

# 製本直送.com API 仕様書

シーズネット株式会社

## 改版履歴

Ver	日付	章/頁	内容
1.0	2015/05/15		初版
1.01	2015/06/17		API Type4 の結果レスポンスに説明を追記
1.02	2015/09/11	3 頁	“FTP サーバーのご準備”の説明文を追記
1.03	2015/09/11	4 頁	発送完了通知についての説明を追記
1.04	2016/01/21	14 頁	API Type4 のリクエストパラメータの名称を一部変更 ([color_type]→[text_color_type])
1.05	2016/02/02	8 頁 , 15 頁	API Type1,3,4 で、速達パラメータに、「特急便」、「急行便」追加
1.1	2016/02/20		API Type6 を追加
1.1.1	2016/05/11	37 頁	エラーコードを追加
1.2.1	2016/05/23	4 頁	テスト環境の URL 変更
1.3.1	2016/06/08		誤記を修正
1.3.2	2016/09/12	36 頁	「表紙の種類・加工」の表の誤記を修正
1.3.3	2016/12/20		誤表記を修正

# 1. 概要

本仕様書は、製本直送.com(<http://www.seichoku.com>)の API の仕様に関するものです。  
製本直送.com の API は、大きく以下の4つのものがあります。

機能名	API 名	概要
注文要求 API(承認機能なし)	Type1	製本直送.com に登録済みの原稿に対して、注文を出す API となります。
注文要求 API(承認機能あり)	Type3	Type1 と基本的に同様の機能ですが、出された注文が管理画面側で承認待ちとなり、承認が必要となります。
原稿登録&注文要求 API	Type4	原稿用の PDF の登録から注文までをワンストップで行う API となります。
原稿リスト取得 API	Type5	製本直送.com に登録済みの原稿を一覧で取得する API となります。
原稿登録 API	Type6	製本直送.com に原稿を登録する API となります。

## 2. 利用制限および注意事項

- ・サポートするファイル形式やファイルサイズなどは、製本直送サービスに準拠します。
- ・各 API のリクエストパラメータは json エンコードされたものを利用します。
- ・json リクエストは指定された文字コードに従って URL エンコードする必要があります。
  1. データを json\_encode
  2. 1でエンコードしたデータを base64\_encode
- ・API 利用時の文字コードは UTF-8 とします。
- ・各 API には、テストモードをご用意しております。API の開発の際はテストモードを利用し、開発完了後に本番モードに切り替えていただけますようお願い致します。

## 3. 事前準備

### (1) API 利用申請

APIをご利用いただくには、製本直送.com への申請が必要となります。申請後、製本直送.com 運営事務局より、利用者固有のアクセスキーとシークレットキーを発行いたします。

No	項目名	概要
1	アクセスキー	API 利用者を特定する固有番号
2	シークレットキー	アクセスキーに対する通信時の認証用キー

### (2) 会員 ID の確認

製本直送.com で登録したアカウントの会員 ID を確認してください。  
会員 ID は、製本直送.com のマイページにログイン後、「会員情報」からご確認いただけます。  
会員 ID は、マイページに記載の「お客様 ID」から、“SCK”を除く【数字部分】のみとなります。

(例)お客様 ID=SCK123456 の場合、API で利用する会員 ID は、123456 となります。

※テストモードをご利用の際、製本直送.com のテスト環境にも会員登録が必要となります。  
以下より会員登録をお願い致します。

#### ■ 製本直送.com テスト環境

<http://stg.seichoku.com/>

※テスト環境へのアクセスには IP 制限をつけております。  
ご利用前に、ご利用の IP アドレスを製本直送運営事務局までご連絡下さい。

### (3) Type4, Type6 をご利用の場合、FTP サーバーのご準備

Type4, Type6 をご利用いただく場合、FTP サーバーをご利用者様ご自身にご準備いただく必要がございます。  
FTP サーバーに製本する PDF をアップロードしていただきますと、API Type4 がその FTP サーバーから PDF ファイルを取得し、製本用原稿を作成し、製本します。

FTP サーバーをご準備いただき、

- ・FTP ホスト
- ・FTP ユーザー名
- ・FTP パスワード
- ・FTP ポート
- ・FTPS の利用の有無
- ・結果レスポンスの受け取り URL

の各情報を、製本直送運営事務局までご連絡下さい。

※・セキュリティ上、FTP サーバーにベーシック認証をつけられる場合は、ベーシック認証のユーザー名、パ

スワードも併せてご連絡ください。

(4) 発送完了通知について(任意)

製本の発送が完了しますと、発送完了通知を POST 送信します(仕様は下に記載)。  
発送完了通知を受信されたい方は、以下の情報を製本直送運営事務局までご連絡下さい。

