

Web API の過去・現在・未来

著者	高久 雅生
著者別名	Takaku Masao
雑誌名	情報の科学と技術
巻	64
号	5
ページ	162-169
発行年	2014-05
URL	http://hdl.handle.net/2241/121387

Web API の過去・現在・未来

高久 雅生*

Web API は情報サービスの、とりわけ Web 上で提供されるサービスにおいて、縁の下の力持ちとして欠かせない位置を占めつつある。本稿では Web API の提供と利用の両面から類型を示しながら、その歴史的な意義と機能を紹介する。学術情報分野を中心に Web API の提供事例とマッシュアップ事例を挙げるとともに、その利用目的やデータ形式、ライセンス、永続性といった点を取り上げて議論し、Web API がもたらしている役割を解説する。

キーワード：Web サービス、Web 2.0、マッシュアップ、Linked Open Data (LOD)、REST、SOAP

1. はじめに

本特集をはじめめる起点として、Web 2.0 を振り返ろう。2000 年代半ばに、当時隆盛しつつあった先進的な Web 上のサービスやその進展を示す用語として「Web 2.0」が喧伝された¹⁾²⁾³⁾。Web 上で提供されるサービスが、人々の生活を支える一種のプラットフォームとして重要な存在となり、利便性を高めていることを実証しつつあった。提唱者の一人である O'Reilly は、Web 2.0 の定義を述べた論考「What is Web」¹⁾において、Web 2.0 の特徴を示すいくつかの側面に触れながら、その主要な原則として以下の 7 つを示している。

1. プラットフォームとしての Web
2. 集合知の活用
3. データ活用の推進
4. ソフトウェアリリースサイクルの終焉
5. 軽量プログラミングモデルの活用
6. 複数デバイス対応のソフトウェア
7. リッチなユーザ体験

これらの原則のうち複数を結びつける要素が「Web API」である。Web 2.0 技術における Web API とは、例えば、Google Maps や Gmail に代表される、膨大なデータを高速に管理・処理するサーバ側の機構に加え、ページロードに伴う遅延を感じさせないための AJAX と呼ばれるブラウザとサーバ側との非同期的なデータのやり取りの仕組みに特徴的にあらわれている。さらに、ブログサービスやソーシャルブックマークサービス等では、利用者がサービス上で蓄積したデータを活用できるよう、いつでも全てのデータを抽出して、他のサービスと互換性を保つための工夫がなされた。こういったプラットフォームサービスが相互運用可能な形式によりデータを出力するという機能の発展形が Web API であり、ブログサービスにおける RSS フィー

ドの仕組みやその派生サービスはその典型であった。XML 形式の機械可読データをやり取りすることにより、相互に異なるシステムが多様に結びつく疎結合型のサービス群を生むこととなった。2006 年に始まったマイクロブログサービス Twitter は、そのサービス全体を Web API により構成することで、自身が提供するよりも第 3 者提供による Twitter クライアントや関連サービスなど、多様なツール群が提供され、自社単体によるよりもはるかに大きなサービスを構成する、さながら一つのエコシステム（生態系）を作るにいたった。このように、2000 年代半ばに隆盛を極めた Web 2.0 というムーブメントを代表とするシステム機能が Web API であったし、その恩恵により、多くのサービスが形作られ、多くの人々の暮らしを支えるプラットフォームとなっていった。本稿では、この Web API とは何か、その役割とサービス事例を解説しながら、論点の整理を試みる。

2. Web API とは

Web API について解説する前に、まず、API について述べておく。API とは Application Programming Interface の頭文字をとった略語であり、ソフトウェア工学の用語で、プログラムモジュール間のやり取りの取り決めを指す。つまり、あらかじめ構築されたシステムモジュールの機能を呼び出す際の規約として、どのような値をパラメータとして呼び出すか、パラメータ名称は何かなど、ソフトウェア

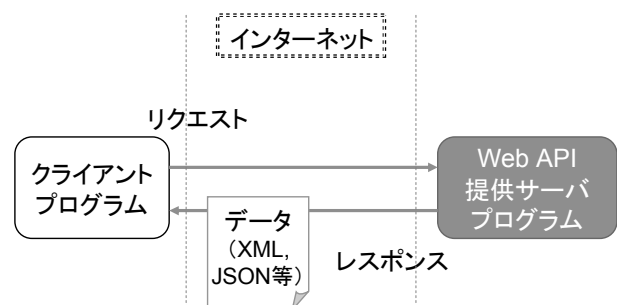


図 1 Web API におけるやり取りの概要

*たかく まさお 筑波大学図書館情報メディア系

〒305-8550 茨城県つくば市春日 1-2

Tel. 029-859-1394

(原稿受領 2014.3.18)

モジュール間で齟齬をきたさないようにするためのものである。Web APIはこの通常のソフトウェアにおけるモジュール呼び出しの概念を、Web上から遠隔に呼び出せるようにした概念である。

