


機械利用可能なインターフェースの整備

林 賢紀*

図書館分野では所蔵資料の発見可能性を向上させるため総合目録等の構築が行われてきたが、WebAPI の活用によりメタデータの収集や提供が容易に、かつ効率的に行えるようになった。本稿では、Web サービス同士を連携させるための API である WebAPI について、図書館分野での利用を例にその目的や利点を紹介した。さらに、標準的に利用されている WebAPI やデータフォーマットの利用による相互運用性の確保や、利用者向けドキュメントの整備と公開による利便性の確保など、WebAPI の整備にあたり設計プロセスや運用時の留意事項を解説した。

キーワード：図書館システム、WebAPI、システム開発、Web サービス、データ連携

 本稿は、クリエイティブ・コモンズ表示 4.0 国際 (CC BY 4.0) ライセンスの下に提供する (<https://creativecommons.org/licenses/by/4.0/deed.ja>)。

1. はじめに

1.1 WebAPI とは

まず、API (Application Program Interface) について整理しておきたい。一般に API とは、ソフトウェアやプログラム、Web サービスを相互に連携させるための定義と手順 (プロトコル) を指す。プログラム、いわば機械同士が連携して情報のやり取りを円滑に行うためのものである。一方がある方法で要求を送信し、相手は指定された方法で応答を返す。このとき、要求の送信 (リクエスト) や応答 (レスポンス) を返す手順や手段は共通化されている。例を図 1 に示す。

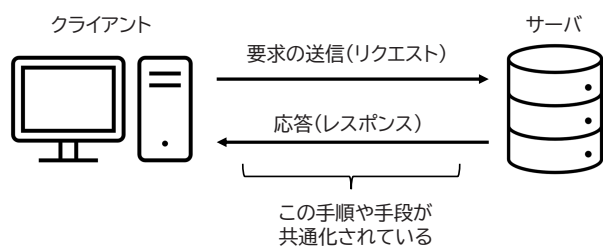


図 1 API の概要

このうち、Web サービス同士を連携させるための API は WebAPI と呼ばれる。多くの WebAPI は REST (REpresentational State Transfer) と呼ばれる原則で構築されている。REST では、Web で利用される通信プロトコルである HTTP を利用してリクエストが送信され、

相手からレスポンスとしてデータが戻ってくる。リクエストは、必要なリソースを URI で記述することにより送信され、HTML、JSON、XML、CSV といった形式のデータがレスポンスとして返される。

リクエストを送るためには Web で通信ができるアプリケーション、たとえば Web ブラウザがあればよく、受け取ったデータはこれらのデータ形式が処理できるアプリケーションで取り扱える。非常にシンプルな仕組みである。

1.2 図書館における WebAPI の利用

現在では、多くの情報が Web により検索ができる。Web の登場以前でも、多くの図書館では OPAC (On-line Public Access Catalog) として自館内の端末で所蔵情報や書誌情報を検索することができたが、この OPAC を Web から利用可能とすることで、インターネットに接続されていればだれでも、どこでも検索ができるようになった。

加えて、WebAPI により、検索した情報を Web ブラウザだけでなく、他のプログラム等からの機械的な取得ができるようになった。WebAPI は学術情報の分野でも以前から利用されており、Web サイトの更新を通知する RSS (RDF Site Summary) により、新着の学術雑誌の目次情報で配信し、RSS リーダーや他のサービスへの取り込みに利用されてきた。また、機関リポジトリなどからメタデータの収集を行うプロトコルとして OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting) が 2002 年に公開¹⁾されて以来、現在も全世界のリポジトリなどで利用されている。

日本国内の図書館では、2006 年頃には RSS を利用した情報の配信が行われた。RSS は、Web サイトの新着を通知する仕組みで、これを利用して新規に受け入れした資料の情報配信が一橋大学附属図書館や農林水産研究情報センター (当時。現在の農林水産研究情報総合センター) にて行われた²⁾。国立国会図書館は、デジタルアーカイブポー

*はやし たかのり 国際農林水産業研究センター 情報広報室 広報資料科

〒305-8686 茨城県つくば市大わし 1-1

E-mail: hayashit0170@jircas.go.jp

 <https://orcid.org/0000-0002-5189-1865>

(原稿受領 2024.11.19)