・発送完了通知の受け取り URL

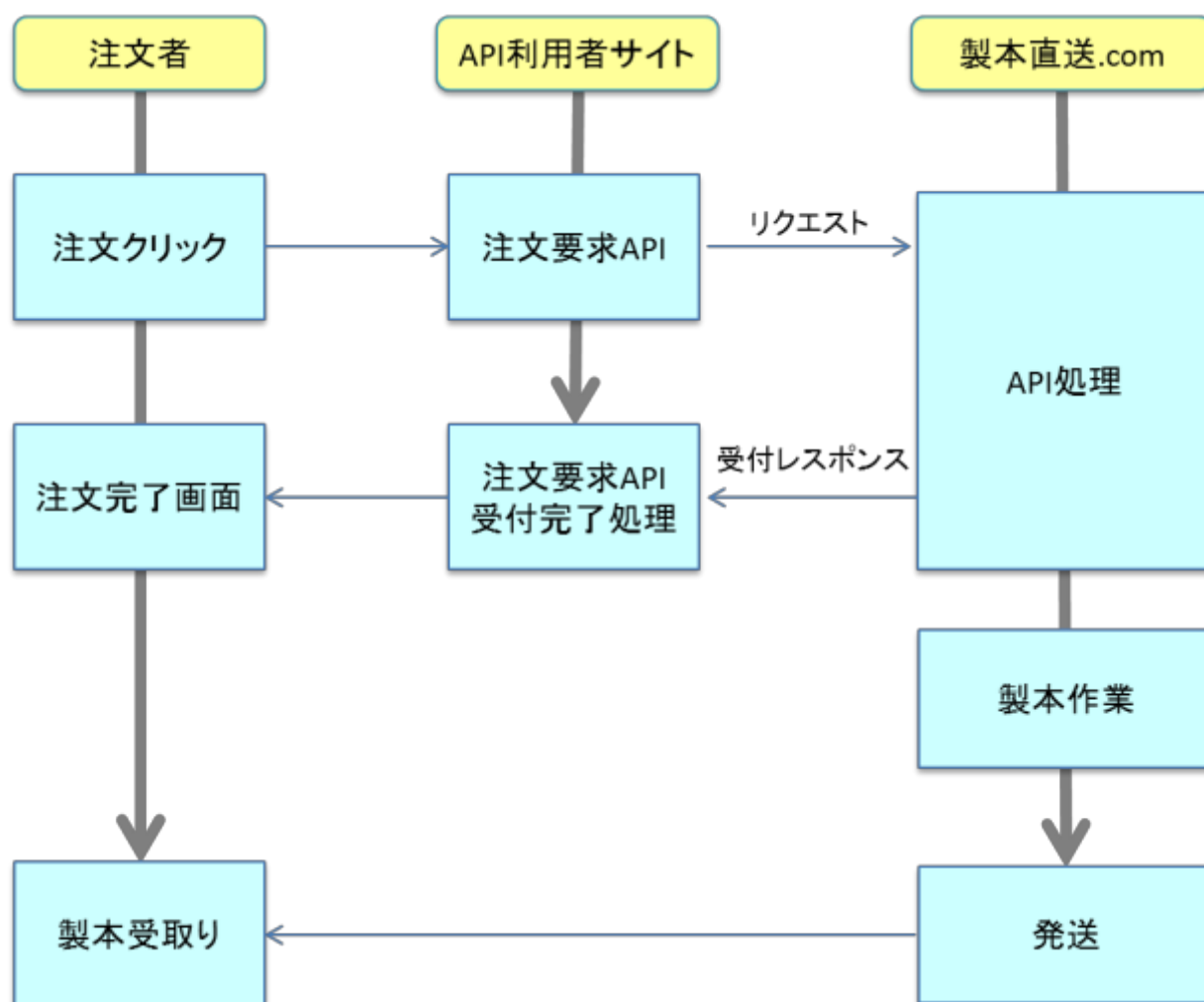
発送完了通知 レスポンスパラメータ

No	論理名	物理名	備考
1	発送完了した注文番号	order_id	
2	発送方法	delivery_code	01:ヤマトメール便 02:ビジネスレター便 03:宅急便
3	お問合せ番号	delivery_number	
4	ステータス	status	常に 4 が返ります

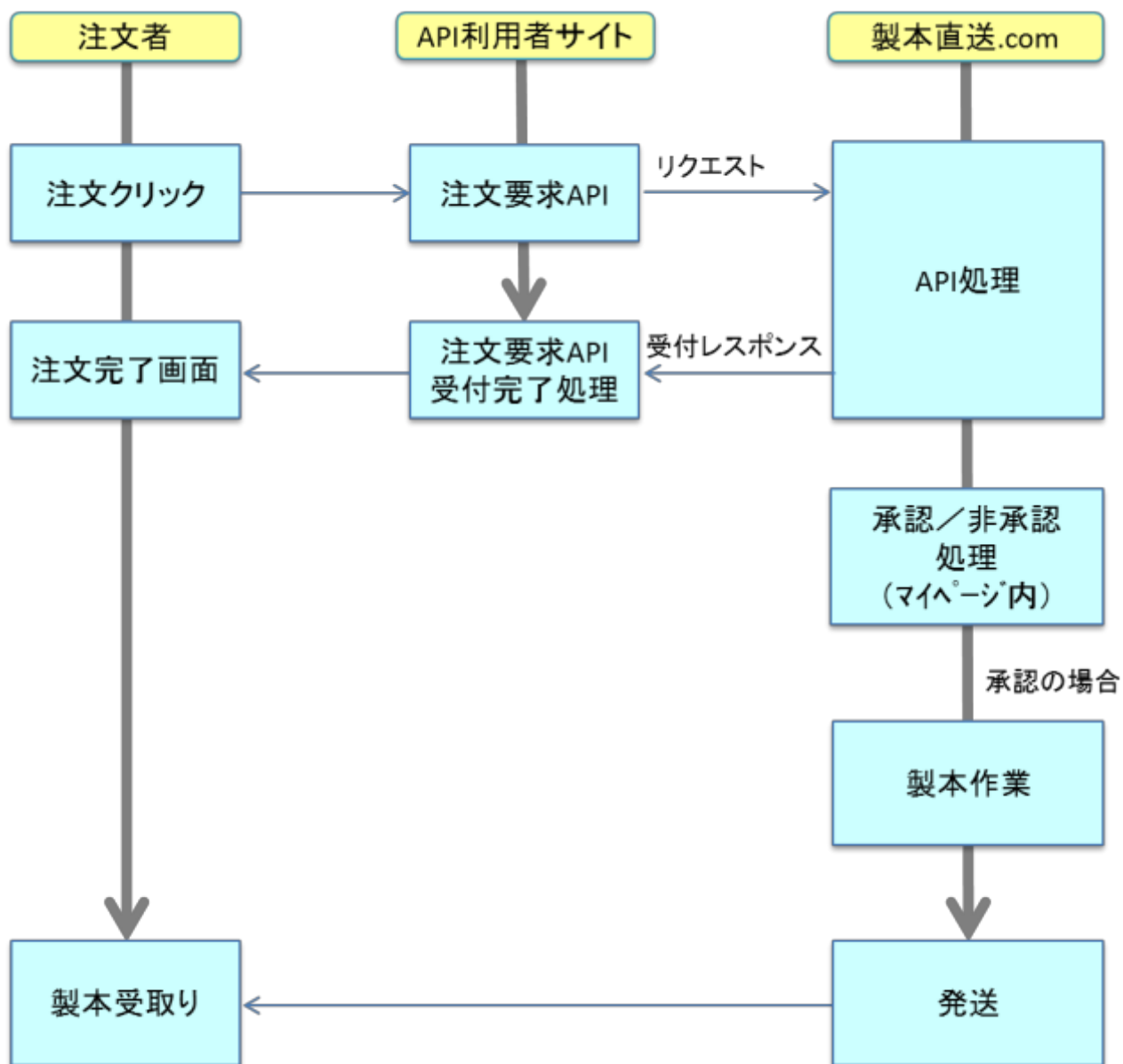
## 4. API Type1, Type3（注文要求 API）

機能名	API 名	概要
注文要求 API(承認機能なし)	Type1	製本直送.com に登録済みの原稿に対して、注文を出す API となります。
注文要求 API(承認機能あり)	Type3	Type1 と基本的に同様の機能ですが、出された注文が管理画面側で承認待ちとなり、承認が必要となります。

### Type1(承認機能なし)の処理の流れ



## Type3(承認機能あり)の処理の流れ



## API: Order

項目名	概要
概要	発注者から製本直送.com への注文要求
送信先 URL	https://www.seichoku.com/api/order.php (※テストモードでも送信先 URL は上記となります)
文字コード	UTF-8
POST パラメータ名	param