Web APIの概要を図1に示す。Web APIは、サーバ側が提供するプログラム機能をインターネット越しにクライアント側のプログラムが呼び出すことにより実行される。Web APIの呼び出しは、通常のWebの仕組みを用いて実行される。WWWを構成する3つの要素：URL、HTTP、HTMLのうちの前2者を用いる。つまり、特定のURLを指定して、HTTPを通じたリクエストを送ることがWeb APIの実行となる。Web APIの実行結果はプログラムから読み出しやすく理解しやすい形式としてクライアント側に返される。XML (Extensible Markup Language) やJSON (JavaScript Object Notation) 形式が用いられることが多く、具体的な形式としては、分野ごとに特有の形式RSS, Atom, RDFなどを用いることも多い。

2.1 Web APIの構成による類型

Web APIの呼び出し方式は2つに大別される。

- ・ REST方式
- ・ SOAP方式

REST (Representational State Transfer) は、Web上で提供されるサービスのHTTPプロトコルにおけるやり取りを標準的なモデルとしてまとめたものである⁴⁵⁾。HTTP上でアクセスできるWebリソースを、そのリソースに対応するURIへの操作としての追加機能、編集機能、削除機能をHTTPメソッドGET, POST, DELETE等と結びつけ、相互にやり取りするメカニズムをモデル化したものである。後述するSOAP方式に比べて、シンプルなモデルであることと、大規模なサービスでも適用可能な拡張性に優れたソフトウェアモデルであったことから、Web APIの呼び出し形式としてもっとも多く使われている。

SOAP⁶⁾はWeb Servicesとも呼ばれるWeb APIサービス規約群であり、RPC (Remote Procedure Call) と呼ばれる、プログラムモジュール同士をネットワーク上で実行するための遠隔通信規約と遠隔処理の枠組みを受け継ぐ目的で確立された⁷⁾。WSDL (Web Services Description Language)⁸⁾といったサービス記述の枠組みと合わせて用いることにより、サービス提供側、サービス利用側のプログラム開発や相互理解をスムーズに行える利点がある。SOAP方式のWeb APIはサービスの記述と発見、利用といったメカニズムを厳密に定義する伝統的なソフトウェア開発の枠組みとの親和性が高い点から、企業などによる大規模なソフトウェアやサービスにおいて適用されている。

Web以前から1980年代以来のソフトウェアモジュール間のやり取りの枠組みであるRPCやCORBAの枠組みをWeb上で実現しようとするSOAP方式を旧来型とすると、REST方式はHTTP通信規約の議論を基にまとめられた比較的新しい枠組みである。GoogleやAmazonといった先進的なWeb企業による初期のWeb API提供においては、

Google Search API (2002年開始) はSOAP方式による提供、Amazon Associates Web Services (2003年開始) はSOAP/REST両方式により提供されていた。一方で、SOAP方式においては、Web API利用にあたってXMLをベースとする比較的大きなソフトウェアライブラリを必要とする点などから、企業内ソフトウェアなどでの利用を除くと、軽量プログラミングモデルを旨とするWeb 2.0以降、Web上の公開サービスではあまり用いられなくなりつつある。たとえば、Musser⁹⁾によると、Web APIポータルサイトProgrammableWeb¹⁰⁾におけるWeb APIの呼び出し方式の構成割合は2008年にREST方式が60%、SOAP方式が25%だったのに対し、2010年にはREST方式が74%、SOAP方式は15%へと変化し、大多数がREST方式に移行したと述べている。

2.2 Web APIのアプリケーション類型

Web APIを利用したアプリケーションプログラムは、スタンドアロンアプリケーションもありうるが、Web上でアプリケーションとして提供されることも多い。このようなWeb APIを利用するアプリケーションをマッシュアップ (Mashup) と呼ぶ。

Web APIのマッシュアップ利用はその構成から、おおむね以下の3つに類型化できる¹¹⁾。

1. サーバ利用型マッシュアップ
2. AJAX型マッシュアップ (ウィジェット型マッシュアップ)
3. データ集約型マッシュアップ

サーバ利用型マッシュアップの概要を図2に示す。エンドユーザはマッシュアッププログラムをブラウザを通じて利用し、この際、利用のためのリクエストを送ると、マッシュアッププログラム側がそのリクエストをその内容に応じた形でWeb API提供サーバ側に送り、Web API提供プログラムが処理結果をXMLやJSONなどのデータとして送り返すと、マッシュアッププログラムはその結果を用いた処理結果をWebページとしてブラウザに送り返す。

