }
  ]
}
```


(3) 読取ユニット内の全パーツ画像取得

読取ユニット内の全パーツ画像をZIPファイルで取得します。

URI	/ConsoleWeb/api/v1/reading/units/{id}/partsDownload		
メソッド	GET		
認証パラメータ	APIキー (X-ConsoleWeb-ApiKeyヘッダ)		
パラメータ	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
	{id}	○	ダウンロードを行うユニットのID (URLに指定します)
レスポンス	正常の場合は、対象のZIPファイル。 エラーの場合は、エラーコードを含む以下の構造のJSON。		
	フィールド	型	説明
	status	文字列	success : 正常 error : エラー
	errorCode	数値	エラー状態を示すID
	message	文字列	メッセージ
エラーコード	HTTPステータスコードと、戻りJSONのerrorCodeにセットされるコードは以下です。		
	HTTP Status	errorCode	意味
	200	0	処理は正常に受付られた
	401	101	認証エラーです。APIキーが誤っているか、APIキーの指定方法が正しくありません。
	404	103	該当のユニットIDは存在しないか、アクセス権限がありません。
	406	105	このユニットIDはまだパーツ画像が準備されていません。 例えば、ページ追加後にパーツ画像生成がまだ完了していない場合などです。
	500	199	その他の不明なエラーです。リクエストが正しいかを確認いただき、再度発生する場合はお問い合わせください。

a. HTTPリクエスト例

※ 青字の値は実行環境によって異なります。

```
GET /ConsoleWeb/api/v1/reading/units/12345/partsDownload HTTP/1.1
Host: company-name.dx-suite.com
X-ConsoleWeb-APIKey: your-api-key...
```

b. HTTPレスポンス ボディ例

正常時はZIPファイルのため省略し、以下はエラー時のボディです。

※ 実際のJSONには改行やインデントはありません。

```
{
  "status": "error",
  "errorCode": 103,
  "message": "Not found. (Unit not found.)"
}
```

c. ダウンロードZIPファイル 構造

生成されるZIPファイル名は、以下の命名規則で作成されます。

"parts-" + 読取ユニットID + "_" + 処理時刻(yyyyMMddHHmmss) + ".zip"

ZIPファイルは、以下のような内容で生成されます。

フォルダ	フォルダ内容
/Org/	パーツ画像(png拡張子)
/Col/	CSVヘッダー名を入れたtxtファイル(出力カラム名を設定した項目のみ出力)
/File/	ページのファイル名とポジションIDを入れたtxtファイル
/Txt/	読取結果を入れたtxtファイル(ページが読取後のみ出力)

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。

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

CSV出力先カラムが設定されていない場合は、読取ページID + "_" + 読取パーツID となります。

(4) 読取結果CSVのエクスポート

読取結果をCSVファイルでエクスポートします。

URI	/ConsoleWeb/api/v1/reading/units/{id}/export	
メソッド	GET	
認証パラメータ	APIキー (X-ConsoleWeb-ApiKeyヘッダ)	
パラメータ	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
	{id}	○ ダウンロードを行うユニットのID (URIで指定します)
レスポンス	正常の場合は、対象のCSVファイル。 エラーの場合は、エラーコードを含む以下の構造のJSON	
	フィールド	型 説明
	status	文字列 success : 正常 error : エラー
	errorCode	数値 エラー状態を示すID
	message	文字列 メッセージ
エラーコード	HTTPステータスコードと、戻りJSONのerrorCodeにセットされるコードは以下です。	
	HTTP Status	errorCode 意味
	200	0 処理は正常に受付られた
	401	101 認証エラーです。APIキーが誤っているか、APIキーの指定方法が正しくありません。
	404	103 該当のユニットIDは存在しないか、アクセス権限がありません。
	406	105 このユニットIDはまだCSVダウンロードができる状態ではありません。 例えば、ページ追加後に読取がまだ完了していない場合などです。
	500	199 その他の不明なエラーです。リクエストが正しいかを確認いただき、再度発生する場合はお問い合わせください。

a. HTTPリクエスト例

※ 青字の値は実行環境によって異なります。

```
GET /ConsoleWeb/api/v1/reading/units/12345/export HTTP/1.1
Host: company-name.dx-suite.com
X-ConsoleWeb-APIKey: your-api-key...
```

b. HTTPレスポンス ボディ例

正常時はCSVファイルのため省略し、以下はエラー時のボディです。

※ 実際のJSONIには改行やインデントはありません。

```
{
  "status": "error",
  "errorCode": 103,
  "message": "Not found. (Unit not found.)"
}
```

(5) 読取ユニット削除

読取ユニットを削除します。

URI	/ConsoleWeb/api/v1/reading/units/{id}/delete		
メソッド	POST (application/x-www-form-urlencoded)		
認証パラメータ	APIキー (X-ConsoleWeb-ApiKeyヘッダ)		
パラメータ	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
	{id}	○	読取ユニットID
レスポンス	以下の構造のJSON。		
	フィールド	型	説明
	status	文字列	success : 正常 error : エラー
	errorCode	数値	エラー状態を示すID
	message	文字列	メッセージ
	id	数値	削除されたユニットのID
エラーコード	HTTPステータスコードと、戻りJSONのerrorCodeにセットされるコードは以下です。		
	HTTP Status	errorCode	意味
	200	0	処理は正常に受付られた
	401	101	認証エラーです。APIキーが誤っているか、APIキーの指定方法が正しくありません。
	400	119	他の処理により該当ユニットが操作されたため、削除に失敗しました。
	400	120	該当のユニットIDは、Elastic Sorter の仕分けユニットから参照されているため削除できません。
	404	103	該当のユニットIDは存在しないか、アクセス権限がありません。
	404	113	該当の組織IDは存在しないか、アクセス権限がありません。
	500	199	その他の不明なエラーです。リクエストが正しいかを確認いただき、再度発生する場合はお問い合わせください。

a. HTTPリクエスト例

※ 青字の値は実行環境によって異なります。

```
POST /ConsoleWeb/api/v1/reading/units/12345/delete HTTP/1.1
Host: company-name.dx-suite.com
X-ConsoleWeb-ApiKey: your-api-key...
```

b. HTTPレスポンス ボディ例

※ 実際のJSONには改行やインデントはありません。

