

帳票 API を利用した カスタマイズ帳票作成

2021 年 10 月 1 日

日本医師会 ORCA 管理機構

内容

0. 改定履歴.....	1
1. 概要	2
2. 対象	2
3. 帳票 API を利用したカスタマイズ帳票作成の動作概要.....	2
4. push-exchanger 及び print001 プラグインの動作概要	3
5. push-exchanger のインストール及び設定	4
インストール.....	4
push-exchanger の設定	4
設定事前準備	5
config.yml の編集	5
pusher との接続確認.....	6
6. print001 プラグインの設定	9
7. プラグインモジュールの差し替え.....	10
8. 外来カルテ 1 号用紙の出力	10
日レセプログラムオプションの設定.....	10
push-exchanger の起動.....	10
日レセによるカルテ発行操作	10
9. karte_no1 プラグインの解説.....	12
10. その他の帳票プラグイン	14
11. push-exchanger を利用しないカスタマイズ帳票の作成	15
12. まとめ.....	15
13. 参考文献.....	16

0. 改定履歴

初版・・・2017 年 3 月 30 日

第 2 版・・・2017 年 11 月 24 日

 インストーラを差し替え

第 3 版・・・2017 年 12 月 15 日

 インストーラを差し替え(32bit 対応)

第 4 版・・・2018 年 4 月 25 日

 インストーラを差し替え

- ・ SSL 接続時のサーバ認証においてホスト名の検証を行わない不具合を修正しました。
- ・ サーバ印刷プラグイン(server_print.rb)による Windows 環境の印刷時にエラーが発生する不具合を修正しました。

第 5 版・・・2020 年 12 月 18 日

 インストーラを差し替え

 内容見直し

第 6 版・・・2021 年 10 月 1 日

 インストーラを差し替え

1. 概要

本文書では日レセクラウド・WebORCA 環境において帳票 API を利用した医療機関側でのカスタマイズ帳票作成について説明する。

前半は帳票 API を利用したカスタマイズ帳票作成の概要について説明する。後半は日レセ PUSH 通知駆動フレームワーク push-exchanger、帳票印刷プラグイン print001 を利用したカスタマイズ帳票の出力について解説を行う。