サーバ利用型マッシュアップでは、リクエストが直列的

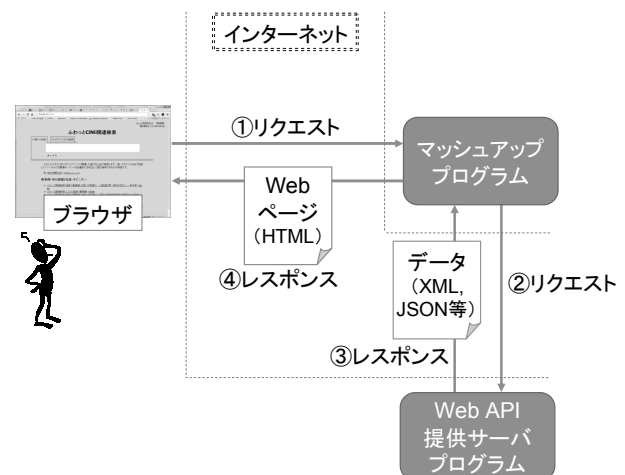


図2 Web APIにおけるやり取りの概要

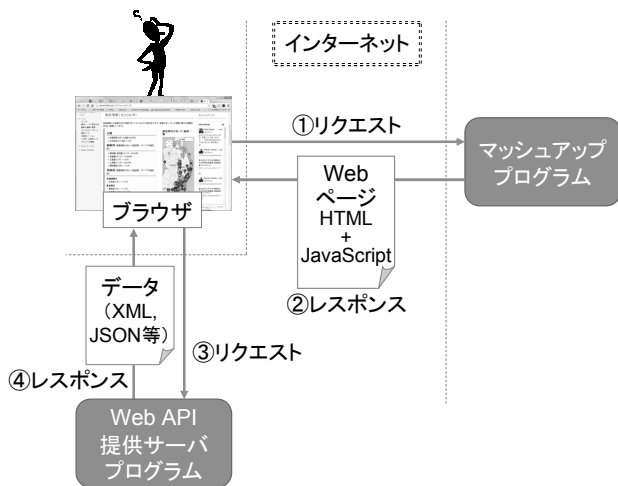


図3 AJAX型マッシュアップにおけるやり取りの概要

に送られ、最終的にブラウザに返されるため、Web API 提供側サーバとマッシュアップ側との応答速度がサービス全体の速度に影響を及ぼすこととなる。この点は次のAJAX型マッシュアップとの大きな相違点である。

AJAX (Asynchronous JavaScript and XML) とは、Web ページ内に埋め込まれた JavaScript プログラムを通じて、Web API とのやり取りを行うことにより、Web ページを構築する手法を指す。図3はAJAX型マッシュアップにおけるやり取りを図示している。ここでは、マッシュアッププログラム側は、リクエストに応じた内容のWeb ページをすぐに返すこととなる。この返されたWeb ページにWeb API を呼び出すJavaScript が付加されていることにより、AJAX 呼び出しが行われ、Web API には呼び出し側のブラウザからリクエストが行われ、Web API からの応答を読み込んで、読み込み結果も JavaScript プログラムが処理した上で、動的にブラウザの表示内容に反映される。先述のサーバ利用型マッシュアップとは異なり、Web API 呼び出し処理結果を待たずにページ表示したり、ブラウザ内のドラッグ操作やクリック操作に応じて動的にページ内容を書き換えたりすることが期待できるため、軽快なUIを構築できるのが特長であり、Google Maps API などのマッシュアップが代表的である。

なお、サーバ利用型マッシュアップとAJAX型マッシュアップとは併用することもできる。例えば、この種のマッシュアップとして最初に公開されたHousing Maps¹²⁾は、地域情報サイトCraigslistと地図情報サービスGoogle Mapsを組み合わせて、全米における不動産情報を並べて選んだりできるツールで、AJAX型APIのGoogle Mapsとサーバ利用型APIであるCraigslistとを組み合わせたツールであった¹³⁾。

データ集約型マッシュアップは、利用者のリクエストに応じてマッシュアップ先のデータを取得するのではなく、あらかじめWeb APIを通じて得られるデータそのものを蓄積しておき、利用者からのニーズには蓄積したデータを用いるタイプのマッシュアップサービスである。この種のアプリケーションの代表格は機関リポジトリの横断検索用

サービスである。機関リポジトリが標準的にサポートしているOAI-PMHプロトコル¹⁴⁾においては、リポジトリ内のレコードの新規登録、更新、削除といった操作結果を一定期間ごとに集計しながら全取得または差分取得する機能があるため、特定のリポジトリ内のメタデータをまるごと取得するメタデータハーベスティングに対応している¹⁵⁾。

3. 事例から：学術情報流通分野を中心に

2000年代初頭から提供がはじまったWeb APIには、既に多くの提供事例があり、さらにその適用事例としてのマッシュアップサービスやアプリケーションも膨大な数に及ぶ。Web API情報全般を集めた開発コミュニティサイトProgrammable Web¹⁰⁾は2014年3月現在、11,122件のAPIと7,375件のマッシュアップ適用事例を掲載している。

3.1 Web APIの提供事例

学術情報流通の分野でもWeb APIを提供している多くの事例がある。たとえば、国内の主要な書誌ユーティリティ、図書館、情報センターは2000年代半ばから、次々とWeb APIの整備と提供を続けてきている。国立国会図書館における検索システムPORTA¹⁶⁾¹⁷⁾は2008年にWeb API提供を開始し、PORTA後継である国立国会図書館サーチ(NDLサーチ)¹⁸⁾¹⁹⁾においても、Web APIの提供は継続し、Dublin Coreに基づくメタデータ形式DCNDLによる提供が行われている。国立情報学研究所が提供するCiNii Articlesも2009年以来、OpenSearchと詳細なRDFメタデータに基づくWeb API提供を行っている²⁰⁾²¹⁾²²⁾²³⁾。CiNiiにおけるWeb APIの活用の取り組みについては本特集の大向による論考も詳しい²³⁾。農林水産研究情報総合センターも農林水産分野の蔵書検索や研究報告等のメタデータをWeb APIを通じて検索、取得できるようにしている²⁴⁾。科学技術振興機構が提供する情報検索サービスJ-GLOBALにおいて、2010年よりWeb APIを提供している²⁵⁾。