## リクエストパラメータ(Type1,3)

No	項目名		型	必須	複数	備考
	論理名	物理名				
1	モード	testmode	半角数字 固定 1 桁	○		0→本番 1→テストモード ※テスト時はテストサーバーにデータが転送されます。
2	API アクセスキー	access_key	半角英数字 一部記号	○		製本直送.com のマイページに記載のものを記入 ※テスト時はテストサーバーのものを記入
3	API シークレットキー	secret_key	半角英数字 一部記号	○		ご利用開始時にメールでご連絡したものを記入 ※テスト時はテストサーバーのものを記入
4	会員 ID	customer_id	半角数字	○		製本直送.com のマイページに記載のものを記入(SCKを除く数字のみ) ※テスト時はテストサーバーのものを記入
5	API タイプ	type	半角数字 固定 1 桁	○		1→Type1 3→Type=3
6	承認フラグ	approve	半角数字	△		type=3 の際必須



				固定 1 桁		0→注文未承認状態で受注
7	注文日時	order_date		日時 固定 19 桁	○	yyyy-mm-dd hh:mm:ss
8	注文番号	order_id		文字列		販売店様側の任意の管理番号。
9	速達フラグ	express		半角数字 固定 1 桁	○	0→普通 1→速達 2→急行便 3→特急便
10	送付先登録フラグ	cre_addr		半角数字 固定 1 桁	○	0→送付先を登録しない 1→送付先を登録する
11	登録済み送付先 ID	pre_regist_addr		半角数字	○	事前に登録済みの送付先に送付する場合は、送付先 ID を記入。送付先 ID を使用しない場合は 0。 (※送付先 ID は、【マイページ】-【登録済み郵送先】で確認)。API で取得できるよう改良予定。
12	送付先情報	addr		Array		登録済み送付先 ID 設定時必須
13	郵便番号1		zip01	半角数字 固定 3 桁	△	
14	郵便番号2		zip02	半角数字 固定 4 桁	△	
15	都道府県番号		pref	半角数字 固定 2 桁	△	都道府県番号参照
16	送付先住所1		addr01	文字列	△	
17	送付先住所2		addr02	文字列	△	
18	送付先電話番号		tel	半角数字 一部記号	△	記号は - のみ
19	受取人名		name	文字列	△	

20	受取人連絡用 email		email	文字列			
21	注文詳細	detail		Array	○	○	
22	原稿 ID		draft_id	半角数字	○		
23	購入部数		quantity	半角数字	○		

## レスポンスパラメータ

No	論理名	物理名		備考
1	結果コード	result		成功時→success 失敗時→failed
2	受注番号	order_id		製本直送側の注文番号 Success 時のみ
3	エラー情報	errInfo		Failed 時のみ
4			error_code	エラーコード表参照
5			Error_message	エラーコード表参照

## 注文要求 API リクエストサンプル

## &lt;リクエスト送信時サンプル &gt; (PHP の例)

```
<?php
$arrData = array(
    'testmode' => 1,
    'access_key' => 'SCAPI00000001',
    'secret_key' => 'xxxxxxxxxxxxxxxxxxxx',
    'customer_id' => 1,
    'type' => 1,
    'order_date' => date('Y-m-d H:i:s'),
    //order_id' => 'shopid0001',
    'express' => 0,
    'cre_addr' => 1,
    'pre_regist_addr' => 0,
    'addr' => array(
        'zip01' => '530',
        'zip02' => '0011',
        'pref' => '27',
        'addr01' => '大阪市北区大深町',
        'addr02' => '3-1 グランフロント大阪',
        'tel' => '06-6150-2821',
        'name' => 'API テスト太郎',
        'email' => 'uketori_mail@cs-net.co.jp'
    ),
    'detail' => array(
        array('draft_id' => '2252', 'quantity' => '1'),
    )
);

$url = "https://www.seichoku.com/api/order.php"; //API Type1,3 用の送信先 URL

$json_data = json_encode($arrData);
$encode_json = base64_encode($json_data);
$postdata = array("param" => $encode_json);
$ch = curl_init($url);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTP_VERSION, CURL_HTTP_VERSION_1_1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $postdata);
$response = curl_exec($ch);
curl_close($ch);
```

```
//確認用
echo "JSON RESULT:<br/>¥n";
print_r($response);
echo "<br/><br/>¥n";

$objResponseJson = json_decode(base64_decode($response));

// レスポンス確認用
echo "OBJECT RESULT:<br/>¥n";
print_r($objResponseJson);
```

## &lt;結果レスポンス サンプル&gt;

## 成功時

```
{
  "result": "success",
  "order_id": "xxxxxx",
}
```

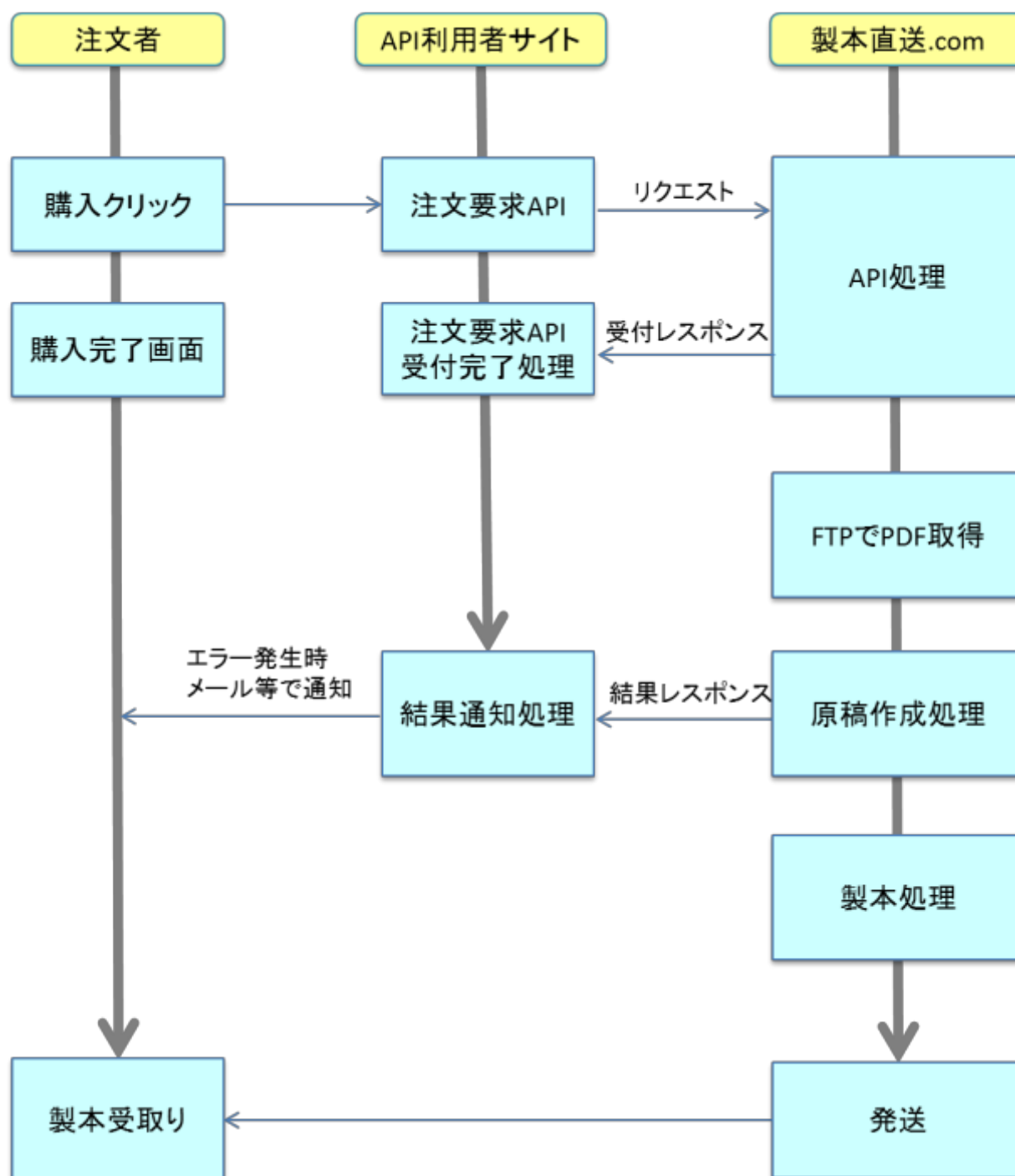
## 失敗時

```
{
  "result": "failed",
  "errInfo": [
    {
      "error_code": "ED002",
      "error_message": "明細1の quantity が不正です。"
    },
    {
      "error_code": "ED002",
      "error_message": "明細2の quantity が不正です。"
    }
  ]
}
```