2. 対象

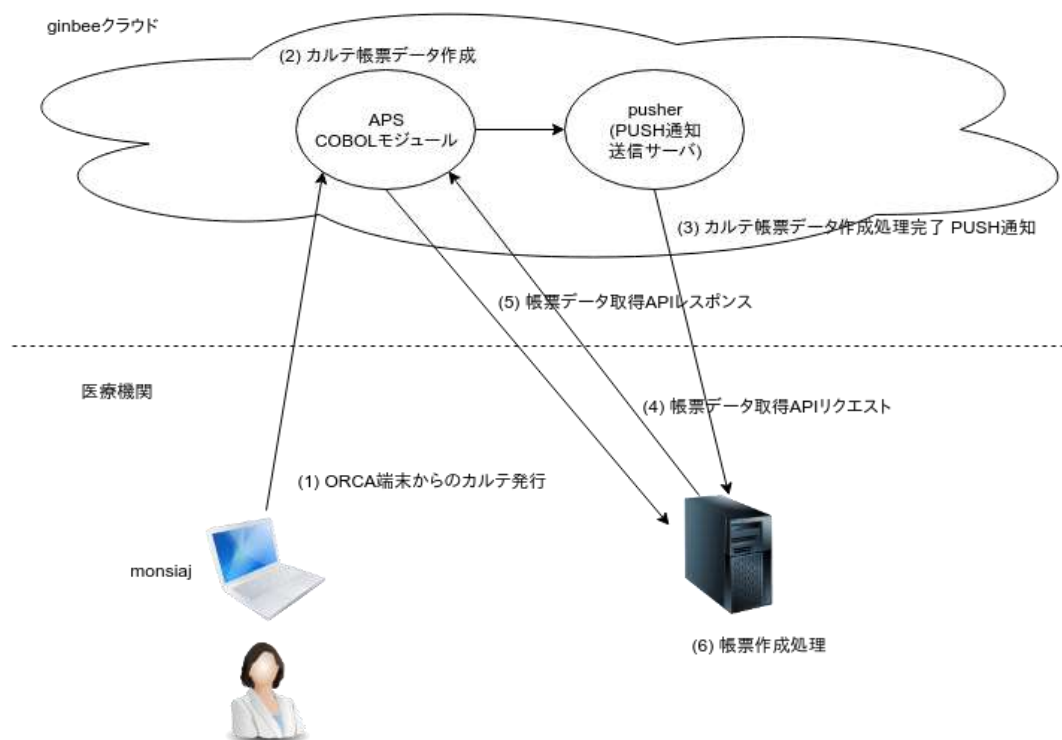
本文書の対象とする環境は以下とする。

- サーバ環境
 - 日レセクラウド
 - WebORCA
- クライアント環境
 - Windows OS 32bit/64bit 版

3. 帳票 API を利用したカスタマイズ帳票作成の動作概要

帳票 API によるカスタマイズ帳票作成の処理の流れについて説明する。

以下のような流れになる。



1. 医療機関の日レセ端末より窓口帳票の印刷指示を行う
 - 事前に日レセ側で帳票 API 利用の設定が必要
2. 日レセクラウド・WebORCA の日レセ APS で帳票データを作成する
3. 日レセクラウド・WebORCA の PUSH 通知サーバ(pusher)より、帳票データ作成完了の PUSH 通知を送信する
4. 医療機関側で PUSH 通知を受信し、PUSH 通知に含まれる帳票 ID を元に帳票データ取得 API へリクエストを行い、帳票データを取得する
5. 医療機関側で帳票データを使用し、帳票の編集、印刷を行う

4. push-exchanger 及び print001 プラグインの動作概要

帳票 API によるカスタマイズ帳票作成の一例として日レセ PUSH 通知駆動フレームワーク push-exchanger とそのプラグインである print001 プラグインを紹介する。

push-exchanger は日レセ PUSH 通知クライアントであり、PUSH 通知を受信して PUSH 通知の種別毎に対応するプラグインを起動する。

print001 プラグインは帳票 API の PUSH 通知に対応するプラグインであり、起動すると帳票データ取得 API へリクエストを行い、帳票データを取得する。

帳票データを取得すると個々の帳票プラグインを呼び出し、帳票編集後、印刷処理を行う。

カルテ プラグイン (帳票作成)	処方箋 プラグイン (帳票作成)	
print001プラグイン (日レセ帳票API処理)		CLAIMプラグインなど
push-exchanger (PUSH通知受信)		

push-exchanger 及び print001 プラグインの詳細については push-exchanger の仕様書を参照。

5. push-exchanger のインストール及び設定

Windows 環境への push-exchanger のインストールと設定について説明する。

インストール

以下のリンクからインストーラをダウンロードし実行する。

<https://ftp.orca.med.or.jp/pub/data/receipt/download/windows/ginbee/push-exchanger-installer-0.0.26.exe>

完了すると C:\Program Files (x86)\push-exchanger に push-exchanger 及び Ruby 実行環境、サンプルプラグイン一式がインストールされている。

push-exchanger の設定

設定は YAML 形式の設定ファイル C:\Program Files (x86)\push-exchanger\config.yml を編集することで行う。

設定事前準備

日レセクラウド・WebORCA 環境の pusher に接続するため以下が必要になる。事前に日レセクラウド・WebORCA 環境管理画面にログインして取得しておく。

- 日レセ API キー
- 日レセ API 用 SSL クライアント証明書
 - CA 証明書(ca.crt)
 - クライアント証明書(*.crt)
 - クライアント証明書秘密鍵(*.pem)
 - 秘密鍵パスフレーズ

config.yml の編集

config.yml の以下の項目を編集する。

項目名	内容	説明
:ws_server	日レセクラウド VPN の場合 : wss://pusher-proxy.orca.orcamo.jp/ws 日レセクラウド TLS1.2 の場合 : wss://pusher-proxy.glorca.orcamo.jp/ws WebORCA の場合 : wss://app.weborca.orcamo.jp/ws	pusher の URI
:api_key	API キー	
:api_server	日レセクラウド VPN の場合 : ap-proxy.orca.orcamo.jp 日レセクラウド TLS1.2 の場合 : ap-proxy.glorca.orcamo.jp WebORCA の場合 : app.weborca.orcamo.jp	API サーバ
:api_port	日レセクラウドの場合 : 8080 WebORCA の場合 : 443	API サーバポート
:use_ssl	True	
:ca_cert	"C:/Program Files (x86)/push-exchanger/cert/ca.crt"	CA 証明書