そのほかの例として、カーリル²⁶⁾は全国の図書館を対象とした横断検索サービスとして2010年に開始したサービスだが、そのサービス全体をWeb APIとしても公開している²⁷⁾。

また、書誌情報以外でも、レファレンス協同データベース²⁸⁾は、レファレンス相談の質問回答を集約したデータベースを構築しており、その蓄積された質問回答などに対する検索Web APIを提供している²⁹⁾。

上記のほか、情報検索、メタデータハーベスティング、リンクリゾルバなどの分野における事例がある。情報検索の分野では、Web出現以前より普及が進んでいた別規格Z39.50³⁰⁾が複数の情報検索システムの相互運用性を保って連携させる手段となってきた³¹⁾。Z39.50はWeb出現以前から開発されてきた規格であり、WebやWeb APIとの互換性はない。Web普及後に、Z39.50の次世代版として標準化されたのがSRU/SRWである³²⁾。SRU/SRWは、SRU

がいわゆる REST 方式であり、SRW は SOAP 方式によるものであり、Web API としてのパラメータ等は同一のものである。先に述べたように、SOAP 方式の SRW はさほど普及しておらず、SRU/SRW と並べて表記されるものの、SRU の利用が中心となっている。

一方で、Z39.50 とは別に急速に発展した Web 検索エンジンなどにおける検索仕様においては独自のものが多い。また、A9.com から、より簡便な検索 Web API 仕様 OpenSearch³³⁾が提案され、普及が進んだ。

学術情報流通の分野ではすでに見たように、情報検索に用いる Web API として、SRU/SRW、OpenSearch などがある。検索以外の他ツールとのやり取りとしては、リンクリゾルバで利用されている OpenURL³⁴⁾、機関リポジトリを中心として用いられている OAI-PMH (Open Archive Initiative Protocol for Metadata Harvesting)¹⁴⁾がある。

また、文献管理ツール Mendeley³⁵⁾も多くのシステム機能に対応する Web API を開発者向けに提供している³⁶⁾。

3.2 Web API の利用事例 (マッシュアップ)

先に述べたように、海外の開発者向けサイト ProgrammableWeb に登録されているマッシュアップ事例だけで 7,375 例あり、膨大なマッシュアップ適用が実現されている。このようなマッシュアップ事例は、企業・団体などによるサービスだけでなく、個人による開発によって実現されているものも多い。このようなマッシュアップ構築にあたってよく使われている Web API を拾ってみると、2014 年 3 月時点の ProgrammableWeb におけるマッシュアップ登録件数では、Google Maps (39%)、Twitter (12%)、Youtube (10%)、Flickr (9%)、Amazon (6%)、Facebook (6%) といった順に多く使われている。それ以外には、国内では、Yahoo! JAPAN や楽天といった国内の大手 IT 系企業による Web API サービスが多く使われている³⁷⁾³⁸⁾³⁹⁾。

学術情報の分野におけるマッシュアップ事例のひとつとして、国内でも早期に公開されたマッシュアップ例は myrmecoleon による所蔵図書館マップ⁴⁰⁾であり、図書 ISBN を指定してその本を所蔵する図書館を Google Maps 上に表示するものであった。また、ゆうき市立図書館では受け入れ雑誌の目次レベルの情報を提供するため、国立国会図書館の雑誌記事論文索引の RSS およびマガジン書店 Fujisan の提供する RSS をもとに、自動的に新着記事を一覧するサービスを開発し提供している⁴¹⁾⁴²⁾。新着雑誌記事速報の詳細は本特集における川嶋の論考も参照のこと⁴³⁾。そのほか、Twitter API と CiNii API を用いた論文情報をつぶやくボットである「論文ったー」⁴⁴⁾⁴⁵⁾や類似文書検索ツールである「ふわっと関連検索」⁴⁶⁾⁴⁷⁾といったマッシュアップツールも開発されている。

また、海外の事例においても、OCLC における WorldCat Search API を用いたマッシュアップ事例の一覧が公開されており、2014 年 3 月時点で 92 事例が公開されている⁴⁸⁾。

機関リポジトリで提供される OAI-PMH¹⁴⁾はもともとメ

タデータハーベスティングとしてデータ集約型マッシュアップを実現する手段である。つまり、大学や研究機関ごとに個別に構築されるシステム内のメタデータを更新情報を含んですべて取得でき、データ集約型の検索サービスを構築することを前提としている。たとえば、全世界の機関リポジトリ情報を集約して収集した検索サービス OAIster⁴⁹⁾がある。OAIster の国内版として、日本国内の機関リポジトリをまとめた横断検索サービスとして国立情報学研究所が提供する JAIRO⁵⁰⁾がある。同様に、ヨーロッパにおける複数機関リポジトリ横断型の検索サービスに OpenAIRE⁵¹⁾がある。また、単に集約したデータを横断的に検索提供するだけでなく、さらにそのマッシュアップサービス自身が Web API を提供することもできる。たとえば AirWay⁵²⁾は国内の機関リポジトリに蓄積された学術文献を取得し、OpenURL を通じてリンクリゾルバに情報を渡すことにより、もともとの提供元にはなかったチャンネルにデータを広げ、活用できるようにしている⁵³⁾⁵⁴⁾⁵⁵⁾。