タル「PORTA」を2007年に一般公開したが、「外部提供インタフェース」として検索結果の提供と、検索結果のメタデータをOAI-PMHで提供する2つのWebAPIを提供していた³⁾。PORTAの後継サービスにあたる国立国会図書館サーチ（NDLサーチ）では、SRU、OpenSearch、OpenURLに対応した検索用のWebAPIを有するほか、OAI-PMHによるメタデータの提供を行っている。また、国内の大学等の機関リポジトリやデジタルアーカイブなど、さまざまな機関が提供するメタデータをOAI-PMHで収集し、NDLサーチで検索可能としている³⁾。国立情報学研究所が提供する学術情報サービスCiNiiも、2009年よりWebAPIを公開しOpenSearchによる検索やRDFを用いたメタデータの提供を始めている⁴⁾。

このように、図書館などの学術機関においても、所有する情報をWebAPI経由で検索可能にする、あるいはメタデータを提供する・収集することが長年行われてきた^{注1)}。

以下、本稿ではWebサービス同士を連携させるためのAPIであるWebAPIについて、コンテンツを提供する側がなぜ公開するのか、またどのように構築すればいいか、その概要を解説する。

2. WebAPIを公開する目的、利点

先に紹介した、国立国会図書館や国立情報学研究所など多くの情報機関では、なぜWebAPIを公開しているのだろうか。Webの登場以前に遡って検討したい。

2.1 WebAPIによらないデータの交換

図書館での情報サービスは自館で収集し所蔵している資料を基に行われる。しかし、収集できる資料には限りがある。

自館にない資料は、相互貸借や複写依頼等の相互利用により他館より取り寄せて利用者に提供してきた。このような相互利用を円滑に行うため、各図書館の所蔵目録を取りまとめた総合目録が作成されてきた。自館の資料が総合目録に掲載されれば、資料の発見可能性を高め利用につなげることができる。

このように、情報検索の基盤構築の方法として「散在する情報源から情報を一か所に集約し参照可能とする」ことが長年行われてきた。

では、どのようにして情報を集約し、また利用可能にするのだろうか。

現在でも、国立情報学研究所が運用するデータベースに参加館が所蔵する書誌と所在情報を登録する形でNACSIS-CATは構築され、NACSIS-CAT/ILLでの情報検索や相互利用、またCiNii Booksなどで誰でも検索が行える。各館で持つ図書館システム（クライアント）とNACSIS-CAT（サーバ）は、相互のメッセージ交換方式を規定したプロトコルであるCATP（Cataloging information Access & Transfer Protocol）で接続され、これに基づいてデータが送受信される。CATPに対応したシステムさえあれば、クライアントから書誌所在情報を送

信してNACSIS-CATに登録しCiNii Booksなどで検索を行うことができ、逆にサーバから書誌情報をダウンロードして、他の図書館で登録された情報を自館の目録に利用できる。

CATPに対応した図書館システムを持つ図書館間では、このようにしてNACSIS-CAT上のデータが利用できる。一方、それ以外の機関や個人では利用できるだろうか。利用資格や手続きは別としても、CATPに対応した図書館システムは数が多くなく、その導入や運用には一定のコストを要する。また、CATPは書誌所在情報の検索から情報の取得、また登録に対応したプロトコルで、たとえば書誌情報の利用のみを目的として対応したシステムを導入することは費用対効果から見て適切ではないだろう。さらに、システムから返されるデータはCATPで定義された独自の形式で記述されており、利用目的に合わせて変換する必要がある。

このため、たとえばNACSIS-CAT上の書誌情報を自らのシステムに取り込んで使いたいと考えても、直接利用することは難しいといえる。

2.2 WebAPIの活用と利点

図書館などの機関で作成された信頼性の高いデータを容易に利用できれば、自らこれらのデータを整備する必要がなくなり、利用する側の負担は少なくなる。また、CSV形式など汎用的なデータ形式で提供されれば、特別なシステムを用意することなく低コストで利用できる。

近年では、大量のデータを機械学習技術で処理し、学習したデータに基づく情報の推測や、生成AIなどのコンテンツ生成が行えるようになった。これらの処理のためにも、学習用のデータを機械的に取得できることが求められている。加えて、WebAPIを利用せず、Webページから特定の情報を機械的に抽出する「ページスクレイピング」と呼ばれる技術により利用された場合、その頻度や回数などによっては利用者が意図せずにサーバに負荷を与えてしまい、トラブルになることも考えられる^{注2)}。