:cert	"C:/Program Files (x86)/push-exchanger/cert/tenant.crt"	証明書
:cert_key	"C:/Program Files (x86)/push-exchanger/cert/tenant.pem"	証明書秘密鍵
:passphrase	パスフレーズ	
:use_weborca	WebORCA の場合 : true	

※:ca_cert、:cert、:cert_key はダブルクォート「"」で括ってください。

※パス区切り文字は「¥」ではなく「/」とすること

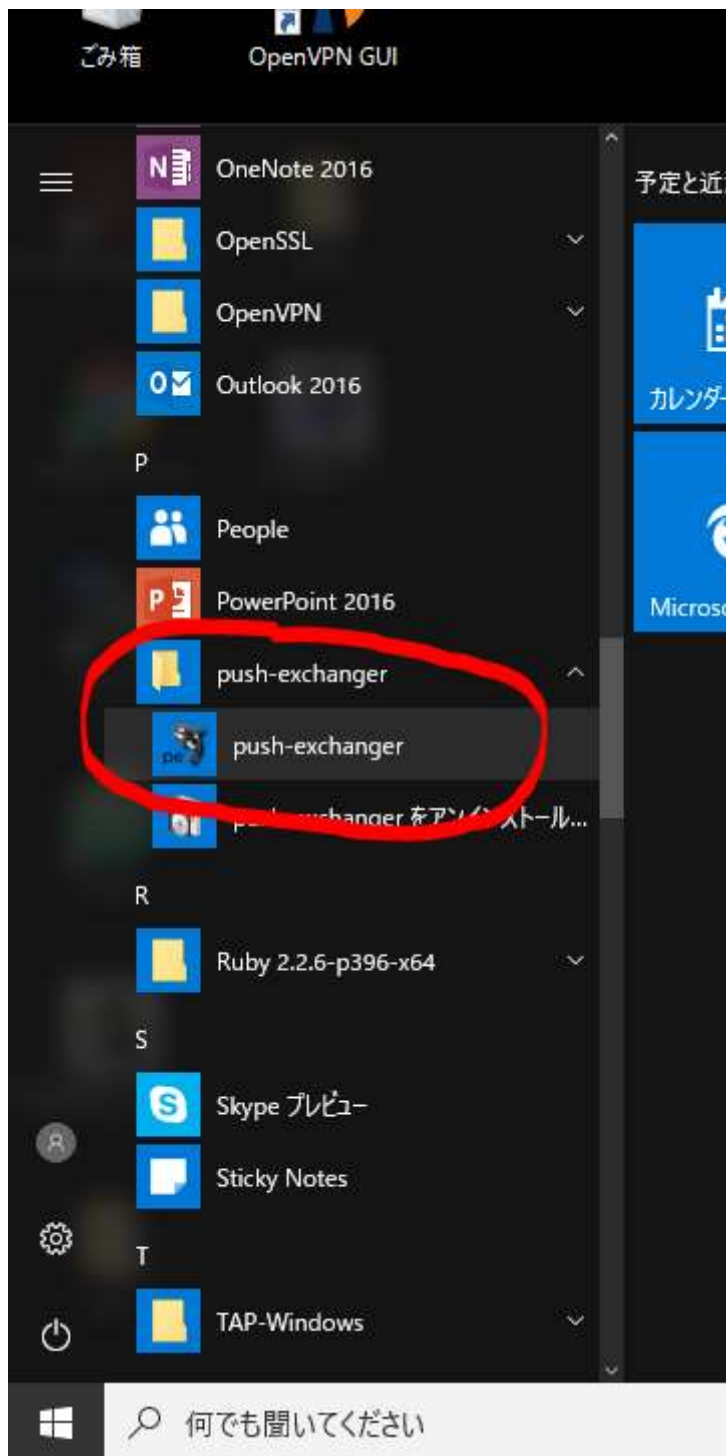