また、ソーシャルメディア上における学術文献の利用動向や評判分析を行うオルトメトリックス (Altmetrics) と呼ばれる試みにおいても、Web API の利用は顕著である⁵⁶⁾。海外では Altmetrics⁵⁷⁾や ImpactStory⁵⁸⁾といったサイトが文献データベースの情報と各種ソーシャルメディアの情報を取得、集約し、マッシュアップした結果を可視化するサービスを提供している。同様の国内のサービスとしては吉田らによる Ceek.jp Altmetrics⁵⁹⁾があり、国内の学術文献に対するソーシャルメディア上での言及数などを可視化するサービスを提供している。これらのサービスは、マッシュアップ元とする学術文献サービスはもちろん、ソーシャルメディア側のサービスも Web API が提供されていることにより、サービスが構築できている。このように、従来は相互に無関係と思われてきた分野同士のデータやサービスを結び付け、効果を測定したり、実際に使う手間を減らすことができることが Web API の持つ直接的な効能のひとつといえる。

また、エンドユーザが利用するマッシュアップアプリケーションだけでなく、マッシュアップサービスを作ることを支援するツールもいくつか公開されている⁶⁰⁾。たとえば、米国 Yahoo! 社が提供する Yahoo Pipes⁶¹⁾では、Web API でのやり取りを GUI を通じて Web API から取得したデータを並べ替えたり、変換したりする処理を自動的に行うことができる。

3.3 Web API コンテストとハッカソン

Web API コンテストやハッカソンなどの動きについても触れておきたい。2006 年に始まった Mashup Awards は国内の IT 企業を中心に開催され、Web API を用いて構築されたマッシュアップアプリケーションをコンテスト形式で審査、表彰するもので、2013 年まで 9 回開催され、2013 年には 460 作品もの応募があった⁶²⁾。国立情報学研究所は 2009 年、2010 年の 2 回にわたり「CiNii ウェブ API コンテスト」を開催した⁶³⁾⁶⁴⁾²³⁾。国立国会図書館もレファ

レンス協同データベースの API を対象に「API 腕自慢」と題する公募企画を 2010 年に実施した²⁹⁾。LOD (Linked Open Data) における Web API およびオープンデータ活用を企図した「Linked Open Data チャレンジ Japan」も 2011 年以來、継続的に開催され、マッシュアップアプリケーションに限らず、データセットやアイデア、可視化なども表彰している⁶⁵⁾。また、Mashup Awards や LOD チャレンジでは、単にコンテストを実施するだけでなく、勉強会やハッカソンなどのイベントを開催することを通じ、Web API を提供するだけでなく、アプリケーション開発やアイデアの醸成を促進する仕組みとして、コンテストを活用している点も特徴的である。ハッカソンを通じたマッシュアップ等への取り組みは本特集の鎌田による論考も参考にしたい⁶⁶⁾。

4. Web API の提供と利用にあたって考慮しなければならないこと

Web API を提供するにあたって、または、利用するにあたって、双方の立場で考慮しなければならないことがある。個別の事例ごとの詳細な議論は本特集の各論考を参照いただくとして、ここでは一般論として考慮すべき点にあらじめ触れておきたい。

4.1 Web API を提供する目的

そもそも、あるサービスが Web API を提供するのにはなぜだろうか。これは、Web API の提供と利用にあたって、もっとも重要な問となるだろう。簡単にまとめると、主に以下の 3 つの役割が考えられる：

1. サービス利用の促進
2. 他のサービスや団体との相互連携
3. 思いもよらないアイデアやアプリケーションの実現

Web API の第 1 の目的は、サービスの利用者を増やしたり、サービスへのアクセスを増加させることである。商用サービスであれば、これらは直接的な利益となるため、この目的はかなり多くの部分を占めると考えられる。たとえば、Google や Amazon などの企業は提供サービスを使う利用者のアクセス増加や利用者チャネルが増加することにより、広告収入や購買収入を増す。このように、マッシュアップなどを通じて、直接的または間接的にそのサービスもしくは扱っているデータを使う機会が増えることにより、利用アクセスや利用そのものが増える。

第 2 の点として、標準的な Web API を用いることにより、同一のシステムや同一の開発企業を前提とせずとも連携を図ることが可能となる。たとえば、Web API が無い状態で別の機関が提供するシステム同士の横断検索を行おうとすると、相手側との合意、調整を行ったうえで、相互に連結するシステムを個別に開発する必要がある。これには非常にコストと労力がかかり、システムが変更された場合に対応しなおすことも難しくなるが、統一的な Web API への対応ができれば、この手間を減らすことができるため、コミュニケーションや開発にかかるコストを省力化できる。

