国立情報学研究所教育研修事業
「大学図書館員のための IT 総合研修」2020 年度
「Web API を使ったデータの入手とその整備」講義資料

大学図書館関係の Web API を実際に使ってみる

2020 年 9 月

東京大学情報システム部　前田朗

目次

1　この資料について

1.1　大学図書館員がWeb APIを使うには

Web APIは「生もの」である。提供されるサービスも、その仕様も変わりゆく。大学図書館員が本務の片手間で使うのであれば、個々のWeb APIの詳細を覚えるよりも、「詳細をよくわかっていなくても、なんとなく使えてしまう」ほうが実用的であろう。この講習ではそのことを踏まえ、基本的な考え方を身に着け、Web APIに慣れてもらうことを目指している。

1.2　プログラミングなしでWeb APIの活用を行う

Web APIの「API」は「Application Interface」の略語である。つまり、人間がインタラクティブに操作するのではなく、アプリケーションが機械的に処理するための機能提供であることを示している。

アプリケーションの開発によりWeb APIを自在に活用するには、プログラミングが必須となり技術的に敷居が高い。そこでWeb APIの理解をめざす大学図書館員のために、プログラミングをせずにWeb APIを利用する方法を示した。

2　利用前の準備

Web API はあくまで手段であり、自分が実現したいことと合わせて利用を考えるべきものである。まずは気になる Web API を探すところから始める。その際には利用申請や技術的な難易度なども考慮するとよい。

2.1　使用する Web API を探す

図書館関係の Web API は数多くあるが、個々のサービスごとに提供内容や仕様が異なる。まずは利用案内を確認する。

実習 2.1-1:　別紙「図書館関係 Web API レビュー 2020」で Web API を通覧する

> **ポイント：　Web API の提供元を選ぶ際に**
> ・使いたいデータがありそうかどうかで、まずは選定する。
> ・Web API は移り変わりが激しく、新規サービス提供やサービス終了がよく起こりうる。別紙一覧以外にも探してほしい。

実習 2.1-2：　国立国会図書館サーチの Web API 利用案内（以下）を確認する

https://iss.ndl.go.jp/information/api/

> **ポイント：　Web API 利用案内の重点確認事項**
> ・データの収録範囲から、使いたいデータがあるか確認。
> ・利用条件に問題がないかを確認する。
> - 有料か無料か
> - アクセス回数などの制約がないか
> - 対象は個人か組織か

実習 2.1-3：　国立国会図書館サーチの Web API 仕様（以下）を確認する

https://iss.ndl.go.jp/information/api/riyou/

ポイント：　Web API への難易度判断

・OpenSearch や SRU といった、URL で検索パラメータ指定をする方式(Web の GET メソッド)であればデータ取得の難易度は低い。一方、Web の SRW や、認証に手数がかかる OAuth は扱いが難しい。

・OpenSearch が出力する、RSS/Atom のデータ構造と語彙は比較的単純。

・採用するメタデータスキーマによって、取得データ解読の難易度が異なる。

・SPQRQL によるデータ取得は、対象データの語彙や構造の事前確認が必要であり難易度が高めに思えた。

2.2　Web API の利用申請

CiNii Web API を例に、Web API の利用申請を実際に体験してみる。

実習 2.2-1：　CiNii Web API の利用案内を確認する

https://support.nii.ac.jp/ja/cinii/api/api_outline

実習 2.2-2：　CiNii Web API の利用申請を実際に行ってみる

https://api.ci.nii.ac.jp/ja/

3　とりあえず試してみる

3.1　サンプルリクエスト活用のススメ

Web API のサービス提供元のドキュメントにサンプルリクエストがあれば、ぜひ実際に試してほしい。

実習 3.1：　国立国会図書館サーチの Web API 仕様から、OpenSearch のリクエスト例を確認する

https://iss.ndl.go.jp/information/wp-content/uploads/2020/03/ndlsearch_api_ap3_20200302_jp.pdf

⇒　以下のリクエスト例が見つかるはず。

https://iss.ndl.go.jp/api/opensearch?title=%e3%81%93%e3%81%93%e3%82%8d&creator=%e5%a4%8f%e7%9b%ae%e6%bc%b1%e7%9f%b3&from=2011&until=2012

ポイント：　サンプルコードは理解への近道

- ドキュメント中にサンプルがあればそれを使う。
- 国立国会図書館サーチのサンプルはそのまま使えるものであったが、中にはリクエストの起点となる URL の補完や、アプリケーションキーの取得と指定が必要なこともある。

