

Epson Connect API 概要

0. 目次

1. Epson Connect APIとは？
2. Epson Connect APIを利用するメリット
3. 利用例
4. 印刷までの流れ

この資料は、Epson Hack Trekの参加検討者にAPIの概要をお伝えするためのものです。より詳細な部分や、実装のためのサンプルコードなどは、利用規約に同意いただいた後に参加者にお配りいたします。ハッカソン開始前に資料を配布する予定であり、また、ハッカソン当日は担当者が質疑応答の対応をいたしますので、本APIを触ったことがない方もぜひ参加ご検討いただければと思います。

1. Epson Connect APIとは

エプソンのクラウドサービス「Epson Connect」と繋がっているエプソンプリンター・スキャナーに対して、外部から印刷等を制御・実行するための手段を提供するためのものです。離れたプリンターに対し、容易に手元のファイルなどを印刷させることを可能にします。

2. Epson Connect APIを利用するメリット

- 専用回線を用意せずとも、プリンターをFAXのように利用することができる
- Epson Connectにアップロードしたファイルは印刷後に消去されるため、ユーザーのプライバシーを保護できる
- 多くのクラウドサービスとの連携が容易にできる

3. 利用例

3.1 コミュニケーションツールとの連携

Slackなど、外部のAPIと連携することで、コミュニケーションツールのグループで投稿された内容を容易に印刷することができます。例えば、「印刷」という言葉を投稿したとき、そのチャットグループで一番新しいPDFや画像データを印刷するBOTを作成することもできます。

3.2 漫画やイラストなどのコンテンツの販売

Epson Connect APIでは、利用者の端末にデータを保存することなく印刷を実行することができます。例えば、漫画やイラストなどを購入すると、所定のプリンターが動き、購入した漫画やイラストが印刷されるWebアプリケーションを作成することも可能です。これにより、購入者に電子データを保存させずに作品を受け渡すことも可能になります。

3.3 イベントの名札の印刷

画像生成のWebアプリケーションにEpson Connect APIを組み込むと、参加者の端末で文字を打ち込んでもらうだけで完成資料を印刷することが実現されます。例えば、イベントの名札の配布シーンで利用できます。イベント用名札の背面画像とレイアウトをあらかじめ用意しておき、来場した参加者に名前を入力してもらうと、Epson Connect APIを通してその場で名札を作成することができます。

4. 印刷までの流れ

次の手順でAPIを利用すると、APIを用いて印刷を実行できます。

1. 認証
2. ジョブ作成
3. 画像アップロード
4. 実行

4.1 認証

印刷するためには、まずはじめに、各認証を行い、APIを利用するためのアクセストークンを発行します。

4.1.1 リクエスト

メソッド:POST

- http-header:
 - Authorization (必須) : クライアントIDとクライアントシークレットをBasic認証形式でエンコードしたもの
 - Content-Type (必須) : application/x-www-form-urlencoded; charset=utf-8

クエリストリング:

- subject (必須) : printerの指定

HTTPボディ:

- grant_type (必須) : “password”を指定してください
- username (必須) : アクセストークンを発行したいプリンターのメールアドレスを指定してください
- password (必須) : usernameで指定したプリンターの所有者メールアドレス。プリンターの所有者メールアドレスによる認証が不要な場合は”(空文字)を指定してください。

4.1.2 レスポンス

```
{
  "access_token" : value1,
  "token_type" : "Bearer",
  "expires_in" : value2,
  "subject_id" : value3,
  "subject_type" : ""
}
```

- access_token(文字列): 再発行したアクセストークン
- token_type(文字列): "Bearer" 固定
- expires_in(数値): 再発行したアクセストークンの有効期限(秒) “3600”(1 時間)固定
- subject_id(文字列): プリンターを識別するプリンターID

4.2 ジョブ作成

印刷をする準備を行います。

4.2.1 リクエスト

メソッド:POST

- http-header:
 - Authorization (必須) : Bearer 形式のアクセストークン
 - Content-Type (必須) : application/x-www-form-urlencoded; charset=utf-8

クエリストリング:

- なし

HTTPボディ:

- job_name (必須) : ジョブを識別するための名前(印刷対象のファイル名など) 最大256文字
- print_mode (必須) : 印刷モード (“document”もしくは“photo”を指定 document・・・文書(Office 文書、PDF)を印刷する場合 photo・・・画像(jpeg)を印刷する場合)

4.2.2 レスポンス

```
{
  "id" : "value1",
  "upload_uri" : "value2"
}
```

- id:作成した印刷ジョブの印刷ジョブ ID
- upload_uri:印刷するファイルをアップロードする URI

4.3 画像アップロード

4.2 ジョブ作成 で取得したURLに向けて、画像アップロードを行います。

4.3.1 リクエスト

メソッド:POST

- http-header:
 - Content-Length (必須) : 送信ファイルサイズを指定します (単位: バイト)。
 - Content-Type (必須) : 送信するファイル種別に応じた MIME-Type。image/jpeg または application/octet-stream

クエリストリング:

- Key (必須) : 印刷ジョブ作成APIで、Keyまで指定したURIを返却します。
- File (必須) : アップロードするファイル名。"1.[拡張子]"で指定します。"sample.pdf"というファイル名の場合、"1.pdf"を指定下さい。

HTTPボディ:

- 送信するファイルデータ

4.3.2 レスポンス

```
{
}
```

4.4 実行

4.2 ジョブ作成で取得したIDを利用し、4.3でアップロードしたファイルを印刷する。

4.4.1 リクエスト

URL中に組み込む必要のあるパラメータ

- printer_id: 印刷実行したいプリンターのプリンターID
- job_id: 印刷実行したい印刷ジョブ ID 印刷ジョブ作成で返却した印刷ジョブ ID を指定します

メソッド:POST

- http-header:
 - Authorization (必須) : Bearer形式のアクセストークン
 - Content-Type (必須) : application/x-www-form-urlencoded; charset=utf-8

クエリストリング:

- なし

HTTPボディ:

- なし

4.4.2 レスポンス

```
{  
}
```