最後の点として、Web API を提供することを通じて、当初は想定されなかったような利用形態やマッシュアップを生み出す、一種のイノベーション効果も期待できる。Web API コンテストやハッカソンなどは、単純に認知度やサービス利用を増やす目的もあるが、この点の効果を促進することを期待して開催されている側面も強い。Web API を活用するマッシュアップの開発は、ハッカー文化、オープンソースソフトウェア文化とも関連深く、個人や有志の開発者たちが本務としない環境下で構築することが多いことが報告されており、利用者自身の手によるプロトタイプ型開発も期待できる⁶⁷⁾⁶⁸⁾。

4.2 提供データの形式

Web API を提供する際にも、利用する際にも、やり取りされるデータがどのようなデータであるかが規定されていなければならない。この際、やり取りするデータは、できるだけ、利用に簡便で、標準化されたものであることが望ましい。そして、そのデータ内容の意味が分かるようデータ仕様を文書化し、明示する必要がある。

Web API のやり取りに用いるデータ形式は、以前は XML を基にしたものが標準的であったが、2000 年代後半からは、より簡潔で分かりやすく、ソフトウェアでも扱いやすい特長を持つ JSON をベースにするものが多くなりつつある。LOD におけるデータ構造を表現する、JSON-LD⁶⁹⁾のような複雑な表現もできる記述形式なども提案され、使われ始めている。

4.3 利用制限とデータライセンス

Web API はクライアントプログラムによる機械的アクセスが行われるため、人手によるブラウジングを行う場合に比べ、サーバ側の処理リソースはより必要となる点に注意が必要である。このため、負荷がかかりすぎる場合などに備えて、Web API でアクセスするクライアントに認証を要したり、一定時間単位のアクセス回数に制限を加えている場合が多い。認証のための仕組みとしては、あらかじめクライアント毎に API キーを発行しておく方法や、ユーザ名とパスワードなどをやり取りする方法などがある。また、SNS (Social Network Service) などのように個別のユーザ単位で利用者情報を受け渡す必要がある場合などに、OAuth⁷⁰⁾⁷¹⁾⁷²⁾と呼ばれる認証用 Web API 仕様が使われている。

さらに、Web API を通じてやり取りするデータは、第 3 者が取得して場合によっては加工を施すなどして再利用する必要があることから、どのような利用方法までは許すか、利用規約などによりその扱いを明示する必要がある。同時に、やり取りされるデータ自体のライセンスも重要となってくる。近年ではオープンデータの潮流があり、機械可読データに Creative Commons ライセンスなどオープンなライセンス⁷³⁾を付与することにより、活用が促されることが期待できる。

4.4 永続性

Web APIにはどの程度の永続性が期待できるだろうか。たとえば、2002年からGoogle社が提供したGoogle Search APIは2006年には新規登録を停止し、2010年にサービスを停止した⁷⁴⁾。Twitter社も2013年にサービス開始以来のTwitter APIを廃止し、新しい認証付のWeb APIへ移行した⁷⁵⁾。これらのWeb APIの廃止は、主として経営戦略上の判断であり、Google Search APIの場合はより簡略な開発が容易なAJAX型への移行であり、Twitter APIの場合は認証を用いて開発者単位のサービスを強化しようとする意図であったと推察される。ほかにも、Web APIは廃止しないものの、一定回数以上のWeb API呼び出しを有料とする形へ移行したGoogle Maps APIの例⁷⁶⁾もある。

マッシュアップサービスの開発にあたっては、Web API側が変更される可能性を考慮に入れて、どの程度まで特定のWeb APIに依存すべきかを検討しておく必要がある。Web APIが停止したとしても代替しうる類似Web APIの存在もあわせて確認しておくといえよう。

逆に、ひとたびWeb APIサービスの提供を開始し、マッシュアップサービスなど、多様なアプリケーション群ができあがると、廃止のみならず、変更を行うだけでもそのWeb APIの利用者に大きな影響を与える場合も多い。これを避けるには、Web APIの提供をはじめる前に、あらかじめそのサービスの目的に合うよう設計を注意深く検討しておく必要がある⁷⁷⁾。

5. おわりに

2014年は、WWW (World Wide Web) の誕生から25年、Web 2.0の隆盛から10年の節目にあたる。Web APIの提供と利用はこの10年間を通じて発展を続けてきた。Web上で提供されるサービスのうち多くがさまざまなWeb API機能を備え偏在化しつつある現在、そして、第3者によるマッシュアップアプリケーションの開発とその果実を得ている今、次の段階として、こうしたWeb APIとアプリケーションとをめぐる優れた生態系をいかに作るか、そうしたサービスをプロデュースしていくインフォプロの力量が問われているように筆者には思える。Web APIは単なるデータのやり取りの仕組みというだけでなく、サービス改善のための手段であり、イノベーションを産み出すための足がかりとなってきた。学術情報分野においても、この点に焦点をあてた、マイニング探検会⁷⁸⁾、Code4Lib JAPAN⁷⁹⁾、国立情報学研究所学術情報ウェブサービス担当者研修⁸⁰⁾といった勉強会や研修も出現している。まずは身近な情報サービスがどのようにこれらのWeb APIを発展させているか、そこから確認してみることをお奨めする。本特集における色彩豊かな論考と事例を通じて、Web APIの技術的な枠組みとそこから生まれるサービス機能の一端を感じていただければ幸いである。