```
{
  "status" : "success",
  "errorCode" : 0,
  "message" : "",
  "id" : 12345
}
```

(6) 読取ページ追加

読取ユニットに対して、ページ画像を追加します。

読取ユニットとは、1つのCSVファイルを生成する画像の集まりです。

URI	/ConsoleWeb/api/v1/reading/pages/add		
メソッド	POST (multipart/form-data)		
認証パラメータ	APIキー (X-ConsoleWeb-ApiKeyヘッダ)		
パラメータ	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}	※1	追加先のユニットID 登録したいページを追加するユニットIDを指定します。 省略した場合は、新しいユニットが自動生成されます。 指定されるユニットIDは現在のAPIキーからアクセス可能でなければなりません。
	{documentId}	※1	ドキュメントID 登録したいページのドキュメントIDを指定します。
	{unitName}		読取ユニット名 ドキュメントIDを指定している場合、ユニットを作成するユニット名を指定します。 省略時は登録した年月日時分秒で設定されます。 {unitId}を指定した場合は無視されます。
	{file}	○	追加する画像データファイル
	{userId}		ユーザーID ※ リクエスト数照会画面で部署毎に集計したい場合に設定します。
※1 : {unitId}、{documentId} はいずれか必須です。 複数指定した場合は {unitId} が優先されます。			
レスポンス	以下の構造のJSON。		
	フィールド	型	説明
	status	文字列	success : 正常 error : エラー
	errorCode	数値	エラー状態を示すID
	message	文字列	メッセージ
	id	数値配列	追加されたページのIDの配列。 自動採番されます。 PDFファイルなど複数ページのコンテンツをアップロードした場合はページ毎にIDが振られます。
	unitId	数値	追加されたユニットのID リクエスト時にユニットIDを指定している場合は、その

			IDです。省略した場合は、新しく作られてユニットIDです。
エラーコード	HTTPステータスコードと、戻りJSONのerrorCodeにセットされるコードは以下です。		
	HTTP Status	errorCode	意味
	200	0	処理は正常に受付られた
	400	102	ユーザーIDが数字ではない不正な値です。
	400	104	ドキュメントIDが必要です。 ({unitId}が指定されていない場合で、{documentId}も指定されていません。)
	400	112	該当のドキュメントIDは存在しないか、アクセス権限がないか、ユニット名に不正文字があります。
	400	114	画像データファイルが添付されていません。
	400	115	ページの登録に失敗しました。
	400	116	未サポートの画像が添付されています。
	401	101	APIキーが誤っているか、APIキーの有効期限が切れています。 ユーザーIDが誤っているか、ログインできないユーザーIDが指定されています。 ※ ユーザーIDを指定した場合のみチェックされます。
	404	103	該当のユニットIDは存在しないか、アクセス権限がありません。
	500	199	その他の不明なエラーです。リクエストが正しいかを確認いただき、再度発生する場合はお問い合わせ下さい。

a. HTTPリクエスト例

※ 青字の値は実行環境によって異なります。

```
POST /ConsoleWeb/api/v1/reading/pages/add HTTP/1.1
Host: company-name.dx-suite.com
X-ConsoleWeb-ApiKey: your-api-key...
Cache-Control: no-cache
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0g

-----WebKitFormBoundary7MA4YWxkTrZu0g
Content-Disposition: form-data; name="documentId"
12345
-----WebKitFormBoundary7MA4YWxkTrZu0g
Content-Disposition: form-data; name="file"; filename="sample.jpg";
Content-Type: image/jpeg
[message-part-body; type: "image/jpeg", size: 920514 bytes]
-----WebKitFormBoundary7MA4YWxkTrZu0g--
```

b. HTTPレスポンス ボディ例

※ 実際のJSONには改行やインデントはありません。

```
{
  "status" : "success",
  "errorCode" : 0,
  "message" : "ReadingPage added.",
  "id" : [67890],
  "unitId" : 12345
}
```

(7) 読取ページの画像の取得

読取ページの画像をZIPファイルで取得します。

URI	/ConsoleWeb/api/v1/reading/pages/images																
メソッド	GET																
認証パラメータ	APIキー (X-ConsoleWeb-APIKeyヘッダ)																
パラメータ	以下のパラメータを設定します。 <table border="1"> <thead> <tr> <th>パラメータ名</th><th>必須</th><th>パラメータ値</th></tr> </thead> <tbody> <tr> <td>{readingUnitId}</td><td>※1</td><td>ユニットID 1つのユニットのみ指定可能</td></tr> <tr> <td>{readingPageId}</td><td>※1</td><td>ページID 複数をカンマ区切りで指定可能</td></tr> <tr> <td>{filenameType}</td><td></td><td>ZIPファイルに含まれる画像のファイル名 1: ページID 2: アップロード時のファイル名 無指定の場合はページIDとなります。</td></tr> </tbody> </table> ※1 : {readingUnitId}、{readingPageId} はいずれか必須です。 複数指定した場合はパラメータエラーになります。		パラメータ名	必須	パラメータ値	{readingUnitId}	※1	ユニットID 1つのユニットのみ指定可能	{readingPageId}	※1	ページID 複数をカンマ区切りで指定可能	{filenameType}		ZIPファイルに含まれる画像のファイル名 1: ページID 2: アップロード時のファイル名 無指定の場合はページIDとなります。			
パラメータ名	必須	パラメータ値															
{readingUnitId}	※1	ユニットID 1つのユニットのみ指定可能															
{readingPageId}	※1	ページID 複数をカンマ区切りで指定可能															
{filenameType}		ZIPファイルに含まれる画像のファイル名 1: ページID 2: アップロード時のファイル名 無指定の場合はページIDとなります。															
レスポンス	正常の場合は、対象のZIPファイル。 ※ ファイル名にアップロード時のファイル名を指定し、複数ページのPDFをアップロードした場合、PDFのファイル名+(連番)となります。 ※ 画像補正で内部形式が変わった場合、拡張子が変わります。 エラーの場合は、エラーコードを含む以下の構造のJSON。 <table border="1"> <thead> <tr> <th>フィールド</th><th>型</th><th>説明</th></tr> </thead> <tbody> <tr> <td>status</td><td>文字列</td><td>success : 正常 error : エラー</td></tr> <tr> <td>errorCode</td><td>数値</td><td>エラー状態を示すID</td></tr> <tr> <td>message</td><td>文字列</td><td>メッセージ</td></tr> </tbody> </table>		フィールド	型	説明	status	文字列	success : 正常 error : エラー	errorCode	数値	エラー状態を示すID	message	文字列	メッセージ			
フィールド	型	説明															
status	文字列	success : 正常 error : エラー															
errorCode	数値	エラー状態を示すID															
message	文字列	メッセージ															
エラーコード	HTTPステータスコードと、戻りJSONのerrorCodeにセットされるコードは以下です。 <table border="1"> <thead> <tr> <th>HTTP Status</th><th>errorCode</th><th>意味</th></tr> </thead> <tbody> <tr> <td>200</td><td>0</td><td>処理は正常に受付られた</td></tr> <tr> <td>400</td><td>104</td><td>入力パラメータの形式に誤りがあります</td></tr> <tr> <td>401</td><td>101</td><td>認証エラーです。APIキーが誤っているか、APIキーの指定方法が正しくありません。</td></tr> <tr> <td>404</td><td>103</td><td>該当のユニットIDまたはページIDは存在しないか、アクセス</td></tr> </tbody> </table>		HTTP Status	errorCode	意味	200	0	処理は正常に受付られた	400	104	入力パラメータの形式に誤りがあります	401	101	認証エラーです。APIキーが誤っているか、APIキーの指定方法が正しくありません。	404	103	該当のユニットIDまたはページIDは存在しないか、アクセス
HTTP Status	errorCode	意味															
200	0	処理は正常に受付られた															
400	104	入力パラメータの形式に誤りがあります															
401	101	認証エラーです。APIキーが誤っているか、APIキーの指定方法が正しくありません。															
404	103	該当のユニットIDまたはページIDは存在しないか、アクセス															

		権限がありません。
500	199	その他の不明なエラーです。リクエストが正しいかを確認いただき、再度発生する場合はお問い合わせください。

※1 : {readingUnitId}、{readingPageId} いずれも指定した場合は、104エラーになります。戻り値は以下となります。

{ "status": "error", "errorCode": 104, "message": "Bad Request. (Invalid parameter : readingUnitId or readingPageIdのいずれか1つのパラメータを設定してください。)" }

a. HTTPリクエスト例

※ 青字の値は実行環境によって異なります。