3.2　Web ブラウザから Web API を実行する

Web ブラウザを使うことで、Web API を容易に試すことができる。

実習 3.2-1：　直前の操作(実習 3.　1)で確認したサンプルリクエストを、Web ブラウザの URL 入力欄に張り付けて、Web API のデータ取得を確認する

ポイント：　URI エンコードと Web ブラウザ(1)

- サンプルリクエスト URL の%だらけの文字列を見ただけでは何をしているか分かりかねるが、これは日本語や記号等の文字が URL 用に変換（URI エンコード）されているためである。
- Web ブラウザによっては、このリクエストを URL 入力欄に張り付けたときに、URI エンコード前の文字を表示してくれる。

ポイント：　Web API 出力結果の確認

・階層構造になっていることが多い。
・情報検索の場合、アイテムの繰り返しの箇所があるはず。
・個々のアイテムはタグでメタデータ項目を確認できる。

ポイント：　Web API と Web ブラウザの相性

・Web API と Web ブラウザには相性がある。画面が崩れて表示されるときは、別の Web ブラウザアプリケーションでも試してみること。
・JSON 形式の場合は、Firefox であれば構造を解析し表示してくれる。
・Web ブラウザ上で表示されず、直接 XML ファイルとしてダウンロードされることもある。その場合は、ファイルをエディタ等で内容を確認できる。
・Web API によっては、Web ブラウザからのアクセスと判断し、HTML で結果を返すものもある。その場合は、後述の OS の標準コマンドによる取得を試みること。

実習 3.2-2：　直前の操作(実習 3.2-1)の検索キーワードを変更する

ポイント：　URL 上でのパラメータの渡し方

① 個々のパラメータは「項目名=値」のスタイルで設定する。使わないパラメータは設定不要。(Web ブラウザからのアクセスは問題ない（自動処理される）が、本来はパラメータの値の URI エンコードが必要)
② 個々のパラメータを’&’で接続しパラメータリストに
③ 起点の URL（BaseURL や Endpoint） + ‘?’ + パラメータリスト

ポイント：　情報検索系 Web API の種類と検索パラメータ

・説明したサンプルは OpenSearch によるデータ取得である。URL 上で検索パラメータを指定する。
・Web API には情報検索で使う SRW など URL でパラメータを渡さない方式もある。
・国立国会図書館サーチでは、OpenSearch 以外に SRU もサポートしている。検索条件の指定や結果データの形式が異なるが、URL でパラメータを渡し、データを取得するといった基本は変わらない。

ポイント：　URI エンコードと Web ブラウザ(2)

・パラメータ中の日本語はそのままで（URI エンコードしなくて）よい。Web ブラウザがサーバに送るときに自動で変換処理してくれる。

・URL 入力欄の文字列をコピーしテキストエディタに張り付けると、URI エンコード後のリクエスト内容が確認できる。

実習 3.2-3：　CiNii Books の以下の説明をみて、タイトル「こころ」、著者名「夏目漱石」で検索リクエストを作成・実行する

https://support.nii.ac.jp/ja/cib/api/b_opensearch

ポイント：　情報検索の Web API でよくあるパラメータ指定

・情報検索系のサービスの場合は、一般に以下の条件設定となる。

- 検索条件
- 取得件数
- 開始ページ
- 取得フォーマット

・アイテムのユニークキーを使い特定の 1 件だけ取得するものもある。

例: CiNii Books（xxxx には個人の CiNii appid をセットする）

https://ci.nii.ac.jp/ncid/BB12082218.json?appid=*xxxx*

実習 3.2-4：　国立国会図書館サーチと CiNii Books のデータ取得結果のデータ形式を比較する

ポイント：　出力フォーマットのドキュメントは必要に応じて確認する

・Web API の取得データのデータ項目は、その項目名から内容を推測できることが多い。

・OpenSearch には title, creator などの規定の語彙がある。

実習 3.2-5：　CiNii Books のデータ取得結果のフォーマットをデフォルトの Atom から JSON に変更し、結果をみる

実習 3.2-6：　Web ブラウザの Web ページ保存機能により、PC にファイルとして取り込む

4　メタデータの全件取得

4.1　Web API による検索結果のページ送り

OpenSearch など情報検索用の Web API は一度に取得できるアイテム数に上限がある。検索結果を全件取得するには、Web 画面からの閲覧と同様に次ページのデータを取得することを繰り返す必要がある。

実習 4.1:　CiNii Books の以下の説明をみて、1 回のリクエストの取得件数上限と、ページ切り替えのオプションがあることを確認する

https://support.nii.ac.jp/ja/cib/api/b_opensearch

4.2　OAI-PMH

OAI-PMH はデータの全件及び差分取得を想定している Web API のプロトコルである。機関リポジトリのシステムで標準的に使われている。

実習 4.2-1：　雑誌記事索引の OAI-PMH データ取得を Web ブラウザで行う

http://iss.ndl.go.jp/api/oaipmh?verb=ListRecords&metadataPrefix=dcndl_simple&set=zassaku&from=2018-08-24&to=2018-08-24