注・参考文献

(Web参照日はすべて、2014年3月8日です)

- 1) Tim O'Reilly. What is Web 2.0: Design patterns and business models for next generation of software, 2005. <http://oreilly.com/web2/archive/what-is-web-20.html>
日本語訳: 「Web 2.0: 次世代ソフトウェアのデザインパターンとビジネスモデル」.
http://japan.cnet.com/sp/column_web20/20090039/.
- 2) 小川浩, 後藤康成. Web2.0 BOOK. インプレスジャパン, 2006, 254p.
- 3) 特集「図書館とWeb2.0」. 情報の科学と技術, Vol.56, No.11, pp.501-535, 2006.
- 4) Roy T. Fielding and Richard N. Taylor. Principled Design of the Modern Web Architecture. ACM Transaction on Internet Technology, Vol.2, No.2, pp.115-150, 2002.
- 5) 山本陽平. Webを支える技術: HTTP, URI, HTML, そしてREST. Web+DB Press プラスシリーズ. 技術評論社, 2010, 377p.
- 6) SOAP Version 1.2 Part 0: Primer (Second Edition), 2007. <http://www.w3.org/TR/soap12-part0/>
- 7) 日本ユニテック DigitalXpress 編集部 (編). SOAP/UDDI/WSDL Web サービス技術基礎と実践徹底解説. 技術評論社, 東京, 2002, 429p.
- 8) Web Services Description Language (WSDL) Version 2.0 Part 0: Primer, 2007. <http://www.w3.org/TR/wsdl20-primer/>
- 9) John Musser. Open APIs: State of the Market, 2010. In: Glue Conference 2010, <http://www.slideshare.net/jmusser/pw-glue-conmay2010>
- 10) ProgrammableWeb. <http://www.programmableweb.com>
- 11) Duane Merrill. Mashups: The new breed of Web app: An introduction to mashups, 2006. <http://www.ibm.com/developerworks/library/x-mashups/>
- 12) HousingMaps. <http://www.housingmaps.com/>
なお、このツールは2014年3月現在、動作していない。
- 13) Jin Yu, Boualem Benatallah, Fabio Casati, and Florian Daniel. Understanding mashup development. IEEE Internet Computing, Vol.12, No.5, pp.44-52, 2008.
- 14) The Open Archives Initiative Protocol for Metadata Harvesting, 2002. <http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm>
日本語訳: 「Open Archives Initiative メタデータ・ハーベスティング・プロトコル」.
<http://www.nii.ac.jp/irp/archive/translation/oai-pmh2.0/OpenArchivesProtocol.htm>
- 15) Herbert Van de Sompel, Michael L. Nelson, Carl Lagoze, and Simeon Warner. Resource Harvesting within the OAI-PMH Framework. D-Lib Magazine, Vol.10, No.12, 2004. <http://www.dlib.org/dlib/december04/vandesompel/12vandesompel.html>
日本語訳: 「OAI-PMH フレームワークを使用したリソースハーベスティング」.
<http://www.nii.ac.jp/irp/archive/translation/resourceharvesting/>
- 16) 中嶋晋平. 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, no.298, pp.10-12, 2008. <http://current.ndl.go.jp/ca1677>
- 17) 国立国会図書館デジタルアーカイブポータル PORTA のご紹介. 国立国会図書館月報, no.574, pp.26-32, 2009.
- 18) 中山正樹. 国立国会図書館のデジタルアーカイブ構想. ず・ぼん, no.17, pp.86-99, 2011.
- 19) 中山正樹. 国立国会図書館におけるデジタルアーカイブ構築知の共有を目指して. 情報管理, Vol.54, No.11, pp.715-724, 2012.
- 20) 大向一輝. 学術情報サービスのユーザモデルとファインダビリティ. 情報の科学と技術, vol.58, no.12, pp.595-601, 2008.
- 21) 大向一輝. 学術情報サービスのメタデータ・デザイン. 情報の科学と技術, vol.60, no.12, pp.495-500, 2010.
- 22) 大向一輝. CiNii Articles のシステムデザインとデータモデル. 情報の科学と技術, vol.62, no.11, pp.473-477, 2012.