```
GET /ConsoleWeb/api/v1/reading/pages/images?readingUnitId=12345 HTTP/1.1
Host: company-name.dx-suite.com
X-ConsoleWeb-APIKey: your-api-key...
```

b. HTTPレスポンス ボディ例

正常時はZIPファイルのため省略し、以下はエラー時のボディです。

※ 実際のJSONには改行やインデントはありません。

```
{
  "status": "error",
  "errorCode": 103,
  "message": "Not found. (Unit not found.)"
}
```

c. ダウンロードZIPファイル 名

生成されるZIPファイル名は、以下の命名規則で作成されます。

・パラメータ:{readingUnitId} を指定した場合、

"pages-" + 読取ユニットID + "_" + 処理時刻(yyyyMMddHHmmss) + ".zip"

・パラメータ{readingPageId} を指定した場合、

"pages--1_" + 処理時刻(yyyyMMddHHmmss) + ".zip"

(8) 読取パーツの情報の取得

読取パーツの情報を取得します。

URI	/ConsoleWeb/api/v1/reading/parts	
メソッド	GET	
認証パラメータ	APIキー (X-ConsoleWeb-ApiKeyヘッダ)	
パラメータ	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
	{readingUnitId}	※1 ユニットID 1つのユニットのみ指定可能
	{readingPageId}	※1 ページID 複数をカンマ区切りで指定可能 (複数のユニットIDにまたがる指定は不可)
	{readingPartsId}	※1 パーツID 複数をカンマ区切りで指定可能 (複数のユニットIDにまたがる指定は不可)
※1 : {readingUnitId}、{readingPageId}、{readingPartsId} 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。		
レスポンス	正常の場合は、以下の構造のJSON。	
	フィールド	型 説明
	status	文字列 success : 正常
	errorCode	数値 正常を示すID(0)
	message	文字列 メッセージ
	readingParts	配列 パーツの配列
	id	数値 パーツのID
	status	数値 パーツのステータス(詳細は後述)
	docsetId	数値 パーツのフォルダID
	documentId	数値 パーツのドキュメントID
	readingUnitId	数値 パーツのユニットID
	readingPageId	数値 パーツのページID
	fileName	文字列 ページのファイル名
	columnNo	数値 CSV出力のカラム番号 (0はCSVに出力しない。 1がA列、2がB列...となる)

	columnName	文字列	CSV出力のヘッダ名
	x	数値	ページ画像に対する座標値(絶対値)
	y	数値	ページ画像に対する座標値(絶対値)
	width	数値	読取パーツの幅
	height	数値	読取パーツの高さ
	result	文字列	読取結果値
	accuracy	数値	読取尤度(現在、項目尤度は提供されないため常に0がセットされる)
	accuracyRank	数値	エンジンが指定した読取値の正確性に関する補助値。 0: 情報なし(デフォルト) 1: 低い精度で正確に読めている 2: 少し低い精度で正確に読めている 3: 少し高い精度で正確に読めている 4: 高い精度で正確に読めている ※エントリー画面のクイックチェックでは、1が赤、4が緑の表示となります。
	createdAt	文字列	パーツの作成日時
パーツのステータスは「 5.1 Intelligent OCR の読み取りステータス 」を参照 エラーの場合は、エラーコードを含む以下の構造のJSON。			
エラーコード	フィールド	型	説明
	status	文字列	error : エラー
	errorCode	数値	エラー状態を示すID
	message	文字列	メッセージ
	HTTPステータスコードと、戻りJSONのerrorCodeにセットされるコードは以下です。		
	HTTP Status	errorCode	意味
	200	0	処理は正常に受付られた
	400	104	入力パラメータの形式に誤りがあります。
	401	101	認証エラーです。APIキーが誤っているか、APIキーの指定方法が正しくありません。
	404	103	指定した{readingUnitId},{readingPageId},{readingPartsId}に紐づく情報が存在しません。

	406	118	複数の読取ユニットIDにまたがる検索条件となっています。
	500	199	その他の不明なエラーです。リクエストが正しいかを確認いただき、再度発生する場合はお問い合わせください。

a. HTTPリクエスト例

※ 青字の値は実行環境によって異なります。

```
GET /ConsoleWeb/api/v1/reading/parts?readingUnitId=12345 HTTP/1.1
Host: company-name.dx-suite.com
X-ConsoleWeb-ApiKey: your-api-key...
```

b. HTTPレスポンス ボディ例

※ 実際のJSONには改行やインデントはありません。