そこでWebAPIである。WebAPIを利用することで、シンプルな方法でデータを提供することができる。また、データを利用する側も、利用しやすい形でデータを受け取ることができ、双方にとって利便性は高い。

CiNiiでもWebAPIを提供しているが、設計に携わった大向氏は「学術情報サービスに求められる多様なニーズに応えるための手段」としてWebAPIは有用であるとし、「サービスとデータの分離の実現」「データの活用を促進する手段」としている⁵⁾。

CiNii ResearchはいくつかのWebAPIを有しているが、たとえば表1のURIにより、JSON-LD形式のデータを取得できる。データを示すURIに.jsonを付与するだけでよい。あとはJSON-LDを処理できるプログラム等に引き渡して処理を行う。

表1 CiNii Research からJSON-LD でデータを取得する URI

```
https://cir.nii.ac.jp/crid/<crid>.json
<crid> は対象データの ID
例: https://cir.nii.ac.jp/crid/1130000795379405696.json
```

多くの機関リポジトリはWebAPIとしてOAI-PMHを備え、他のサービスにメタデータを提供することができる。日本国内の機関リポジトリでは、国立情報学研究所が運用する学術機関リポジトリデータベース（IRDB）が、OAI-PMHによりメタデータを定期的に収集（ハーベスト）している。さらに、IRDBが各所にメタデータを提供するハブとなり、CiNiiや国立国会図書館等に提供されそれぞれで検索できるほか、ジャパンリンクセンター（JaLC）に提供されてDOIの取得が行われる。多くの場合、OAI-PMHによるデータの収集は毎日、毎週など定期的に行われ、前回の収集以後に追加や更新されたメタデータを自動的に収集する。メタデータの全件あるいは差分を手動で把握し送付するのに比較して、効率的にメタデータを提供することが可能となった。

表2にOAI-PMHによりIDを指定してメタデータ1件を取得するリクエストURLの例を示す。OAI-PMHでは、特定のデータ1件か、更新日の範囲やデータセットを指定して複数件のデータを取得できる。

表2 OAI-PMHでメタデータを取得するリクエストURLの例

```
https://jircas.repo.nii.ac.jp/oai?verb=GetRecord&metadataPrefix=jpcoar_1.0&identifier=oai:jircas.repo.nii.ac.jp:02000087
```

この例では、jircas.repo.nii.ac.jpから、特定のメタデータレコード（verb=GetRecord）としてidentifierがoai:jircas.repo.nii.ac.jp:02000087のメタデータをjpcoar_1.0の形式（metadataPrefix）で取得することを表す。

このように、多くのWebAPIが「メタデータを提供して他で利用可能とする」「データの活用を促進する」ために公開され、実際に利用されている^{注3)}。

3. WebAPIの構築・設計

3.1 概要

本項では、図書館において自館の所蔵資料の情報を利用者向けにWebAPIで提供するケースを元に、構築や設計を行うポイントを検討する。

APIの整備にあたっての参考資料として、デジタル庁では「デジタル社会推進実践ガイドブック」の一つとして「API導入実践ガイドブック」（以下、「ガイドブック」という。）⁶⁾をまとめている。このガイドブックでは、API設計の手順として以下を解説とともに示している。本項では、この手順に基づき検討を行う。

- 1) 要件の明確化
- 2) 標準的なAPIの調査
- 3) 公開データ項目の検討

- 4) データ項目定義
- 5) 関連主要APIとの関連性の整理
- 6) 関係ドキュメントの整備

3.2 要件の明確化

「ガイドブック」では、「利用データの整理およびその活用イメージを作り（略）データ抽出、統合、加工等の一連の流れやセキュリティ、アクセス頻度、提供条件等、APIに求められる要件を整理し明確化します」とある。なお、セキュリティについては別項で述べる。

まず、求められる要件を整理する。自館の情報をどこに、だれに提供したいか、またどのように利用されたいかによって選定するWebAPIも異なる。自館の所蔵資料の情報が集約され検索できることで利用者の利便性が向上することは、先に述べたとおりである。そこで、自館の情報をWebAPI経由でNDLサーチに提供すれば、NDLサーチで検索ができるようになり利便性が高まることが期待できる。まずは「NDLサーチで自館の所蔵を検索可能とする」が要件となろう。提供するデータは、書誌情報および所蔵情報が考えられる。これらをどのようにして提供するかは、次節以降で検討する。