※　「国立国会図書館サーチが提供する OAI-PMH」（以下）のサンプルより
https://iss.ndl.go.jp/information/api/api-lists/oai-pmh_info/

実習 4.2-2：　直前の操作(実習 4.2-1)の Web API リクエストをコマンドで行う

【Windows - コマンドプロンプト】

① Windows のコマンドプロンプトを起動する

② 次のコマンドを投入し、powershell モードに切り替える

```
powershell
```

③ Invoke コマンドを投入する。パラメータ指定は次のとおり。

```
Invoke-WebRequest “any_url” -OutFile outputfile.txt
```

④ 作業ディレクトに outputfile.txt としてデータが取得できる

【Machintosh - ターミナル】

参照:「curl 入門(mac 編)」
https://qiita.com/kaizen_nagoya/items/f13df3e2c9fe6c3bf6fc

① Machintosh のターミナルを起動する

② curl コマンドをコマンドでインストールする（未インストールの場合）

```
brew install curl
```

③ curl コマンドを投入する。パラメータ指定は次のとおり。

```
curl -O “any_url”
```

④ 作業ディレクトにデータが取得できる

ポイント：　Web ブラウザで Web API が HTML として保存されるときは

・コマンドを使うことでその問題を回避できる。

・事前の URI エンコードが必要になるが、Web ブラウザの URL 入力欄にエンコード前の URL を張り付け、それをまたコピーするテクニックがある。

補足：　Windows 版 curl について

・Windows に curl をインストールして使うこともできる。

https://curl.haxx.se/download.html

実習 4.2-3：　直前の操作（実習 4.2-1）の取得結果をテキストエディタで開き、メタデータと resunptionToken の確認を行う

補足：　Web API 取得結果表示には高機能テキストエディタを

・　Windows 標準のメモ帳と異なり、高機能なテキストエディタ(Mery など)では、XML のタグを色分けで表示する機能がついている。

・プログラマ向けのエディタとなるが **Atom エディタ**は、XML や JSON のデータを細かく色分けして表示してくれる。

・テキストエディタの正規表現などを活用すれば、テキストエディタ内でデータのチェックをある程度行える。

実習 4.2-4：　直前の操作(実習 4.2-3)で確認した resunptionToken から、以下の書式の URL を作成し、続きのデータを取得する

http://iss.ndl.go.jp/api/oaipmh?verb=ListRecords&resumptionToken=*xxxx*

※　*xxxx* には取得した resunptionToken の値をセットする

補足：　JAIRO Cloud のデータ取得

- JAIRO Cloud は OAI-PMH のアクセス制限していないので、自機関のメタデータを取得できる。Web API リクエストの起点である BaseURL も「**リポジトリのトップ URL + /oai**」で固定であり、簡単に確認できる。
- NDL サーチと異なり、Web ブラウザにも XML 形式で値を返すため、100 件ずつのデータ取得となり手間はかかるが Web ブラウザのみでデータ取得も可能である。

補足：　ジャパンサーチの検索結果の全取得

- OAI-PMH とは別であるが、ジャパンサーチは検索結果を 2000 件ずつ取得するためのオプションがある。
- 次の 2000 件の情報を取得するには、**scrollId** を使ってアクセスする。考え方は OAI-PMH の resumptionToken と同じ。
- 参照:　「簡易 Web API 概説 - ジャパンサーチ」
  https://jpsearch.go.jp/static/developer/webapi/

補足：　Web API 以外の文献データ取得手段

- 検索結果のファイル出力機能。
- オープンデータ。

検索機能の提供やリアルタイム性はないが、Web API ではなくファイルとして公開されているデータを使うことも考慮されたい。

- CiNii Books
  https://www.nii.ac.jp/CAT-ILL/about/infocat/od/index.html
- ERDB-JP
  https://erdb-jp.nii.ac.jp/ja/exports

5　SPARQL によるデータ取得

SPARQL はリンクトデータ(RDF)に対して使われる検索式である。SPARQL Endpoint を提供しているサービスであれば、Web API によるリクエストができる。この講習では、Web ブラウザ上での SPARQL の実行についてさわりだけ紹介する。データの取得や何ができるかの試しであれば Web アプリケーションの利用だけでよい。OpenSearch と異なり検索結果の全件取得が容易である。