```
{
  "status": "success",
  "errorCode": 0,
  "message": "ReadingParts listing.",
  "readingParts": [
    {
      "id": 12345,
      "status": 16,
      "docsetId": 1455,
      "documentId": 4236,
      "readingUnitId": 28494,
      "readingPageId": 48876,
      "fileName": "sample.jpg",
      "columnNo": 1,
      "columnName": "申し込みNo",
      "x": 100,
      "y": 200,
      "width": 120,
      "height": 30,
      "result": "2019-001",
      "accuracy": 0,
      "accuracyRank": 0,
      "createdAt": "2019-01-01 12:34:56"
    },
    {
      "id": 12346,
      "status": 16,
      "docsetId": 1455,
      "documentId": 4236,
      "readingUnitId": 28494,
      "readingPageId": 48876,
      "fileName": "sample.jpg",
      "columnNo": 1,
      "columnName": "注文品番",

```

```
"x": 100,  
"y": 300,  
"width": 240,  
"height": 30,  
"result": "A234567",  
"accuracy": 0,  
"accuracyRank": 0,  
"createdAt": "2019-01-01 12:34:56"
```

```
}  
]  
}
```

(9) 仕分けユニット追加

仕分けユニットに対して、ページ画像を追加します。

URI	/Sorter/api/v1/add	
メソッド	POST	
認証パラメータ	APIキー (X-ConsoleWeb-ApiKeyヘッダ)	
パラメータ	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
	{sorterRuleId}	○ 仕分けルールID
	{sortingUnitName}	ユニット名 ※ 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る ※ 漢字、ひらがな、カタカナ、英数字、1～128文字で指定してください。
	{file}	○ 仕分けしたいファイル ※ 複数ファイル同時はできる
	{runSortingFlag}	継続して仕分け処理を開始する true: 実施する, false: 実施しない ※ デフォルト値はfalse
	{sendOcrFlag}	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runSortingFlagがtrueでないと作用しません。
	{userId}	ユーザーID ※ リクエスト数照会画面で部署毎に集計したい場合に設定します。 ※ userIdを指定する場合、runSortingFlagとsendOcrFlagの両方が true でないと、読取ページ追加が実施されません。
レスポンス	正常の場合は、下記の構造のJSON。	
	フィールド	型 説明
	status	文字列 success : 正常 error : エラー
	errorCode	数値 エラー状態を示すID
	message	文字列 メッセージ

	sortingUnitId	数値	ユニットID
	エラーの場合は、エラーコードを含む以下の構造のJSON。		
	フィールド	型	説明
	status	文字列	success : 正常 error : エラー
	errorCode	数値	エラー状態を示すID
	message	文字列	メッセージ
エラーコード	sortingUnitId	数値	仕分けユニットID (runSortingFlagまたはsendOcrFlagを指定し、 ユニット登録までは成功している場合に設定され ます)
	HTTPステータスコードと、戻りJSONのerrorCodeにセットされるコードは以下です。		
	HTTP Status	errorCode	意味
	200	0	処理は正常に受付られた
	400	102	ユーザーIDが数字ではない不正な値です。
	400	103	sorterRuleId パラメータが指定されていません。
	400	114	画像データファイルが添付されていません。
	400	116	未サポートの画像が添付されています。
	401	101	ユーザーIDが誤っているか、ログインできないユーザーID が指定されています。 ※ ユーザーIDを指定した場合のみチェックされます。
	403	103	当該のSorter Rule IDは存在しないか、アクセス権限があ りません。
	404	103	当該のSorter Rule IDは存在しないか、アクセス権限があ りません。
	406	106	この仕分けユニットIDはまだ仕分け準備がされていま せん。もしくはOCRへ送信するための準備ができていま せん。 原因として以下が考えられます。 <ul style="list-style-type: none"> ページ画像の登録中にエラーが起こった(画像ファ イルが不正・破損している場合など)(runSortingFlagを 指定している場合) 仕分け処理でエラーが発生した(sendOcrFlagを指定 している場合) 仕分けの結果、ユニット内の全ての帳票が仕分け不可

			となった(sendOcrFlagを指定している場合)
	500	199	その他の不明なエラーです。リクエストが正しいかを確認いただき、再度発生する場合はお問い合わせください。

a. HTTPリクエスト例

※ 青字の値は実行環境によって異なります。

```
POST /Sorter/api/v1/add HTTP/1.1
Host: company-name.dx-suite.com
X-ConsoleWeb-APIKey: your-api-key...
Cache-Control: no-cache
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarybKfB5vzTGsNQMGe

-----WebKitFormBoundarybKfB5vzTGsNQMGe
Content-Disposition: form-data; name="sorterRuleId"
123
-----WebKitFormBoundarybKfB5vzTGsNQMGe
Content-Disposition: form-data; name="file"; filename="sample.jpg";
Content-Type: image/jpeg
[message-part-body; type: "image/jpeg", size: 920514 bytes]
```

b. HTTPレスポンス ボディ例

※ 実際のJSONには改行やインデントはありません。