## 5. API Type4（原稿登録＆注文要求型）

機能名	API 名	概要
原稿登録＆注文要求 API	Type4	原稿用の PDF の登録から注文までをワンストップで行う API となります。

### Type4 の処理の流れ



## API: setDraft

項目名	概要
概要	発注者から製本直送.com への注文要求
送信先 URL	<a href="https://www.seichoku.com/user_data/setDraft.php">https://www.seichoku.com/user_data/setDraft.php</a> (※テストモードでも送信先 URL は上記となります)
文字コード	UTF-8
POST パラメータ名	param

## リクエストパラメータ(Type4)

No	項目名	物理名		型	必須			備考
	論理名				表紙 不要 コース	表紙 自動 コース	表紙 作成 コース	
1	モード	testmode		半角数字 固定 1 桁	○	○	○	0→本番 1→テストモード ※テスト時はテストサーバーにデータが送信されます。
2	API アクセス キー	access_key		半 角 英 数 記号	○	○	○	製本直送.com のマイページに記載のものを記入 ※テスト時はテストサーバーのものを記入
3	API シークレ ットキー	Secret_key		半 角 英 数 記号	○	○	○	ご利用開始時にメールでご連絡したものを記入 ※テスト時はテストサーバーのものを記入
4	製本直送会 員 ID	customer_id		半角数字	○	○	○	製本直送.com のマイページに記載のものを記入(SCKを除く数字のみ) ※テスト時はテストサーバーのものを記入
5	注文番号	order_id		文字列				販売店様側の任意の管理番号
6	製本情報	list		配列				
7	原稿パス		path	文字列	○	○	○	原稿 PDF が置かれている ftp ドキュメントルートからのパス
8	原稿ファイル 名		filename	半 角 英 数 字	○	○	○	登録する原稿 PDF のファイル名
9	表紙自動コ ースフラグ		easy	半角数字 1 桁	○	○	○	1→表紙自動作成コース (PDF の1枚目が表紙) 0→それ以外
10	表紙不要コ		nocover	半角数字	○	○	○	1→表紙不要コース

	ースフラグ			1 桁				0→それ以外
11	ページサイズ		page_size	半角数字	○	○	○	※巻末の用紙サイズ参照
12	任意サイズの横幅		niniwidth	半角数字	△	△	△	仕上がり横幅(mm) page_size=10 の場合必須
13	任意サイズの高さ		niniheight	半角数字	△	△	△	仕上がり縦幅(mm) page_size=10 の場合必須
14	綴じ方		bind_type	半角数字 1 桁	○	○	○	1→中綴じ 2→無線綴じ
15	綴じ方向		bind_direction	半角数字 1 桁	○	○	○	1→右綴じ 2→左綴じ
16	本文用紙		paper_type	半角数字	○	○	○	※巻末の用紙種類参照
17	本文仕上がりカラー		text_color_type	半角数字	○	○	○	1→カラー 2→モノクロ
18	表紙の種類		cover_paper_data	カンマ区切りの半角数字	○	○	○	※巻末の表紙の種類、加工参照
19	表紙の厚さ		cover_thick	半角数字 1 桁	○	○	○	0→表紙厚め、 1→表紙柔らかめ
20	遊び紙		asobi	半角数字 1 桁	○	○	○	0→遊び紙なし 1→遊び紙あり(水色) 2→遊び紙あり(アイボリー)
21	表紙自動コースのウラ表紙		uracover	半角数字 1 桁		○		表紙自動作成の場合必須。 0→ウラ表紙白紙 1→PDF の最後のページをウラ表紙に割り当てる
22	塗り足し		nuritasi	半角数字 1 桁	○	○	○	0→塗り足しなし 1→塗り足しあり
23	全面印刷		zenmen	半角数字 1 桁	○	○	○	印刷カバー率が 50%を超える場合は1、超えない場合は0
24	注文冊数		rot_num	半角数字	○	○	○	注文冊数
25	表紙画像のパス		hyousi_path	文字列			○	easy=0, nocover=0 の場合必須。 表紙となる画像ファイル(jpg, png)が置かれている ftp ドキュメントルートからのパス
26	表紙画像のファイル名		hyousi_filename	半角英数字			○	easy=0, nocover=0 の場合必須。 表紙となる画像ファイルのファイル名 ※画像ファイルは、用紙サイズよりも周囲(上下左右)3mm ずつ大きく作成する必要があります(塗り足し分)。
27	速達フラグ	haisou		半角数字	○	○	○	0→普通 1→速達



								2→急行便 3→特急便
28	送付先登録フラグ	cre_addr		半角数字	○	○	○	0→送付先を登録しない 1→送付先を登録する
29	登録済み送付先 ID	pre_regist_addr		半角数字				事前に登録済みの送付先に送付する場合は、送付先 ID を記入(送付先 ID は、【マイページ】-【登録済み郵送先】で確認)。API で取得できるよう改良予定。
30	郵便番号1	zip01			△	△	△	送付先の郵便番号の前半。 pre_regist_addr が空の場合必須
31	郵便番号2	zip02			△	△	△	送付先の郵便番号の後半。 pre_regist_addr が空の場合必須
32	都道府県番号	pref			△	△	△	※都道府県コード参照 pre_regist_addr が空の場合必須
33	送付先の市町村名	addr01			△	△	△	pre_regist_addr が空の場合必須
34	送付先の市町村名以降	addr02			△	△	△	pre_regist_addr が空の場合必須
35	送付先の電話番号	tel			△	△	△	pre_regist_addr が空の場合必須
36	受取人名	name			△	△	△	pre_regist_addr が空の場合必須
37	受取人のメールアドレス	email						