※パス名に日本語を含まないようにしてください。

設定編集後、所定のディレクトリに CA 証明書、クライアント証明書を配置する。

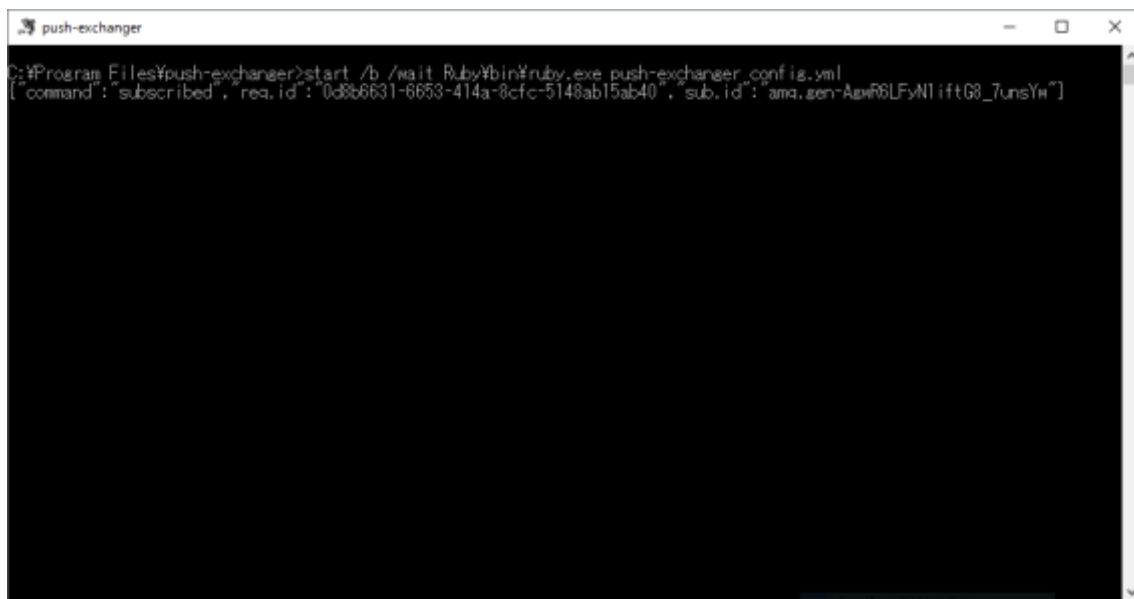
pusher との接続確認

push-exchanger を起動し、pusher との接続を確認する。

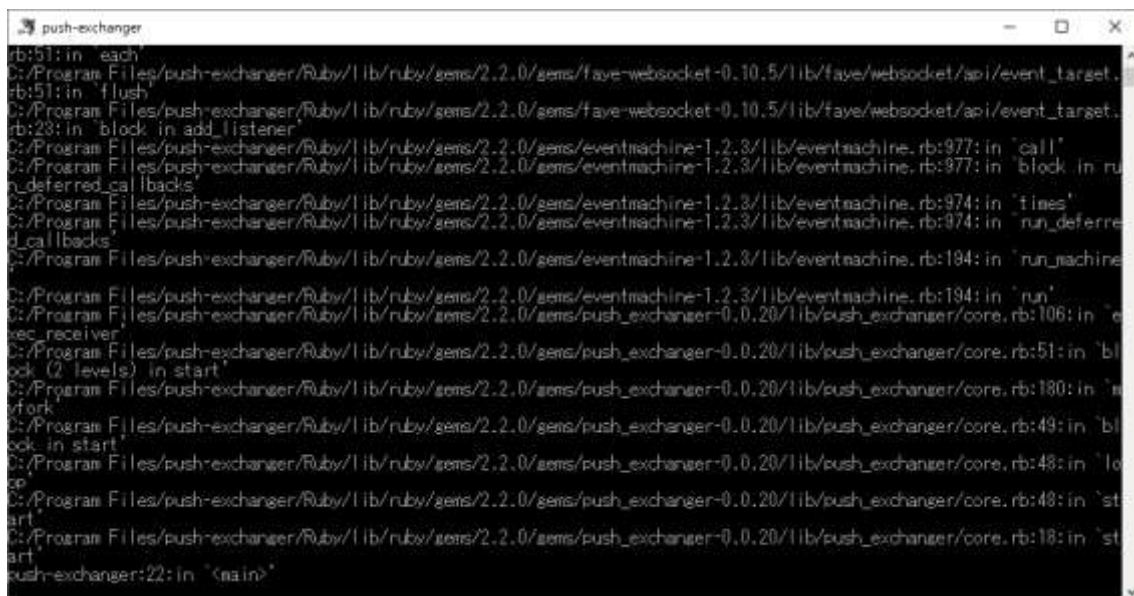
スタートメニューから push-exchanger フォルダを選択し push-exchanger をクリックして起動する。