```
{
  "status":"success",
  "errorCode":0,
  "message":"SortingUnit add success.",
  "sortingUnitId":1350729
}
```

(10) 仕分け処理の実施

仕分けユニットIDを元に仕分けを実施します。

URI	/Sorter/api/v1/sorting																
メソッド	POST																
認証パラメータ	APIキー (X-ConsoleWeb-ApiKeyヘッダ)																
パラメータ	以下のパラメータを設定します。 <table border="1"> <thead> <tr> <th>パラメータ名</th><th>必須</th><th>パラメータ値</th></tr> </thead> <tbody> <tr> <td>{sortingUnitId}</td><td>○</td><td>仕分けユニットID</td></tr> <tr> <td>{sendOcrFlag}</td><td></td><td> 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 ※ 未設定の場合は false が設定されます </td></tr> <tr> <td>{userId}</td><td></td><td> ユーザーID ※ リクエスト数照会画面で部署毎に集計したい場合に設定します。 ※ userIdを指定する場合、sendOcrFlagがtrueでないと、読取ページ追加が実施されません。 </td></tr> </tbody> </table>		パラメータ名	必須	パラメータ値	{sortingUnitId}	○	仕分けユニットID	{sendOcrFlag}		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 ※ 未設定の場合は false が設定されます	{userId}		ユーザーID ※ リクエスト数照会画面で部署毎に集計したい場合に設定します。 ※ userIdを指定する場合、sendOcrFlagがtrueでないと、読取ページ追加が実施されません。			
パラメータ名	必須	パラメータ値															
{sortingUnitId}	○	仕分けユニットID															
{sendOcrFlag}		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 ※ 未設定の場合は false が設定されます															
{userId}		ユーザーID ※ リクエスト数照会画面で部署毎に集計したい場合に設定します。 ※ userIdを指定する場合、sendOcrFlagがtrueでないと、読取ページ追加が実施されません。															
レスポンス	下記の構造のJSON。 <table border="1"> <thead> <tr> <th>フィールド</th><th>型</th><th>説明</th></tr> </thead> <tbody> <tr> <td>status</td><td>文字列</td><td>success : 正常 error : エラー</td></tr> <tr> <td>errorCode</td><td>数値</td><td>エラー状態を示すID</td></tr> <tr> <td>message</td><td>文字列</td><td>メッセージ</td></tr> </tbody> </table>		フィールド	型	説明	status	文字列	success : 正常 error : エラー	errorCode	数値	エラー状態を示すID	message	文字列	メッセージ			
フィールド	型	説明															
status	文字列	success : 正常 error : エラー															
errorCode	数値	エラー状態を示すID															
message	文字列	メッセージ															
エラーコード	HTTPステータスコードと、戻りJSONのerrorCodeにセットされるコードは以下です。 <table border="1"> <thead> <tr> <th>HTTP Status</th><th>errorCode</th><th>意味</th></tr> </thead> <tbody> <tr> <td>200</td><td>0</td><td>処理は正常に受付られた</td></tr> <tr> <td>400</td><td>102</td><td>ユーザーIDが数字ではない不正な値です。</td></tr> <tr> <td>400</td><td>103</td><td>リクエストパラメータが不正です。</td></tr> <tr> <td>401</td><td>101</td><td>APIキーが誤っているか、APIキーの有効期限が切れています。 ユーザーIDが誤っているか、ログインできないユーザーID</td></tr> </tbody> </table>		HTTP Status	errorCode	意味	200	0	処理は正常に受付られた	400	102	ユーザーIDが数字ではない不正な値です。	400	103	リクエストパラメータが不正です。	401	101	APIキーが誤っているか、APIキーの有効期限が切れています。 ユーザーIDが誤っているか、ログインできないユーザーID
HTTP Status	errorCode	意味															
200	0	処理は正常に受付られた															
400	102	ユーザーIDが数字ではない不正な値です。															
400	103	リクエストパラメータが不正です。															
401	101	APIキーが誤っているか、APIキーの有効期限が切れています。 ユーザーIDが誤っているか、ログインできないユーザーID															

			が指定されています。 ※ ユーザーIDを指定した場合のみチェックされます。
	404	103	当該仕分けユニットIDは存在しないか、アクセス権限がありません。
	406	106	この仕分けユニットIDはまだ仕分け準備がされていません。もしくはOCRへ送信するための準備ができていません。
	500	199	その他の不明なエラーです。リクエストが正しいかを確認いただき、再度発生する場合はお問い合わせください。

a. HTTPリクエスト例

※ 青字の値は実行環境によって異なります。

```
POST /Sorter/api/v1/sorting HTTP/1.1
Host: company-name.dx-suite.com
X-ConsoleWeb-ApiKey: your-api-key...
Content-Type: application/x-www-form-urlencoded
Content-Length: 17

sortingUnitId=456
```

b. HTTPレスポンス ボディ例

※ 実際のJSONには改行やインデントはありません。

```
{ "status": "success",
  "errorCode": 0,
  "message": "Starting sorting. [sorting_unit_id: 456]"
}
```

(11) 仕分け結果をOCRに送信

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/api/v1/send																						
メソッド	POST																						
認証パラメータ	APIキー (X-ConsoleWeb-APIKeyヘッダ)																						
パラメータ	以下のパラメータを設定します。 <table border="1"> <thead> <tr> <th>パラメータ名</th><th>必須</th><th>パラメータ値</th></tr> </thead> <tbody> <tr> <td>{sortingUnitId}</td><td>○</td><td>仕分けユニットID</td></tr> <tr> <td>{userId}</td><td></td><td> ユーザーID ※ リクエスト数照会画面で部署毎に集計したい場合に設定します。 </td></tr> </tbody> </table>		パラメータ名	必須	パラメータ値	{sortingUnitId}	○	仕分けユニットID	{userId}		ユーザーID ※ リクエスト数照会画面で部署毎に集計したい場合に設定します。												
パラメータ名	必須	パラメータ値																					
{sortingUnitId}	○	仕分けユニットID																					
{userId}		ユーザーID ※ リクエスト数照会画面で部署毎に集計したい場合に設定します。																					
レスポンス	下記の構造のJSON。 <table border="1"> <thead> <tr> <th>フィールド</th><th>型</th><th>説明</th></tr> </thead> <tbody> <tr> <td>status</td><td>文字列</td><td>success : 正常 error : エラー</td></tr> <tr> <td>errorCode</td><td>数値</td><td>エラー状態を示すID</td></tr> <tr> <td>message</td><td>文字列</td><td>メッセージ</td></tr> </tbody> </table>		フィールド	型	説明	status	文字列	success : 正常 error : エラー	errorCode	数値	エラー状態を示すID	message	文字列	メッセージ									
フィールド	型	説明																					
status	文字列	success : 正常 error : エラー																					
errorCode	数値	エラー状態を示すID																					
message	文字列	メッセージ																					
エラーコード	HTTPステータスコードと、戻りJSONのerrorCodeにセットされるコードは以下です。 <table border="1"> <thead> <tr> <th>HTTP Status</th><th>errorCode</th><th>意味</th></tr> </thead> <tbody> <tr> <td>200</td><td>0</td><td>処理は正常に受付られた</td></tr> <tr> <td>400</td><td>102</td><td>ユーザーIDが数字ではない不正な値です。</td></tr> <tr> <td>400</td><td>103</td><td>リクエストパラメータが不正です。</td></tr> <tr> <td>401</td><td>101</td><td> APIキーが誤っているか、APIキーの有効期限が切れています。 ユーザーIDが誤っているか、ログインできないユーザーIDが指定されています。 ※ ユーザーIDを指定した場合のみチェックされます。 </td></tr> <tr> <td>404</td><td>103</td><td>当該仕分けユニットIDは存在しないか、アクセス権限がありません。</td></tr> <tr> <td>406</td><td>106</td><td>この仕分けユニットIDはまだOCRへ送信するための準備ができていません。</td></tr> </tbody> </table>		HTTP Status	errorCode	意味	200	0	処理は正常に受付られた	400	102	ユーザーIDが数字ではない不正な値です。	400	103	リクエストパラメータが不正です。	401	101	APIキーが誤っているか、APIキーの有効期限が切れています。 ユーザーIDが誤っているか、ログインできないユーザーIDが指定されています。 ※ ユーザーIDを指定した場合のみチェックされます。	404	103	当該仕分けユニットIDは存在しないか、アクセス権限がありません。	406	106	この仕分けユニットIDはまだOCRへ送信するための準備ができていません。
HTTP Status	errorCode	意味																					
200	0	処理は正常に受付られた																					
400	102	ユーザーIDが数字ではない不正な値です。																					
400	103	リクエストパラメータが不正です。																					
401	101	APIキーが誤っているか、APIキーの有効期限が切れています。 ユーザーIDが誤っているか、ログインできないユーザーIDが指定されています。 ※ ユーザーIDを指定した場合のみチェックされます。																					
404	103	当該仕分けユニットIDは存在しないか、アクセス権限がありません。																					
406	106	この仕分けユニットIDはまだOCRへ送信するための準備ができていません。																					

	500	199	その他の不明なエラーです。リクエストが正しいかを確認いただき、再度発生する場合はお問い合わせください。
--	-----	-----	---

a. HTTPリクエスト例

※ 青字の値は実行環境によって異なります。