### 受付レスポンス json パラメータ

No	論理名	物理名		備考
1	結果コード	result		成功時→success 失敗時→failed
2	エラー情報	errInfo		result =Failed 時のみ
3			error_code	※巻末のエラーコード表参照
4			Error_message	※巻末のエラーコード表参照

## 結果レスポンス json パラメータ

No	論理名	物理名			備考
1	結果コード	result			成功時→success 失敗時→failed
2	エラー情報	errInfo			result =Failed 時のみ
3			error_code		※巻末のエラーコード表参照
4			Error_message		※巻末のエラーコード表参照
5	登録完了情報	resultInfo			Result=success 時のみ
6			count		登録原稿件数
7			list		登録原稿配列
8				id	原稿 ID
9				customer_id	会員 ID(SCK を除く数字のみ)
10				file_name	原稿ファイル名
11				thumbnail	サムネイル画像のファイル名
12				price	製本原価(税込、円)
13				page_size	※巻末の【用紙サイズ】参照
14				page_size_x	用紙サイズ任意:縦
15				page_size_y	用紙サイズ任意:横
16				page_num	ページ数
17				text_paper_type	※巻末の【用紙種類】参照
18				text_color_type	1→カラー 2→モノクロ
19				cover_paper_type	※巻末の【表紙の種類・加工】の 先頭の数字参照
20				cover_thick	0→表紙厚め、 1→表紙柔らかめ
21				bind_type	1→中綴じ 2→無線綴じ
22				bind_direction	1→右綴じ 2→左綴じ
23				zenmen	印刷カバー率が 50%を超える場合は1、超えない場合は 0
24				bookcover	ブックカバー(※未実装)
25				asobi	0→遊び紙なし 1→遊び紙あり(水色) 2→遊び紙あり(アイボリー)
26				course	1→表紙不要コース 2→表紙自動コース 3→表紙作成コース
27				memo	メモ
28				create_date	作成日
29				order_id	注文番号

## 原稿登録&amp;注文要求 API Type4 リクエストサンプル

## &lt;リクエスト送信時サンプル&gt; (PHP の例)

```
<?php
$arrData = array(
    "testmode" => 1,
    "access_key"=>"SCAPI0000001",
    "secret_key"=>"xxxxxxxxxxxxxxxxxxxx",
    "customer_id"=>1,
    "order_id" => 'shop-00111',
    "list" => array(
        array(
            'path' => 'httpdocs/ftp_test/',
            'filename'=>'test.pdf',
            'easy' => 0,
            'nocover' => 0,
            'page_size' => 4,
            //'niniwidth' => "",
            //'niniheight' => "",
            'bind_type' => 2,
            'bind_direction' => 1,
            'paper_type' => 4,
            'text_color_type' => 1,
            'cover_paper_data' => '101,2,1',
            'cover_thick' => 0,
            'asobi' => 0,
            //'uracover' => 1,
            'nuritasi' => 1,
            'zenmen' => 1,
            'rot_num' => 1,
            'hyousi_path' => 'httpdocs/ ftp_test /cover/',
            'hyousi_filename' => 'book.png',
        ),
    ),
    'haisou' => 0,
    'cre_adr' => 0,
    //'pre_regist_addr' => "",
    'zip01' => '532',
```

```
'zip02' => '0003',
'pref' => '27',
'addr01' => '大阪市淀川区宮原',
'addr02' => '2-14-4 MF 新大阪ビル8F',
'tel' => '06-6150-2821',
'name' => 'シーズネット株式会社',
'email' => 'seichoku@cs-net.co.jp',
);

$url = "https://www.seichoku.com/user_data/setDraft.php"; //API Type4 用の送信先 URL

$json_data = json_encode($arrData);
$encode_json = base64_encode($json_data);
$postdata = array("param" => $encode_json);
$ch = curl_init($url);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTP_VERSION, CURL_HTTP_VERSION_1_1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $postdata);
$response = curl_exec($ch);
curl_close($ch);

//確認用
echo "JSON RESULT:<br/>¥n";
print_r($response);
echo "<br/><br/>¥n";

$objResponseJson = json_decode(base64_decode($response));

//レスポンス確認用
// ここでのレスポンスは“受付レスポンス”であって、“結果レスポンス”ではないことに注意
// 結果レスポンスの処理は後述する<結果レスポンス サンプル>を参照のこと
echo "OBJECT RESULT:<br/>¥n";
print_r($objResponseJson);
```

### <受付レスポンス サンプル>

#### 成功時

```
{
  "result":"success",
}
```

#### 失敗時

```
{
  "result":"failed",
  "errInfo":[
    {
      "error_code":"ED002",
      "error_message":"明細1の quantity が不正です。"
    },
    .....
  ]
}
```

### <結果レスポンス サンプル>

“結果レスポンス”は、API お申込時にご連絡いただいた、「結果レスポンスの受け取り URL」にレスポンスが送信されます。結果レスポンスを受信して、原稿作成に失敗していた場合は、再度 APIType4 をリクエストするか、あるいは利用者に入稿した原稿を確認してもらうなどの処理が必要になります（図9のフロー参照）。

“結果レスポンス”は、上記 URL に対して json データを base64 エンコードしたものを、パラメータ名 [param] として POST 送信しております。その為、受け取った POST 値を解析する際は、\$\_POST['param'] の値を base64 デコードして頂けましたら、元の json データを取得することが可能となります。

```
<?php
/**
 * 受け取ったパラメータをデコード処理する
 */

// POST されてきた値の取得
$params = htmlspecialchars(trim($_POST['param']));

// json オブジェクトとして処理する場合
$objParam = base64_decode($params);
```

```
// 配列として処理する場合
$arrParam = json_decode(base64_decode($param));

// JSON データをログ出し
$filepath = "log.txt";
$fp = fopen($filepath, "a+");
$log = print_r($objParam, true).PHP_EOL;
fwrite($fp, $log);
fclose($fp);

// 配列化データをログ出し
$filepath2 = "log2.txt";
$fp2 = fopen($filepath2, "a+");
$log2 = print_r($arrParam, true).PHP_EOL;
fwrite($fp2, $log2);
fclose($fp2);
```