3.3 標準的なAPIの調査

すでに標準的に利用されているWebAPIを利用することで、開発・構築のほか利用者が利用する際にかかるコストを削減することができる。先に明確にした要件に従い、WebAPIを選定することになる。

「WebAPIによるシステム連携ガイドライン」⁷⁾によれば、NDLサーチとの連携が可能なWebAPIとしては書誌情報等のメタデータを提供する目的でOAI-PMHあるいはRSSが、また書誌情報等の詳細ページへ直接リンクするためにOpenURLに対応するとある。いずれもデータの提供や、書誌情報を元にしたリンクに標準的に利用されているWebAPIであり、その仕様や実例も多く公開されている。

いずれのWebAPIでもNDLサーチに対して書誌情報を提供できる。しかし、RSSはその仕組み上、追加や更新となったデータの差分を送信することはできるが「このデータは削除された」ことを指定する方法を持たない。全件のデータを再度提供し入れ替えることで対応ができるが、出力や更新にも負荷がかかることが想定される。このため「WebAPIによるシステム連携ガイドライン」では、削除データを取り扱うことができる「OAI-PMHによるハーベスティングを可能にすることを強く推奨する」としている。

3.4 公開データ項目の検討

データの相互運用性を保つため、WebAPIで取得できるデータのフォーマットや項目は、国際標準やデファクトとなっている標準を活用する。書誌情報の記述項目や方法はDublin CoreやMARC21/MARCXML, NACSIS-

CATP などがあるが、NDL サーチでは Dublin Core を拡張した DC-NDL (国立国会図書館ダブリンコアメタデータ記述) あるいは OAI-PMH で標準的に対応する必要がある SimpleDC (OAI-DC) が利用できる。SimpleDC は Dublin Core の 15 の要素がそのまま用いられており、必要最小限のメタデータを簡易に記述できる一方、細かな情報の属性を保ったまま記述しにくい。たとえば「作成者の所属」を表現したくとも所属を表す要素を持たないため、この情報が失われてしまう。

自館が持つ情報をすべて DC-NDL に対応させる必要はなく、「必須」「あれば必須」などの推奨度合をもとに取捨選択して対応するのが適切である。自館で独自に定義した項目など DC-NDL で直接には対応していない場合も、適切な項目に対応づけるほか「連携に際してその情報が必要か」を検討する必要がある。たとえば「貸出中」「整理中」「製本中」などリアルタイムに更新される情報は、直接にその情報を提供するのではなく、書誌詳細画面などその情報が確認できる URI を提供して利用者に参照させるなどの方法も考えられる。

3.5 データ項目定義

データの項目名は名詞を基本とし、複数形での記載が推奨される。人が見たときにわかりやすいことが前提であり、意味や定義が曖昧であると他のサービスで利用する際に不備が起きる可能性がある。また、日付、時刻などは国際標準による記述方法が推奨される。使用しているコード類も同様で、標準的に利用できるコードがあれば利用すべきであろう。採用したこれらの記述方法を入力規約としてまとめ、API 仕様書のようなドキュメントとして公表することで、WebAPI を利用する者の助けとなる。

NDL サーチとの連携を行う場合は DC-NDL の利用を前提としており、入力規約は DC-NDL のものに従うことになる。

3.6 関連主要 API との関連性の整理

公開されている WebAPI は、独自のデータ項目の追加その他の機能拡張が行われることがある。また、他のサービスや WebAPI の仕様変更の影響を受けることがある。たとえば、NDL サーチにおいても、細部の調整はもちろんであるが利用頻度の低下などにより Z39.50 など従来提供していた API の提供を中止した例がある。また、DC-NDL も改定されることがある。もちろん、OAI-PMH そのものにも変更が生じることも考えられる。

従って、連携先の WebAPI の変更に伴い自館のサービスには影響が出るか、またデータ項目やその定義を変更する必要があるかなど、どのような WebAPI を利用あるいは改修して構築したか、その関係性を整理して把握しておく必要がある。