```
POST /Sorter/api/v1/send HTTP/1.1
Host: company-name.dx-suite.com
X-ConsoleWeb-ApiKey: your-api-key...
Content-Type: application/x-www-form-urlencoded
Content-Length: 17

sortingUnitId=456
```

b. HTTPレスポンス ボディ例

※ 実際のJSONには改行やインデントはありません。

```
{
  "status": "success",
  "errorCode": 0,
  "message": "Starting sending. [sorting_unit_id: 456]"
}
```

(12) 仕分け結果 CSV エクスポート

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/api/v1/csvDownload	
メソッド	POST	
認証パラメータ	APIキー (X-ConsoleWeb-APIKeyヘッダ)	
パラメータ	以下のパラメータを設定します。	
	パラメータ名	必須
	{sortingUnitId}	○
		仕分けユニットID
レスポンス	正常時は、対象のCSVファイル。 エラー時は、下記の構造のJSON。	
	フィールド	型
	status	文字列
	errorCode	数値
	message	文字列
エラーコード	HTTPステータスコードと、戻りJSONのerrorCodeにセットされるコードは以下です。	
	HTTP Status	errorCode
	200	0
	400	199
	401	101
	404	103
	406	106
	500	199

a. HTTPリクエスト例

※ 青字の値は実行環境によって異なります。

```
POST /Sorter/api/v1/csvDownload HTTP/1.1
Host: company-name.dx-suite.com
X-ConsoleWeb-ApiKey: your-api-key...
Content-Type: application/x-www-form-urlencoded
Content-Length: 17

sortingUnitId=456
```

b. HTTPレスポンス ボディ例

正常時はCSVファイルのため省略し、以下はエラー時のボディです。

※ 実際のJSONには改行やインデントはありません。

```
{
  "status": "error",
  "errorCode": 101,
  "message": "Authentication error."
}
```


(13) 仕分けユニット検索

仕分けユニットのステータスを返却します。

URI	/Sorter/api/v1/status		
メソッド	POST		
認証パラメータ	APIキー (X-ConsoleWeb-ApiKeyヘッダ)		
パラメータ	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
	{sortingUnitId}	○	仕分けユニットID
レスポンス	正常の場合は、下記の構造のJSON。		
	フィールド	型	説明
	status	文字列	success : 正常 error : エラー
	statusCode	数値	仕分けステータスを表すコード値 (詳細は後述)
	statusName	文字列	仕分けステータス名称
	errorCode	数値	エラー状態を示すID
	message	文字列	メッセージ
	pageCountAll	数値	仕分け対象の全頁数
	statusList	配列	仕分け先別のOCR処理一覧
	readingUnitId	数値	仕分け先の読み取りユニットID ※ 仕分けされたページが無い、 OCRに送信していない仕分け先は 0 が設定されます。
	sendCountSuccess	数値	OCR送信済み件数
	sendCountFailure	数値	OCR送信エラー件数
	sendCountAll	数値	OCR送信対象件数
	statusCode	数値	仕分けステータスを表すコード値 (詳細は後述)
	statusName	文字列	ステータス名称
	仕分けステータスのコード表は「 5-2 Elastic Sorter の仕分けステータス 」を参照		

	エラーの場合は、下記の構造のJSON。		
	フィールド	型	説明
	status	文字列	error : エラー
	errorCode	数値	エラー状態を示すID
	message	文字列	メッセージ
エラーコード	HTTPステータスコードと、戻りJSONのerrorCodeにセットされるコードは以下です。		
	HTTP Status	errorCode	意味
	200	0	処理は正常に受付られた
	401	101	認証エラーです。APIキーが誤っているか、APIキーの指定方法が正しくありません。
	404	103	該当の仕分けユニットIDは存在しないか、アクセス権限がありません。
	500	199	その他の不明なエラーです。リクエストが正しいかを確認いただき、再度発生する場合はお問い合わせください。

a. HTTPリクエスト例

※ 青字の値は実行環境によって異なります。

```
POST /Sorter/api/v1/status HTTP/1.1
Host: company-name.dx-suite.com
X-ConsoleWeb-ApiKey: your-api-key...
Content-Type: application/x-www-form-urlencoded
Content-Length: 17

sortingUnitId=456
```

b. HTTPレスポンス ボディ例

正常時はCSVファイルのため省略し、以下はエラー時のボディです。

※ 実際のJSONには改行やインデントはありません。

```
{
  "status": "success",
  "statusCode": 150,
  "statusName": "CSV出力完了",
  "errorCode": 0,
  "message": "",
  "pageCountAll": 2,
  "statusList": [{
    "readingUnitId": 0,
    "sendCountSuccess": 0,
```