## 成功時

```
{
  "result": "success",
  "resultInfo": {
    "count": 2,
    "list": [
      {
        "id": "2275",
        "customer_id": "1",
        "file_name": "test.pdf",
        "thumbnail": "hxlmiKWQMB_s.png",
        "price": "200",
        "page_size": "1",
        "page_size_x": null,
        "page_size_y": null,
        "page_num": "4",
        "text_paper_type": "16",
        "text_color_type": "1",
        "cover_paper_type": "100",
        "cover_thick": "1",
        "bind_type": "1",
        "bind_direction": "2",
        "zenmen": "1",
        "bookcover": "0",
        "asobi": "0",
```

```

        "course":"3",
        "memo":null,
        "create_date":"2015-05-12 15:57:02",
        "order_id":2825
    },
    {
        "id":"2276",
        "customer_id":"1",
        "file_name":"test2.pdf",
        "thumbnail":"KCCg0laWIR_s.png",
        "price":"450",
        "page_size":"1",
        "page_size_x":null,
        "page_size_y":null,
        "page_num":"4",
        "text_paper_type":"16",
        "text_color_type":"1",
        "cover_paper_type":"100",
        "cover_thick":"1",
        "bind_type":"1",
        "bind_direction":"2",
        "zenmen":"1",
        "bookcover":"0",
        "asobi":"0",
        "course":"3",
        "memo":null,
        "create_date":"2015-05-12 15:57:03",
        "order_id":2826
    }
}

```

**失敗時**

```

{
    "result":"failed",
    "errInfo":[
        {
            "error_code":"ED002",
            "error_message":"明細 1 の quantity が不正です。"
        },
        . . . . .
    ]
}

```

## 6. API Type5(原稿リスト取得)

機能名	API 名	概要
原稿リスト取得 API	Type5	製本直送.com に登録済みの原稿を一覧で取得する API となります。

### API:getDraftList

項目名	概要
概要	製本直送.com へ登録済みの原稿リストを取得する
送信先 URL	<a href="https://www.seichoku.com/api/getDraftList.php">https://www.seichoku.com/api/getDraftList.php</a> (※テストモードでも送信先 URL は上記となります)
文字コード	UTF-8
POST パラメータ名	param

N o	項目名		型	必須	複数	備考
	論理名	物理名				
1	モード	testmode	半角数字 固定 1 桁	○		0→本番 1→テストモード ※テスト時はテストサーバーにデータが送信されます。
2	API アクセスキー	access_key	半角英数字 一部記号	○		製本直送.com のマイページに記載のものを記入 ※テスト時はテストサーバーのものを記入
3	API シークレットキー	secret_key	半角英数字 一部記号	○		ご利用開始時にメールでご連絡したものを記入 ※テスト時はテストサーバーのものを記入
4	会員 ID	customer_id	半角数字	○		製本直送.com のマイページに記載のものを記入(SCK を除く数字のみ) ※テスト時はテストサ



						ーバーのものを記入
--	--	--	--	--	--	-----------

## レスポンスパラメータ

No	論理名	物理名		備考
1	結果コード	result		成功時→success 失敗時→failed
2	原稿リスト	list		Success 時のみ
3	原稿 ID		id	原稿 ID
4	会員 ID		customer_id	会員 ID (SCK を除く数字のみ)
5	ファイル名		file_name	原稿ファイル名
6	サムネイル画像		thnmbnail	サムネイル画像のファイル名
7	原価		price	製本原価 (税込、円)
8	ページサイズ		page_size	※巻末の【用紙サイズ】参照
9	ページ数		page_num	ページ数
10	本文用紙		text_paper_type	※巻末の【用紙種類】参照
11	本文仕上がりカラー		text_color_type	1→カラー 2→モノクロ
12	表紙の種類		cover_paper_type	※巻末の【表紙の種類・加工】の先頭の数字参照
13	表紙の厚さ		cover_thick	0→表紙厚め、 1→表紙柔らかめ
14	綴じ方		bind_type	1→中綴じ 2→無線綴じ
15	綴じ方向		bind_direction	1→右綴じ 2→左綴じ
16	全面印刷		zenmen	印刷カバー率が50%を超える場合は1、超えない場合は0
17	遊び紙		asobi	0→遊び紙なし 1→遊び紙あり (水色) 2→遊び紙あり (アイボリー)
18	選択コース		course	1→表紙不要コース 2→表紙自動コース 3→表紙作成コース
19	メモ		memo	メモ
20	原稿登録日		create_date	作成日
21	エラー情報	errInfo		Failed 時のみ
22			error_code	※巻末のエラーコード表参照
23			Error_message	※巻末のエラーコード表参照
24			error_code	※巻末のエラーコード表参照
25			Error_message	※巻末のエラーコード表参照

## 原稿リスト取得 API (Type5)リクエストサンプル

## &lt;リクエスト送信時&gt;(PHP の例)

```
<?php
$arrData = array(
    "testmode" => 1,
    "access_key" => 'SCAPI00000001',
    "secret_key" => 'xxxxxxxxxxxxxxxxxxxx',
    "customer_id" => 1,
);

$url = "https://www.seichoku.com/api/getDraftList.php"; //API Type5 用の送信先 URL

$json_data = json_encode($arrData);
$encode_json = base64_encode($json_data);
$postdata = array("param" => $encode_json);
$ch = curl_init($url);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTP_VERSION, CURL_HTTP_VERSION_1_1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $postdata);
$response = curl_exec($ch);
curl_close($ch);
/確認用
echo "JSON RESULT:<br/>¥n";
print_r($response);
echo "<br/><br/>¥n";

$objResponseJson = json_decode(base64_decode($response));
// レスポンス確認用
echo "OBJECT RESULT:<br/>¥n";
print_r($objResponseJson);
```

## &lt;レスポンス サンプル&gt;

## 成功時

```
{
  "result": "success",
  "count": "1"
  "list": [
    {
      "id": "2275",
      "customer_id": "1",
      "file_name": "test.pdf",
      "thumbnail": "hxlmiKWQMB_s.png",
      "price": "200",
      "page_size": "1",
      "page_size_x": null,
      "page_size_y": null,
      "page_num": "4",
      "text_paper_type": "16",
      "text_color_type": "1",
      "cover_paper_type": "100",
      "cover_thick": "1",
      "bind_type": "1",
      "bind_direction": "2",
      "zenmen": "1",
      "bookcover": "0",
      "asobi": "0",
      "course": "3",
      "memo": null,
      "create_date": "2015-05-12 15:57:02",
      "order_id": 2825
    }
  ],
}
```

## 失敗時

```
{
  "result": "failed",
  "errInfo": [
    {
      "error_code": "ED002",
      "error_message": "明細 1 の quantity が不正です。"
    },
    . . . . .
  ]
}
```

## 7. API Type6（原稿登録要求）

機能名	API 名	概要
原稿登録 API	Type6	製本直送.comに原稿を登録するAPIとなります。

### API: registerDraft

項目名	概要
概要	製本直送.com への原稿登録要求
送信先 URL	<a href="https://www.seichoku.com/user_data/registerDraft.php">https://www.seichoku.com/user_data/registerDraft.php</a> (※テストモードでも送信先 URL は上記となります)
文字コード	UTF-8
POST パラメータ名	Param