以下のようなコマンドプロンプトが表示される。



push-exchanger はコマンドプロンプト中で動作し、pusher から受信したメッセージやエラーメッセージをコマンドプロンプト中に表示する。上記のような subscribed となっている JSON 文字列が表示されると接続に成功している。



上記のように Ruby のエラーメッセージが表示される場合は接続に失敗している。

C:\Program Files (x86)\push-exchanger\log\push-exchanger.log.txt に作成されるログを確認した上で、以下の確認を行う。

- 設定ファイルの確認
 - 不正な YAML 形式となっていないか
 - API キーが正しいか
 - CA 証明書、クライアント証明書の記載が正しいか

- CA 証明書、クライアント証明書を正しく設置しているか
- 正しいパスフレーズを設定しているか

6. print001 プラグインの設定

print001 プラグインの設定は

C:¥Program Files (x86)¥push-exchanger¥plugin¥print001¥print001.yml を編集することで行う。

ここでは以下のプリンタ及び印刷の設定のみを行う。

項目名	説明
:printer_settings:	プリンタ設定ハッシュ
:default:	デフォルトのプリンタ指定
:print_command:	印刷コマンド

:printer_settings:にて帳票 ID と出力するプリンタを指定する。帳票 ID が:default:の場合はデフォルトプリンタとなる。ここではデフォルトプリンタの設定のみとする。

プリンタ名には Windows に登録されているプリンタを設定する。

プリンタ名に空白が含まれる場合は以下のようにダブルクォートで括る。

プリンタ名にマルチバイト文字は利用不可。

```
:printer_settings:
  :default: "RICOH IPSiO NX85S RPD"
```

:print_command:に実際に印刷するコマンドを記載する。%FILE%、%PRINTER%はコマンド実行時にそれぞれ PDF ファイル名、プリンタ名に置き換えられる。コマンドやプリンタ名に空白が含む場合は全体をシングルクォートで括った上でそれぞれダブルクォートで括る必要がある。

```
:print_command: "'C:¥Program Files (x86)¥Adobe¥Acrobat Reader
DC¥Reader¥AcroRd32.exe" /N /T "%FILE%" "%PRINTER%"'
```

上記の例ではアクロバットリーダーを起動して印刷を行う設定となっている。

その他の設定については push-exchanger の仕様書を参照。

7. プラグインモジュールの差し替え

push-exchanger のインストーラーに含まれているサンプル古いため最新のものと差し替える。

1. C:\Program Files (x86)\push-exchanger\plugin\print001\plugin フォルダ内にあるフォルダおよびファイルすべて削除する。
2. 帳票データ取得 API
https://www.orca.med.or.jp/receipt/tec/push-api/report_data_api.html
に掲載している帳票プラグインをダウンロードおよび解凍する。
3. 解凍後、フォルダ内のフォルダおよびファイルすべてを C:\Program Files (x86)\push-exchanger\plugin\print001\plugin へコピーする。