```
"sendCountFailues": 0,  
"sendCountAll": 0,  
"statusCode": 0,  
"statusName": ""  
},  
{  
  "readingUnitId": 1,  
  "sendCountSuccess": 2,  
  "sendCountFailues": 0,  
  "sendCountAll": 2,  
  "statusCode": 130,  
  "statusName": "OCRベリファイ完了"  
}  
}]  
}
```

(14) 仕分けユニット削除

仕分けユニットのステータスを返却します。

URI	/Sorter/api/v1/delete		
メソッド	POST		
認証パラメータ	APIキー (X-ConsoleWeb-ApiKeyヘッダ)		
パラメータ	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
	{sortingUnitId}	○	仕分けユニットID
レスポンス	下記の構造のJSON。		
	フィールド	型	説明
	status	文字列	success : 正常 error : エラー
	errorCode	数値	エラー状態を示すID
	message	文字列	メッセージ
エラーコード	HTTPステータスコードと、戻りJSONのerrorCodeにセットされるコードは以下です。		
	HTTP Status	errorCode	意味
	200	0	処理は正常に受付られた
	401	101	認証エラーです。APIキーが誤っているか、APIキーの指定方法が正しくありません。
	404	103	該当の仕分けユニットIDは存在しないか、アクセス権限がありません。
	500	199	その他の不明なエラーです。リクエストが正しいかを確認いただき、再度発生する場合はお問い合わせください。

a. HTTPリクエスト例

※ 青字の値は実行環境によって異なります。

```
POST /Sorter/api/v1/delete HTTP/1.1
Host: company-name.dx-suite.com
X-ConsoleWeb-ApiKey: your-api-key...
Content-Type: application/x-www-form-urlencoded
Content-Length: 17

sortingUnitId=456
```

b. HTTPレスポンス ボディ例

※ 実際のJSONには改行やインデントはありません。

```
{
  "status" : "success",
  "errorCode" : 0,
  "message" : ""
}
```

5. 各アプリケーションのステータス仕様

以下に各アプリケーションで利用しているステータスを掲載します。

(1) Intelligent OCR の読み取りステータス

読み取り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
1	ユニットのみ登録された状態
2	ページの追加処理中
3	ページの追加完了
121	自動画像補正中
122	自動画像補正エラー
123	自動画像補正完了
6	自動画像補正・パーツ生成 前処理
7	パーツ生成エラー
8	パーツ生成中
9	パーツ生成完了
10	NX読取エラー
11	NX読取中(定型の場合)
111	非定型読取中(Multiformの場合)
12	NX読取完了(定型の場合)
112	非定型読取完了(Multiformの場合)
13	エントリー中
14	エントリーエラー
15	エントリー完了(仮入力)
115	ベリファイ中
16	ベリファイ完了(承認済)
17	加工処理エラー
18	加工処理中
19	加工処理完了

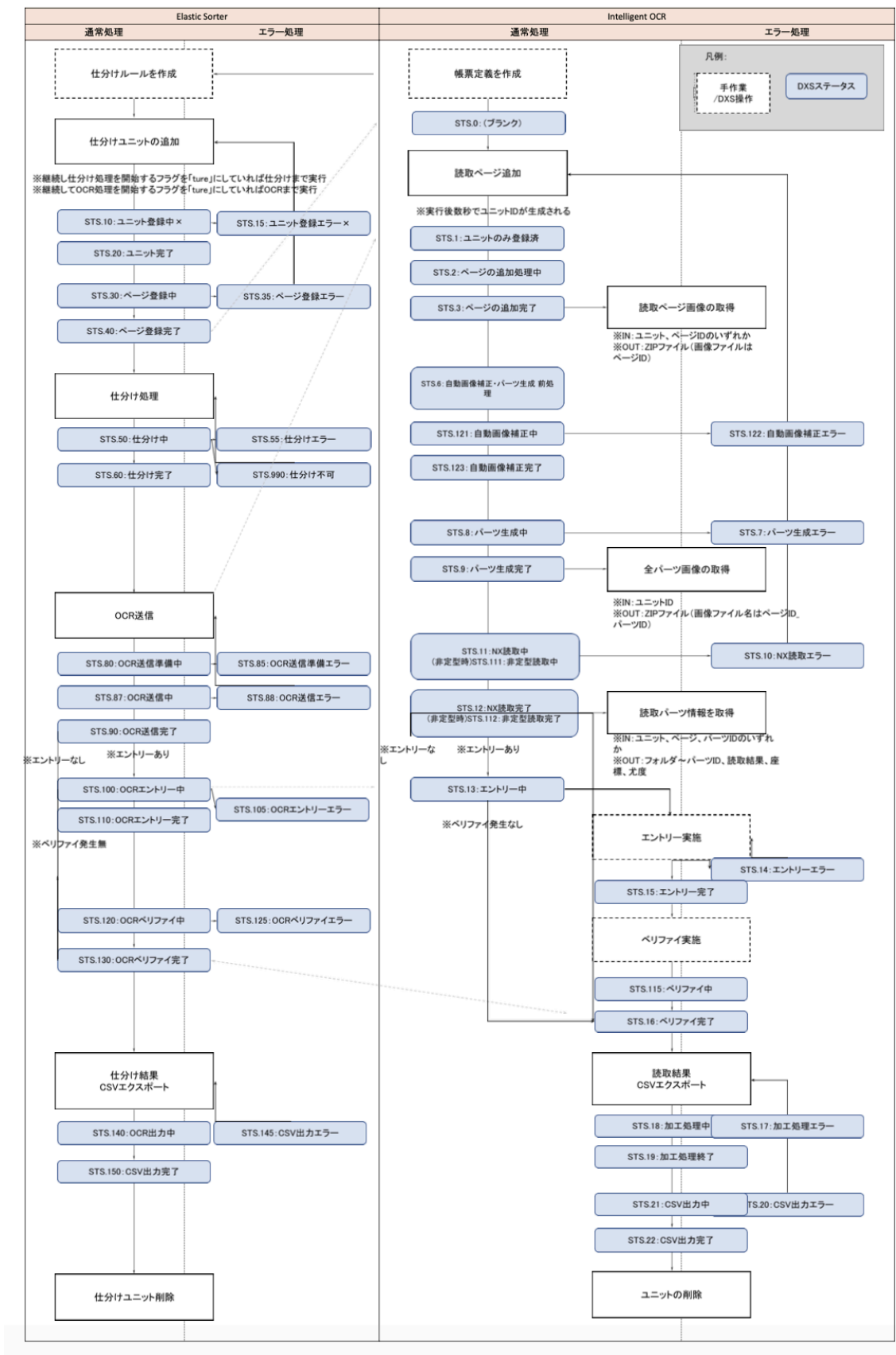
20	CSV出力エラー
21	CSV出力中
22	CSV出力完了

(2) 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
20	ユニット完了
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
65	仕分け一部確定
70	仕分け確定
990	仕分け不可
80	OCR送信準備中
85	OCR送信準備エラー
87	OCR送信中
88	OCR送信エラー
90	OCR送信完了
100	OCRエントリー中
110	OCRエントリー完了
120	OCRベリファイ中
130	OCRベリファイ完了
140	CSV出力中
145	CSV出力エラー
150	CSV出力完了

(3) ステータス別ワークフロー図



6. サンプルコード

各APIを呼び出すcurlコマンドのサンプルを掲載します。環境に合わせて、青字部分を変更して実行ください。

※ curl(カール)とは、ファイルを送受信するコマンドラインツールです。[公式ウェブサイト](#)からダウンロードし利用することができます。

(1) ドキュメント検索

フォルダID「123」を指定した場合