### リクエストパラメータ

No	項目名		型	必須			備考	
	論理名	物理名			表紙不要コース	表紙自動コース	表紙作成コース	
1	モード	testmode		半角数字 固定 1 桁	○	○	○	0→本番 1→テストモード ※テスト時はテストサーバーにデータが送信されます。
2	API アクセス キー	access_key		半角英数 記号	○	○	○	製本直送.com のマイページに記載のものを記入 ※テスト時はテストサーバーのものを記入
3	API シークレットキー	Secret_key		半角英数 記号	○	○	○	ご利用開始時にメールでご連絡したものを記入 ※テスト時はテストサーバーのものを記
4	製本直送会員 ID	customer_id		半角数字	○	○	○	製本直送.com のマイページに記載のものを記入(SCK を除く数字のみ) ※テスト時はテストサーバーのものを記入
5	製本情報	list		配列				
6	原稿パス		path	文字列	○	○	○	原稿 PDF が置かれている ftp ドキュメントルート

								からのパス
7	原稿ファイル名		filename	半角英数字	○	○	○	登録する原稿 PDF のファイル名
8	表紙自動コースフラグ		easy	半角数字 1桁	○	○	○	1→表紙自動作成コース (PDF の1枚目が表紙) 0→それ以外
9	表紙不要コースフラグ		nocover	半角数字 1桁	○	○	○	1→表紙不要コース 0→それ以外
10	ページサイズ		page_size	半角数字	○	○	○	※巻末の用紙サイズ参照
11	任意サイズの の横幅		niniwidth	半角数字	△	△	△	仕上がり横幅(mm) page_size=10 の場合必須
12	任意サイズの の高さ		niniheight	半角数字	△	△	△	仕上がり縦幅(mm) page_size=10 の場合必須
13	綴じ方		bind_type	半角数字 1桁	○	○	○	1→中綴じ 2→無線綴じ
14	綴じ方向		bind_direction	半角数字 1桁	○	○	○	1→右綴じ 2→左綴じ
15	本文用紙		paper_type	半角数字	○	○	○	※巻末の用紙種類参照
16	本文仕上がりカラー		text_color_type	半角数字	○	○	○	1→カラー 2→モノクロ
17	表紙の種類		cover_paper_data	カンマ区切りの半角数字	○	○	○	※巻末の表紙の種類、加工参照
18	表紙の厚さ		cover_thick	半角数字 1桁	○	○	○	0→表紙厚め、 1→表紙柔らかめ
19	遊び紙		asobi	半角数字 1桁	○	○	○	0→遊び紙なし 1→遊び紙あり(水色) 2→遊び紙あり(アイボリー)
20	表紙自動コースのウラ表紙		uracover	半角数字 1桁		○		表紙自動作成の場合 必須。 0→ウラ表紙白紙 1→PDF の最後のページをウラ表紙に割り当て
21	塗り足し		nuritasi	半角数字 1桁	○	○	○	0→塗り足しなし 1→塗り足しあり
22	全面印刷		zenmen	半角数字 1桁	○	○	○	印刷カバー率が 50%を超える場合は1、超えない場合は 0
24	表紙画像のパス		hyousi_path	文字列			○	easy=0, nocover=0 の場合必須。表紙となる画像ファイル (jpg, png)が置かれている ftp ドキュメ

							ントルート からのパス
25	表紙画像の ファイル名		hyousi_fil enam e	半角英数 字		○	easy=0, nocover=0 の場 合必須。表紙となる画 像ファイル のファイル名 ※画像ファイルは、用紙 サイズよりも周囲(上 下 左右)3mm ずつ大きく作 成する必要があります (塗り足し分)。

## 受付レスポンス json パラメータ

No	論理名	物理名		備考
1	結果コード	result		成功時→success、 失敗時→failed
2	エラー情報	errInfo		result = Failed 時のみ
3			error_code	※エラーコード表参照
4			error_message	※エラーコード表参照

## 結果レスポンス json パラメータ

No	論理名	物理名			備考
1	結果コード	result			成功時→success、 失敗時→failed
2	エラー情報	errInfo			result = Failed 時のみ
			error_code		※エラーコード表参照
			error_message		※エラーコード表参照
	登録完了情 報	resultI nfo			result = success 時のみ
			list		登録原稿配列
				id	原稿 ID
				customer_id	会員 ID (SCK を除く数字のみ)
				file_name	原稿ファイル名
				thumbnail	サムネイル画像ファイル名
				price	製本原価 (税込、円)
				page_size	※巻末の【用紙サイズ】参照
				page_num	ページ数
				text_paper_t ype	※巻末の【用紙種類】参照
				text_color_ty pe	1→カラー 2→モノクロ
				cover_paper_	※巻末の【表紙の種類・加工】の

				type	先頭の数字参照
				bind_type	1→中綴じ 2→無線綴じ
				bind_direction	1→右綴じ 2→左綴じ
				create_date	作成日

## 原稿登録 API Type6 リクエストサンプル

### < リクエスト送信時サンプル > (PHP の例)

```
<?php
$arrData = array(
    "testmode" => 1,
    "access_key"=>"SCAPI00000001",
    "secret_key"=>"xxxxxxxxxxxxxxxxxxxx",
    "customer_id">1,
    "list" => array(
        array(
            'path' => 'httpdocs/ftp_test/',
            'filename'=>'test.pdf',
            'easy' => 0,
            'nocover' => 0,
            'page_size' => 4,
            //'niniwidth' => "",
            //'niniheight' => "",
            'bind_type' => 2,
            'bind_direction' => 1,
            'paper_type' => 4,
            'text_color_type' => 1,
            'cover_paper_data' => '101,2,1',
            'cover_thick' => 0,
            'asobi' => 0,
            //'uracover' => 1,
            'nuritasi' => 1,
            'zenmen' => 1,
            'rot_num' => 1,
            'hyousi_path' => 'httpdocs/ ftp_test /cover/',
```

```
        'hyousi_filename' => 'book.png'
    )
)
);

$url = " https://www.seichoku.com/user_data/registerDraft.php ";
//API Type6 用の送信先 URL

$json_data = json_encode($arrData);
$encode_json = base64_encode($json_data);
$postdata = array("param" => $encode_json);
$ch = curl_init($url);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTP_VERSION, CURL_HTTP_VERSION_1_1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $postdata);
$response = curl_exec($ch);
curl_close($ch);

//確認用
echo "JSON RESULT:<br/>¥n";
print_r($response);
echo "<br/><br/>¥n";

$objResponseJson = json_decode(base64_decode($response));
// レスポンス確認用
echo "OBJECT RESULT:<br/>¥n";
print_r($objResponseJson);
```

## <受付レスポンス サンプル>

### 成功時

```
{
    "result":"success",
}
```



## 失敗時

```
{
  "result": "failed",
  "errInfo": [
    {
      "error_code": "ED002",
      "error_message": "明細1の quantity が不正です。"
    },
    .....
  ]
}
```