3.7 関係ドキュメントの整備

WebAPI は外部から利用されることが前提となる。このため、システム開発の成果物として納品される仕様書等のドキュメントとは別に、より平易な利用者向けのドキュメントを整備する必要がある。

ドキュメントの例として、NDL サーチでは Web ページ「API のご利用について」⁸⁾ として利用の資格や手続き方法を、「API 仕様の概要」⁹⁾ で提供している API の種類とリクエスト例、詳細な仕様書を公開しているので参考とされたい。

利用者向けのドキュメントには、以下のように「API 仕様書」や「利用規約」などが考えられる。

(1) API 仕様書

以下の内容を明確に記述する。

- ・API 機能
- ・API 利用方法
- ・エラーコード
- ・リクエスト
- ・レスポンス
- ・提供データに関する説明

API 機能には、取り扱うデータや操作内容などの機能概要を記述する。また API 利用方法は、API を利用するための URI (エンドポイント) や送受信されるデータのエンコード方式、認証の有無や利用制限 (回数の制限) などを記述する。エラーコードは、API 利用時に「該当のデータがない」などのエラーが発生した場合レスポンスとして返される内容を記述する。

リクエストは「URI をどう記述してリクエストするか」またレスポンスは「どのようなデータが返ってくるか」をそれぞれ記述する。いずれもサンプルがあればわかりやすい。先のエンドポイントも含め、URI は動詞や動作を表す言葉ではなく具体的な名詞として、人間が読んだ時にリソースの内容が把握できる単語を用いる。URI の設計方法の詳細は「API テクニカルガイドブック」¹⁰⁾ を参照されたい。

提供データに関する説明には、データの更新日や更新頻度が含まれる。また、3.4 で検討した出力対象となるデータ項目や、既存のデータ項目から WebAPI のためにどのように変換したかの定義 (マッピング)、3.5 で定義した項目ごとの入力規約や使用しているコードの説明を記述する。

NDL サーチに関しては、概要などは上記の Web ページに、リクエストやレスポンスの例など詳細は「国立国会図書館サーチ外部提供インタフェース仕様書」¹¹⁾ として公開されている。

(2) 利用規約

利用の条件や禁止事項、免責事項などを明確に記述する。「ガイドブック」には「API 利用規約テンプレート」として利用規約の条文の例と解説が含まれるので、参考とされ

たい。主な事項を以下に挙げる。

- ・コンテンツの利用条件（ライセンスなど）
- ・クレジット表示（表示の要否、内容など）
- ・アクセス条件
- ・禁止事項
- ・免責事項

ライセンスについては、元のデータの入手先、たとえば自館で作成あるいは購入しているか、他所から API 経由で入手したか、などによっても異なる。また、クレジット表示については、必要があれば「〇〇図書館から API 経由で入手したデータを利用」など出典の表示を求めることも可能であろう。

アクセス条件には、利用にあたっては利用者が環境を整備すること、システムの負荷によっては利用制限を行うことがあること、仕様の変更は事前通知なしに行える、などの条件が含まれる。また、API キーなど利用者ごとに ID を発行して利用状況の把握などを行う場合は、API キーの発行にかかる手続きなどを利用規約に加える。

禁止事項には、API の提供者等への不利益となる行為や、大量のアクセスを行うなど運用に支障を与える行為など、利用者が行ってはならない行為を定める。免責事項では、WebAPI の利用に関して利用者の環境に不具合を生じさせないこと、また提供する情報の正確性、完全性などを保証するものではないことを明らかにする。

3.8 セキュリティ

WebAPI の設置は、外部から常時アクセスできる入口を別に設けることになり、セキュリティリスクを持つことになる。「API テクニカルガイドブック」では以下のような REST API におけるセキュリティ対策例が示されており、その概要を紹介する。

特に WebAPI は手動ではなく自動的かつ定期的に処理されることが多く、また一秒に何回という単位でリクエストを送信することも可能である。このため、システムへの負荷が上がりやすい。利用者の利用方法によっては双方が予想しない形で障害が発生する可能性もあり、利用履歴などを把握してサーバへの負荷と利用件数のバランスをとって運用することが必要である。

(1) 通信の改ざんや盗聴防止

TLS (SSL) をサポートとした通信により暗号化を行うことで通信の改ざんや盗聴を防止することが必須である。これは WebAPI に限らず、Web で行う通信すべてに求められる。