```
curl -X GET -H "X-ConsoleWeb-APIKey: your-api-key..."  
"https://company-name.dx-suite.com/ConsoleWeb/api/v1/documents?docsetId=123  
&documentName=MyTest"
```

※ documentNameに日本語を設定する場合は、URLエンコードしてから設定します。
(日本語等の2バイトの言語(マルチバイト言語)をURL文字列に変換します。)
例)「テスト」の場合→「%e3%83%86%e3%82%b9%e3%83%88」

(2) 読取ユニット検索

ドキュメントID「123」を指定した場合

```
curl -X GET -H "X-ConsoleWeb-APIKey: your-api-key..."  
https://company-name.dx-suite.com/ConsoleWeb/api/v1/reading/units?documentId=123
```

ドキュメントID「123」と作成日の範囲「2019-01-01 10:00 ~ 14:00」を指定した場合

```
curl -X GET -H "X-ConsoleWeb-APIKey: your-api-key..."  
"https://company-name.dx-suite.com/ConsoleWeb/api/v1/reading/units?documentId=123&createdFrom=2019-01-01+10%3A00%3A00&createdTo=2019-01-01+14%3A00%3A00"
```

※ createdFrom / createdToに日時を設定する場合は、URLエンコードしてから設定します。
例)「2019-01-01 10:00:00」の場合→「2019-01-01+10%3A00%3A00」

(3) 読取ユニット内の全パーツ画像取得

ユニットID「12345」を指定した場合(デフォルトのファイル名で取得 [※参照](#))

```
curl -X GET -H "X-ConsoleWeb-APIKey: your-api-key..."  
https://company-name.dx-suite.com/ConsoleWeb/api/v1/reading/units/12345/partsDownload -J -O
```

ユニットID「12345」を指定し、任意のダウンロードファイル名を指定する場合

```
curl -X GET -H "X-ConsoleWeb-APIKey: your-api-key..."  
https://company-name.dx-suite.com/ConsoleWeb/api/v1/reading/units/12345/partsDownload -o  
your-filename.zip
```

(4) 読取結果CSVのエクスポート

ユニットID「12345」を指定した場合

```
curl -X GET -H "X-ConsoleWeb-APIKey: your-api-key..."  
https://company-name.dx-suite.com/ConsoleWeb/api/v1/reading/units/12345/export
```

(5) 読取ユニット削除

ユニットID「12345」を指定した場合

```
curl -X POST -H "X-ConsoleWeb-APIKey: your-api-key..."  
https://company-name.dx-suite.com/ConsoleWeb/api/v1/reading/units/12345/delete
```

(6) 読取ページ追加

新しいユニットとして追加する場合

```
curl -X POST -H "X-ConsoleWeb-APIKey: your-api-key..." -F "documentId=123" -F "file=@test.pdf"  
https://company-name.dx-suite.com/ConsoleWeb/api/v1/reading/pages/add
```

既存ユニットID「12345」に追加する場合

※ "documentId"パラメータの代わりに、"unitId" パラメータを指定します。

```
curl -X POST -H "X-ConsoleWeb-APIKey: your-api-key..." -F "unitId=12345" -F "file=@test.pdf"  
https://company-name.dx-suite.com/ConsoleWeb/api/v1/reading/pages/add
```

(7) 読取ページの画像の取得

ユニットID「12345」を指定した場合（デフォルトのファイル名で取得 [※参照](#)）

```
curl -X GET -H "X-ConsoleWeb-APIKey: your-api-key..."  
https://company-name.dx-suite.com/ConsoleWeb/api/v1/reading/pages/images?readingUnitId=12345  
-J -O
```

ユニットID「12345」を指定し、任意のダウンロードファイル名を指定する場合

```
curl -X GET -H "X-ConsoleWeb-APIKey: your-api-key..."  
https://company-name.dx-suite.com/ConsoleWeb/api/v1/reading/pages/images?readingUnitId=12345  
-o your-filename.zip
```

(8) 読取パーツの情報の取得

ユニットID「12345」を指定した場合

```
curl -X GET -H "X-ConsoleWeb-APIKey: your-api-key..."  
https://company-name.dx-suite.com/ConsoleWeb/api/v1/reading/parts?readingUnitId=12345
```

(9) 仕分けユニット追加

仕分けルールID「123」を指定した場合

```
curl -X POST -H "X-ConsoleWeb-APIKey: your-api-key..." -F "file=@test.pdf" -F "sorterRuleId=123"  
https://company-name.dx-suite.com/Sorter/api/v1/add
```

(10) 仕分け処理の実施

仕分けユニットID「456」を指定した場合

```
curl -X POST -H "X-ConsoleWeb-APIKey: your-api-key..." -d "sortingUnitId=456"  
https://company-name.dx-suite.com/Sorter/api/v1/sorting
```

(11) 仕分け結果をOCRに送信

仕分けユニットID「456」を指定した場合

```
curl -X POST -H "X-ConsoleWeb-APIKey: your-api-key..." -d "sortingUnitId=456"  
https://company-name.dx-suite.com/Sorter/api/v1/send
```

(12) 仕分け結果 CSV エクスポート

仕分けユニットID「456」を指定した場合

```
curl -X POST -H "X-ConsoleWeb-APIKey: your-api-key..." -d "sortingUnitId=456"  
https://company-name.dx-suite.com/Sorter/api/v1/csvDownload
```

(13) 仕分けユニット検索

仕分けユニットID「456」を指定した場合

```
curl -X POST -H "X-ConsoleWeb-APIKey: your-api-key..." -d "sortingUnitId=456"  
https://company-name.dx-suite.com/Sorter/api/v1/status
```

(14) 仕分けユニット削除

ユニットID「456」を指定した場合

```
curl -X POST -H "X-ConsoleWeb-APIKey: your-api-key..." -d "sortingUnitId=456"  
https://company-name.dx-suite.com/Sorter/api/v1/delete
```

以上



〈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