8. 外来カルテ 1 号用紙の出力

外来カルテ 1 号用紙(karte_no1 プラグイン)のカスタマイズ帳票を出力する。

日レセプログラムオプションの設定

外来カルテ 1 号用紙の帳票 API を有効にするため、日レセでプログラムオプションの設定を行う。

1. システム管理より「1910 プログラムオプション情報」を選択し、プログラムオプション設定画面を表示する
2. プログラム名一覧から ORCHC01 を選択する
3. オプション内容の入力欄に以下の 2 行を追加する

API_DATA=1 API_CUSTOM_ID=karte_no1

push-exchanger の起動

スタートメニューから push-exchanger を起動し、pusher と接続する。

日レセによるカルテ発行操作

日レセでカルテ発行操作を行う。

1. 患者登録画面を開く
2. 患者 ID を入力してカルテ発行を行う患者を開く
3. 生年月日の右のコンボボックスから「1 カルテ発行あり」を選択する
4. 登録ボタン押下

プログラムオプションの設定が正しくされていれば push-exchanger のコマンドプロトに以下のような帳票 API の PUSH 通知が表示されます。表示されない場合はプログラムオプションの設定を確認する。

```
C:\Program Files (x86)\push-exchanger>start /b /wait Ruby\bin\ruby.exe push-  
exchanger config.yml  
  
{ "command": "subscribed", "req.id": "d977c8cf-1eac-4c9f-977e-  
3778eaae4f7a", "sub.id": "amq.gen-btIEDmYr_DenTgJKjOvEKw"}  
  
{ "command": "event", "sub.id": "amq.gen-  
btIEDmYr_DenTgJKjOvEKw", "data": { "body": [ { "Custom_ID": "karte_no1", "Data_ID"  
": "ONLINE#fe28c157-a429-4e87-bc36-  
e64f4dabad24#20170328102248#0001#1001", "Form_ID": "karte_no1", "Form_Na  
me": "カルテ 1 号紙  
"}, { "Custom_ID": "", "Data_ID": "", "Form_ID": "", "Form_Name": "" }, { "Custom_ID": "  
", "Data_ID": "", "Form_ID": "", "Form_Name": "" }, { "Custom_ID": "", "Data_ID": "", "For  
m_ID": "", "Form_Name": "" }, { "Custom_ID": "", "Data_ID": "", "Form_ID": "", "Form_N  
ame": "" }, { "Custom_ID": "", "Data_ID": "", "Form_ID": "", "Form_Name": "" }, {"Custo  
m_ID": "", "Data_ID": "", "Form_ID": "", "Form_Name": "" }, {"Custom_ID": "", "Data_ID  
": "", "Form_ID": "", "Form_Name": "" }, {"Custom_ID": "", "Data_ID": "", "Form_ID": "", "  
Form_Name": "" }, {"Custom_ID": "", "Data_ID": "", "Form_ID": "", "Form_Name": "" } ],  
"event": "print001", "time": "2017-03-28T10:22:49+0900", "user": "ormaster" }
```

print001 プラグインの設定が正しければアドビリーダーが起動されカスタマイズカルテ 1
が印刷される。

印刷されない場合は、push-exchanger のログを確認し、print001.yml の設定を見直す。

正常動作していれば帳票 API のレスポンスデータが

C:\Program Files (x86)\push-exchanger\tmp\print001-data\に作成されている。

また印刷に利用した PDF データが

C:\Program Files (x86)\push-exchanger\tmp\print001\に保存されている。

9. karte_no1 プラグインの解説

karte_no1 プラグインは print001 プラグインの帳票プラグインという形で実装されている。帳票プラグインの仕様は以下である。

- プラグイン名は帳票 ID(Form_ID)またはカスタム ID(Custom_ID)と一致すること
- クラス名はプラグイン名と一致させる(karte_no1 であれば Karte_no1 となる)
- 帳票作成処理を行う export メソッドを定義すること
 - export メソッドはファイル名、帳票データ(JSON)、オプションの 3 つの引数を受け取る
 - export メソッド内で帳票データを利用して引数のファイル名の PDF を作成する