### <結果レスポンス サンプル>

“結果レスポンス”は、API お申込時にご連絡いただいた、「結果レスポンスの受け取り URL」にレスポンスが送信されます。結果レスポンスを受信して、原稿作成に失敗していた場合は、再度 APIType6 をリクエストするか、あるいは利用者に入稿した原稿を確認してもらうなどの処理が必要になります (Type 4 の図9のフロー参照)。

“結果レスポンス”は、上記 URL に対して json データを base64 エンコードしたものを、パラメータ名 [param] として POST 送信しております。その為、受け取った POST 値を解析する際は、\$\_POST['param']の値を base64 デコードして頂けましたら、元の json データを取得することが可能となります。

```
<?php
/**
 * 受け取ったパラメータをデコード処理する
 */
// POST されてきた値の取得
$params = htmlspecialchars(trim($_POST['param']));

// json オブジェクトとして処理する場合
$objParam = base64_decode($params);

// 配列として処理する場合
$arrParam = json_decode(base64_decode($params));

// JSON データをログ出し
```

```
$filepath = "log.txt";
$fp = fopen($filepath, "a+");
$log = print_r($objParam, true).PHP_EOL;
fwrite($fp, $log);
fclose($fp);

// 配列化データをログ出し
$filepath2 = "log2.txt";
$fp2 = fopen($filepath2, "a+");
$log2 = print_r($arrParam, true).PHP_EOL;
fwrite($fp2, $log2);
fclose($fp2);
```

### 成功時

```
{  "result": "success",
   "resultInfo":
   {
       "list": [
           {
               "id": "2275",
               "customer_id": "1",
               "file_name": "test.pdf",
               "thumbnail": "hxlmiKWQMB_s.png",
               "price": "200",
               "page_size": "1",
               "page_num": "4",
               "text_paper_type": "16",
               "text_color_type": "1",
               "cover_paper_type": "100",
               "cover_thick": "1",
               "bind_type": "1",
               "bind_direction": "2",
               "zenmen": "1",
               "bookcover": "0",
               "asobi": "0",
               "course": "3",
               "memo": null,
               "create_date": "2015-05-12 15:57:02",
```

```
        "order_id":2825
      ]
    }
  }
```

#### 失敗時

```
{
  "result":"failed",
  "errInfo":[
    {
      "error_code":"ED002",
      "error_message":"明細1の quantity が不正です。"
    }, .....
  ]
}
```

## 8. 巻末

### 用紙サイズ

code	用紙サイズ
1	A4
2	A5
3	B5
4	B6
5	A6
6	新書判
10	任意サイズ

### 用紙種類

Code	用紙種類
4	b7 バルキー
6	ラフクリーム
7	ホワイトしらおい
16	HS スノーフォース
9	モンテシオン
11	マシュマロ CoC
12	コート紙 1 (グロス, 100g/㎡)
13	コート紙 2 (グロス 128g/㎡)
14	コート紙 1 (マット 100g/㎡)
15	コート紙 2 (マット 128g/㎡)

### 表紙の種類・加工

Code	表紙の種類・加工
100,2,1	表紙カラー＆ラミネート加工なし
101,2,1	表紙カラー＆ラミネート加工（マット）
102,2,1	表紙カラー＆ラミネート加工（グロス）
104,2,1	表紙モノクロ＆ラミネート加工なし
105,2,1	表紙モノクロ＆ラミネート加工（マット）
106,2,1	表紙モノクロ＆ラミネート加工（グロス）

<b>108,2,1</b>	表紙なし
----------------	------

※表紙なしの場合、“nocover”を「1」にして、“cover\_paper\_data”を「108, 2, 1」にセットして下さい。

## 都道府県コード

code,都道府県	code,都道府県	code,都道府県	code,都道府県
1,北海道	13,東京都	25,滋賀県	37,香川県
2,青森県	14,神奈川県	26,京都府	38,愛媛県
3,岩手県	15,新潟県	27,大阪府	39,高知県
4,宮城県	16,富山県	28,兵庫県	40,福岡県
5,秋田県	17,石川県	29,奈良県	41,佐賀県
6,山形県	18,福井県	30,和歌山県	42,長崎県
7,福島県	19,山梨県	31,鳥取県	43,熊本県
8,茨城県	20,長野県	32,島根県	44,大分県
9,栃木県	21,岐阜県	33,岡山県	45,宮崎県
10,群馬県	22,静岡県	34,広島県	46,鹿児島県
11,埼玉県	23,愛知県	35,山口県	47,沖縄県
12,千葉県	24,三重県	36,徳島県	

## エラーコード表

Code	Message
<b>EF000</b>	FTP の接続ができません
<b>EF001</b>	path に移動できません
<b>EF002</b>	ファイルが存在しません
<b>EF003</b>	ファイルのダウンロード失敗

Code	Message
<b>E000</b>	パラメータ解析エラー
<b>EM000</b>	access_key の値が不正です
<b>EM001</b>	secret_key の値が不正です
<b>EM002</b>	customer_id の値が不正です

<b>EM003</b>	testmode の値が不正です
<b>EM004</b>	type の値が不正です
<b>EM005</b>	order_date の値が不正です
<b>EM006</b>	order_id の値が不正です
<b>EM007</b>	access_key secret_key 一致しません
<b>EM008</b>	list の構造が不正です
<b>EM009</b>	既に登録されたデータです
<b>EA000</b>	zip01 の値が不正です
<b>EA001</b>	zip02 の値が不正です
<b>EA002</b>	pref の値が不正です
<b>EA003</b>	addr01 の値が不正です
<b>EA004</b>	tel の値が不正です
<b>EA005</b>	name の値が不正です
<b>ED001</b>	draft_id の値が不正です
<b>ED002</b>	quantity の値が不正です
<b>EL001</b>	有効な原稿データが存在しません

Code	Message	エラー内容
<b>EZ001</b>	パラメータ設定が不正です。	『表紙不要コース』でページ数が 49 ページ以上に設定されています。
<b>EZ002</b>	パラメータ設定が不正です。	『表紙不要コース』で綴じ方に『無線綴じ』が設定されています。
<b>EZ003</b>	パラメータ設定が不正です。	『表示自動コース』でページ数が 22 ページ未満且つ、綴じ方が『無線綴じ』に設定されています。
<b>EZ004</b>	パラメータ設定が不正です。	『表示自動コース』でページ数が 51 ページ以上 且つ、綴じ方が『中綴じ』に設定されています。
<b>ES001</b>	変換に失敗しました。	ファイル変換処理で失敗しました。
<b>ES002</b>	セキュリティの掛かった PDF ファイルです。	アップロードされた PDF ファイルに何らかのセキュリティが付与されている為、変換処理が行えません。

<b>ES003</b>	変換に失敗しました。	ファイル変換処理で失敗しました。
<b>ES004</b>	変換に失敗しました。	ファイル変換処理で失敗しました。
<b>ES005</b>	変換に失敗しました。	ファイル変換処理で失敗しました。
<b>ES006</b>	変換に失敗しました。	ファイル変換処理で失敗しました。