5.1　SPARQL を Web アプリから体験する

実習 5.1-1：　NDL Authorities SPARQL Endpoint（以下）にアクセスする

https://id.ndl.go.jp/auth/ndla/?query=

実習 5.1-2：　国立国会図書館件名標目で「図書館」の下位の件名をすべて調べる

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?sub ?label WHERE {
  ?top rdfs:label "図書館" ;
  skos:narrower ?sub .
?sub rdfs:label ?label .
}
```

実習 5.1-3：Web API として使うときにどのようなパラメータを送っているかを確認する。 出力形式を HTML から XML や JSON に変更し Web ブラウザの URL 欄の表示をみればよい。

補足：　SPARQL 関係の雑情報

- 検索の自由度が高い。階層構造を指定してマッチングできる。FILTER によるパターンマッチも備えるが、重い処理をさせるとタイムオーバーになりかねない。
- 対象の RDF がどのようなグラフ構造と語彙を持っているかを理解が必要。
- SPARQL Endpoint がない RDF でも、fuseki を使えば、ローカルの PC 内で SPAQL の実行が可能。
- OpenRefine の RDF Refine 拡張を使えば、SPARQL を使ったデータの照合ができる。

6　Web API 取得データのスプレッドシート化

Web API 取得データは、XML や JSON といった形式になっている。それをより人間が扱いやすいスプレッドシート形式に変換する手法をいくつか示す。

6.1　OpenRefine

［説明資料］

利用法は別紙「図書館 Web API のための OpenRefine 活用方(1)」で説明する。

> 補足：　OpenRefine でスプレッドシート化できないデータ
> ・OpenRefine ではうまくレコード単位で取り込めるメタデータとそうでないメタデータがある。
> ・たとえば、NDL サーチや CiNii の RSS/Atom の取得結果は、レコード単位の扱いができない。
> ・同じ CiNii でも JSON 形式や、雑誌記事索引。JAIRO Cloud の OAI-PMH 出力、ジャパンサーチの Scroll であれば動作した。Web API の取得結果の確認や名寄せ対象の確認にはこのままでも十分。

6.2　Web ブラウザで使える文献 Web API 取得結果のスプレッドシート化

利用には Google アカウントが必要なため、講師デモのみ。OpenRefine に取り込む際は、事前にエディタ等で UTF-8 の BOM を「なし」にすること。

［説明スライド］

https://www.slideshare.net/genroku/webweb-api-google-colab-235981593

［前準備］

① Google Colaboratory (Google Colab)にアクセス

https://colab.research.google.com/

※ Google アカウントがあれば即利用可

② 「ファイル」⇒「ノートブックを開く」⇒「GitHub」を選択

③ 「GitHub URL を入力するか、組織またはユーザーで検索します」の欄に、次の Google Colab 用コードの入手元（前田朗の GitHub）を入力する

https://github.com/maedaak/

④ 「**リポジトリ**」プルダウンから以下を選択し、パスに表示された変換プログラムを取り込む

- **ndl_search2csv4GoogleColab**　（国立国会図書館 OpenSearch 用）
- **ndlsearch_oai2csv4GoogleColab**　（国立国会図書館 OAI-PMH 用）
- **cinii_articles_json2csv4GoogleColab**（Cinii Articles JSON 用）
- **junii22csv4GoogleColab**　（OAI-PMH junii2 用）

［利用法］

① 利用する Web API からデータを取得し、ファイルに保存する
② Google Colab にアクセス
③ 「**ファイル**」⇒「**ノートブックを開く**」から、①のデータに対応した前準備で取り込んだスクリプトを開く
④ プログラムがいくつかのコードに分かれて表示される。各コードブロックの左にある実行ボタン（▷）を先頭から順にクリックする
⑤ ファイルの取り込みの処理のブロックで、①のファイルを指定する
⑥ コードブロックを最後まで実行すると、処理結果の CSV が PC にダウンロードされる

ポイント：　Web API を自在に使うには

- 講習の範囲から外れるがプログラムを組めれば自在に扱える
- メタデータスキーマの解析のプログラムは作成に手のかかる箇所なため、公開されているプログラムがあればその活用を。

6.3　NDL 書誌データ取得シート

利用には Excel 環境が必要。講師デモのみ。検索パラメータの指定が可能であり、非常に簡単に使うことができる。OpenRefine には処理結果の Excel をそのまま取り込むことができなかったが、TSV に変換すれば可能であった。

http://www.slis.doshisha.ac.jp/~ushi/ToolNDL/

［前準備］

Excel が実行できる環境で、上記のサイトからファイルをダウンロードする

［利用法］

ダウンロードしたExcelファイルを開き、検索条件を指定し実行すると、Excelシート上にWeb API取得結果が得られる。