(2) 情報漏えい防止

対策として、利用者情報の登録を終えた者に API キーを発行し、これにより認証を行い利用させることが強く推奨される。ただし、API キーはコピーされうるため、強固な対策ではない。より正確な利用者認証が必要な場合は、OpenID Connect による認証を行うなどの方法もある。

また、OAI-PMH のように API キーの使用を想定していない WebAPI の場合は、「API テクニカルガイドブック」に記載はないが IP アドレスによるアクセス制限を行うのも方法であろう。いずれも、提供する情報の守秘性に依拠して検討すべき事項である。

(3) サーバ負荷の管理

WebAPI は機械的に行われることから、大量のアクセスにより DoS (Denial of Service attack: サービス拒否攻撃) の標的にされる可能性が高くなる。

対策として、前項の API キーを元に利用制限を行うことができる。例えば、一日あたりのアクセス回数の上限を設定する、などである。大量のアクセスを発見した場合は該当の利用者へ通知を行うこともできるが、あまり厳しい制限を行うと利便性を損ねるため、利用者の立場から適切な制限を行う必要がある。また、同一のリクエストに対してはキャッシュされたデータからレスポンスを返すなど、負荷を下げる工夫もできる。

(4) クロスドメインの通信許可

クロスサイトスクリプティング (XSS) を防止するため、Web ブラウザではクロスドメイン、すなわち Web サービスとは別のドメインとの通信を行わない機能が組み込まれている。WebAPI では、Web サービスとは別のドメインにアクセスすることになるため、クロスドメインの通信を許可する CORS (Cross Origin Resource Sharing: オリジン間リソース共有) への対応が必要となる。

4. WebAPI の運用

WebAPI の導入と運用を円滑に行うためには、情報提供とモニタリングが必要である。通常、WebAPI は自動処理により利用されるため、利用者が仕様の変更気づけない、あるいは誤った情報を取得し続けてしまう可能性がある。このため、仕様変更やサービス停止の情報提供には十分な周知期間を設け、丁寧に行う必要がある。3.7 で整備したドキュメントを Web サイトで公開し利用者に提供するほか、3.8 (2) で紹介した利用者登録は大規模な仕様の変更などの際に連絡手段となり得る。

通常の Web サーバの運用では、アクセスされたページや負荷の状況などをログにより把握する。WebAPI の運用でも同様であるが、利用制限の閾値を設定あるいは見直すために、利用者 ID ごとの利用件数などを取得、分析する必要があるだろう。

導入した WebAPI をより多くの者に利用してもらうために、コミュニティを活用する方法もある。一般にオープンソースソフトウェアの開発では、開発者のコミュニティが形成されこの中で有益な議論が行われる。図書館と類似機関が持つ学術情報であれば、図書館関連のコミュニティで発表し紹介するのも一つの方法であろう。

5. まとめ

本稿では、機械利用可能なインターフェースの整備について、WebAPIを取り上げ図書館分野での事例を元に公開の目的や利点を整理した。また、デジタル庁が作成したドキュメントを元に構築や設計、運用に関するポイントを述べた。

従来は、システムとサービスに特化した手順とデータ形式で複数のシステム間の連携が行われてきた。WebAPIを利用することでよりシンプルで、かつ汎用性の高いデータ形式で連携が行えるようになった。その設計に関しては、標準的に用いられているAPIやデータ項目、形式を採用することで、相互運用性を確保することができる。また、連携する相手がプログラム等を開発しやすいよう、ドキュメントを整え提供する必要がある。

図書館をはじめとする情報機関では、所蔵する資料の発見可能性を高め利用につなげるため、目録の作成や交換などに取り組んできた。これらは現在ではメタデータの作成と交換に置き換わり、機関リポジトリ等ではWebAPIによりメタデータを外部に提供することで収録している資料の情報を各所で検索できる基盤が作られてきた。

同時に、これらの情報は誰でも利用することができる。本稿で紹介したポイントを元に、多くのWebAPIが公開され多様な資料を発見し、そして利用可能になる未来が望まれる。