帳票 API を利用した帳票データの取得、印刷処理は print001 プラグインで行うようになっている。

karte_no1 は monpe Ruby ライブラリを利用して monpe のテンプレート帳票ファイルを使用して PDF を作成する。

monpe Ruby ライブラリについては monpe Ruby ライブラリの仕様書を参照。

karte_no1 ディレクトリを参照すると以下のようになっている。

- HCM01.red
 - カルテ 1 号用紙頭書きの monpe 帳票ファイル
- HCM011.red
 - カルテ 1 号用紙続紙の monpe 帳票ファイル

- karte_no1.rb
 - プラグイン本体の Ruby スクリプト

karte_no1.rb の抜粋を以下に示す。Karte_no1 クラスに export メソッドが実装されていることがわかる。export メソッド内で帳票データを参照し、monpe Ruby ライブラリを用いてデータ埋め込みを行っている。

```
# coding : utf-8

require 'monpe'
require 'pp'
require_relative '../lib/orca-string'
require_relative '../lib/orca-wareki'

class Karte_no1

  FIRST_PAGE_DISEASE_ROWS = 8
  MIDDLE_PAGE_DISEASE_ROWS = 22
  RED_FILE = 'HCM01.red'
  RED_FILE2 = 'HCM011.red'

  def export(fname, form_data, opt)
    offset_set = opt[:pr_conf][:offset_settings]
    if offset_set
      offset_1 = offset_set[RED_FILE.to_sym]
      offset_n = offset_set[RED_FILE2.to_sym]
    else
      offset_1 = nil
      offset_n = nil
    end

    forms = []
    form_data['Forms'].each do |hash|
```

```

    @data = hash['data']
    if @data['Order_Class'] == '2'
      f = make_later_page
      f.set_offset(*offset_n) if offset_n
      forms << f
      next
    end

    ary = edit_ary
    f = make_1st_page(ary)
    f.set_offset(*offset_1) if offset_1
    forms << f
    start_row = FIRST_PAGE_DISEASE_ROWS
    while ary.size > start_row
      f = make_later_page(ary, start_row)
      f.set_offset(*offset_n) if offset_n
      forms << f
      start_row += MIDDLE_PAGE_DISEASE_ROWS
    end
  end

  Monpe.export(forms,fname)
end

```

<省略>

10. その他の帳票プラグイン

カルテ 1 号紙以外に以下帳票のサンプルを提供している。

詳細は帳票データ取得 API を参照

https://www.orca.med.or.jp/receipt/tec/push-api/report_data_api.html

- ・ 処方箋(院外)
- ・ 請求書兼領収書
- ・ 診療費明細書
- ・ お薬情報
- ・ お薬手帳
- ・ 予約票
- ・ 予約一覧
- ・ 予約患者一覧
- ・ 支払証明書
- ・ 診療録(カルテ 3 号紙)
- ・ 入院診療録(カルテ 1 号紙)
- ・ 退院証明書
- ・ 入院請求書兼領収書
- ・ 入院診療費明細書
- ・ 入院お薬情報
- ・ 入院お薬手帳
- ・ 入院注射箋
- ・ 入院指示箋
- ・ 入院診療録(カルテ 3 号紙)

11. push-exchanger を利用しないカスタマイズ帳票の作成

帳票 API を利用したカスタマイズ帳票の作成については現状においては push-exchanger を利用した実装例しかないが、他のプログラミング言語で独自に実装することが可能である。また帳票テンプレートとして monpe を挙げたがその他のライブラリやソフトを利用しても問題ない。

12. まとめ

- 帳票 API を利用して日レセクラウド・WebORCA 環境でもカスタマイズ帳票の作成が可能である
- push-exchanger と print001 プラグインを利用して医療機関の Windows 環境でカスタマイズ帳票の作成が可能である

- ただし必ずしも push-exchanger を利用する必要はなく独自プログラムでも同じ仕組みが利用できる

➤

13. 参考文献

- 日医標準レセプトソフト PUSH 通知仕様書
- 日レセ PUSH 通知駆動フレームワーク push-exchanger 仕様書
- monpe Ruby ライブラリ仕様書
- 帳票データ取得 API について
- 帳票データを作成するための日レセの設定について
- push-exchanger の設定について
- 画像データ取得 API
- オブジェクト指向スクリプト言語 Ruby <https://www.ruby-lang.org/ja/>