注・参考文献

- 注 1) 図書館分野におけるWebAPI利用の事例は、本誌64巻5号(2014.5)の特集「Web API活用術」を参照されたい。
注 2) 2010年5月に、岡崎市中央図書館Webサイト上の新着図書データを自動で取得するプログラムを実行し、結果としてサイトの一部機能を利用できなくなったとされて利用者が逮捕された事件が発生した。事件の概要は以下の文献を参照されたい。新出, 研究例会報告 第277回 Librahack 事件と図書館, 図書館界, 63巻1号, 2011, p.45-47. https://doi.org/10.20628/toshokankai.63.1_45, (参照 2024-11-10)

注 3) 標準的な技術の利用という観点からのWebAPIの事例は、本誌71巻4号(2021.4)の特集「学術情報流通を支える標準化技術」でも紹介されている。あわせて参照されたい。

- 1) Open Archives Initiative. The Open Archives Initiative Protocol for Metadata Harvesting. Protocol Version 2.0. 2002. <https://www.openarchives.org/OAI/openarchivesprotocol.html>, (accessed 2024-11-10)
- 2) 林賢紀, 宮坂和孝. RSS (RDF Site Summary) を活用した新たな図書館サービスの展開—OPAC2.0 へ向けて—. 情報管理, 2006, vol.49, no.1, p.11-23. <https://doi.org/10.1241/johokanri.49.11>, (参照 2024-11-10)
- 3) 中嶋晋平. CA1677 図書館によるAPIの公開—PORTAの事例から—. カレントアウェアネス, 2008, no.298, p.10-12, (参照 2024-11-10)
- 4) 国立情報学研究所. CiNiiのリニューアルについて. <https://support.nii.ac.jp/ja/news/cinii/20090401>, (参照 2024-11-10)
- 5) 大向一輝. CiNiiのウェブAPI戦略. 情報の科学と技術, 64巻5号, 2014, p.170-174. https://doi.org/10.18919/jkg.64.5_170, (参照 2024-11-10)
- 6) デジタル庁. API導入実践ガイドブック. 政府相互運用性フレームワーク (GIF) 460 実践ガイドブック. 2024. https://github.com/JDA-DM/GIF/blob/main/460_実践ガイドブック/docx/464-1_API導入実践ガイドブック.docx, (参照 2024-11-10)
- 7) 国立国会図書館. WebAPIによるシステム連携ガイドライン. Version 2.0, 2024, https://ndlsearch.ndl.go.jp/file/renkei/info/reference/WebAPIguideline_ver.2.0_20240105.pdf, (参照 2024-11-10)
- 8) 国立国会図書館. 17.APIのご利用について. 国立国会図書館サーチヘルプ, <https://ndlsearch.ndl.go.jp/help/api>, (参照 2024-11-10)
- 9) 国立国会図書館. API仕様の概要. 国立国会図書館サーチヘルプ, 2024, <https://ndlsearch.ndl.go.jp/help/api/specifications>, (参照 2024-11-10)
- 10) デジタル庁. APIテクニカルガイドブック. 政府相互運用性フレームワーク (GIF) 460 実践ガイドブック. 2024. https://github.com/JDA-DM/GIF/blob/main/460_実践ガイドブック/docx/464-2_APIテクニカルガイドブック.docx, (参照 2024-11-10)
- 11) 国立国会図書館. 国立国会図書館サーチ外部提供インタフェース. 1.2 版, https://ndlsearch.ndl.go.jp/file/help/api/specifications/ndlsearch_api_20240712.pdf, (参照 2024-11-10)

Special feature: Working towards a machine-friendly data infrastructure. “Development of machine-readable interfaces”. Takanori HAYASHI (Public Relations and Publications Section, Information and Public Relations Office, Japan International Research Center for Agricultural Sciences, 1-1 Ohwashi, Tsukuba Ibaraki, 305-8686, JAPAN)

Abstract: In the field of libraries, initiatives such as the development of union catalogs have been undertaken to enhance the discoverability of library holdings. The use of Web APIs has further streamlined and improved the efficiency of metadata collection and distribution. This paper introduces the purpose and advantages of Web APIs, which serve as interfaces for integrating web services, using examples from the library domain. Additionally, it provides a detailed explanation of key considerations in the design and operation of Web APIs, including ensuring interoperability through the use of widely adopted Web APIs and data formats, as well as improving usability by preparing and publishing user documentation.

Keywords: Library Information System / WebAPI / System development / Web service / Data integration