- 23) 大向一輝. CiNii のウェブ API 戦略と活用. 情報の科学と技術, vol.64, no.5, 2014.
- 24) 林賢紀. 農林水産研究情報総合センターのサービスの継承と現在: API の提供を中心に. 情報管理, vol.56, no.9, pp.582-591, 2013.
- 25) J-GLOBAL—ヘルプ—WebAPI の提供.
<https://jglobal.jst.go.jp/help/webapi.php>
- 26) カーリル—日本最大の図書館蔵書検索サイト. <http://calil.jp/>
- 27) 吉本龍司. 進化する図書館蔵書検索サイト「カーリル」. 情報管理, vol.55, no.2, pp.97-105, 2012.
- 28) 国立国会図書館. レファレンス協同データベース.
<http://crd.ndl.go.jp/reference/>
- 29) 堤恵, 佐藤久美子, 牧野めぐみ. 「レファ協」で拓くレファレンスサービスの新たな地平. 情報の科学と技術, vol.61, no.5, pp.187-193, 2011.
- 30) ANSI/NISO Z39.50-2003, Information Retrieval (Z39.50): Application Service Definition and Protocol Specification, 2003. <http://www.loc.gov/z3950/agency/Z39-50-2003.pdf>
- 31) Clifford A. Lynch. The Z39.50 Information Retrieval Standard: Part I: A Strategic View of Its Past, Present and Future. D-Lib Magazine, 1997.
<http://www.dlib.org/dlib/april97/04lynch.html>
- 32) The Library of Congress. SRU: Search/Retrieval via URL—SRU, CQL and ZeeRex. <http://www.loc.gov/standards/sru/>
- 33) Specifications/OpenSearch/1.1/Draft 5 - OpenSearch.
<http://www.opensearch.org/Specifications/OpenSearch/1.1>
- 34) ANSI/NISO Z39.88-2004 (R2010) The OpenURL Framework for Context-Sensitive Services, 2010.
<http://www.niso.org/standards/z39-88-2004>
- 35) Mendeley. <http://mendeley.com/>
- 36) ビクトール・ヘニング. 研究者コミュニケーションを根本から変える文書管理の変革 Mendeley CEO が語る学術情報流通の将来. 情報管理, vol.55, no.4, pp.253-261, 2012.
- 37) 加藤貴之. Web API 実践リファレンスブック. 毎日コミュニケーションズ, 2007.
- 38) ZAPA. 公開 API 活用ガイド. I/O books. 工学社, 2010, 223p.
- 39) 3Dogs. WebAPI 辞典. Pocket 詳解. 秀和システム, 2012.
- 40) myrmecoleon. 所蔵図書館マップ.
<http://myrmecoleon.sytes.net/map/>
- 41) 牧野雄二, 川嶋斉. 新着雑誌記事速報から始めてみよう: RSS-API を活用した図書館サービス. JLA 図書館実践シリーズ, no.21. 日本図書館協会, 2012, 161p.
- 42) 長谷川拓哉. 公共図書館における API 活用: 「新着雑誌記事速報」を例に. 専門図書館, no.255, pp.86-89, 2012.
- 43) 川嶋斉. 図書館サービスを Web API で拡張する. 情報の科学と技術, vol.64, no.5, 2014.
- 44) 論文ったー. <https://twitter.com/ronbuntter>
- 45) 山田俊幸. 空気を読んで論文を紹介する「論文ったー」. 専門図書館, no.255, pp.27-33, 2012.
- 46) 高久雅生. ふわっと CiNii 関連検索. <http://fuwat.to/cinii>
- 47) 高久雅生, 江草由佳. 簡易類似文書検索手法「ふわっと関連検索」の予備的評価と分析. 情報処理学会研究報告. 研究報告情報基礎とアクセス技術 (IFAT), vol.2010-IFAT-99, no.14, pp.1-6, 2010.
- 48) OCLC Developer Network. Gallery.
<http://www.oclc.org/developer/gallery.en.html>
- 49) WorldCat: OAIster. <http://oaister.worldcat.org/>
- 50) JAIRO: Japanese Institutional Repositories Online.
<http://jairo.nii.ac.jp/>
- 51) Find—Publications, data, projects, ...—OpenAIRE.
<https://beta.openaire.eu/search/find>
- 52) Airway プロジェクト.
http://airway.lib.hokudai.ac.jp/index_ja.html
- 53) Shigeki Sugita, Kunie Horikoshi, Masako Suzuki, Shin Kataoka, E. S. Hellman, and Keiji Suzuki. Linking Service to Open Access Repositories. D-Lib Magazine, Vol.13, No.3/4, 2007.
<http://www.dlib.org/dlib/march07/sugita/03sugita.html>
- 54) 紙谷五月, 野中雄司, 杉田茂樹. 機関リポジトリへのアクセス経路. 情報の科学と技術, vol.58, no.12, pp.610-614, 2008.
- 55) 吉田暁史, 横谷弘美. 学術情報流通における OAI-PMH の役割. 大手前大学論集, vol.10, pp.237-257, 2009.
- 56) 坂東慶太. Altmetrics の可能性: ソーシャルメディアを活用した研究評価指標. 情報管理, vol.55, no.9, pp.638-646, 2012.
- 57) Altmetric - We Make Article Level Metrics Easy.
<http://www.altmetric.com/>
- 58) ImpactStory. <http://impactstory.org/>
- 59) Ceek.jp Altmetrics —学術文献のインパクト.
<http://altmetrics.ceek.jp/>
- 60) Lars Grammel and Margaret-Anne Storey. A survey of mashup development environments. In Mark Chignell, James Cordy, Joanna Ng, and Yelena Yesha, editors, The Smart Internet, Vol.6400 of Lecture Notes in Computer Science, pp.137-151. Springer, 2010.
- 61) Yahoo! Pipes: Rewire the web. <http://pipes.yahoo.com/>
- 62) リクルートホールディングス. 日本最大級の Web 開発コンテスト「Mashup Awards 9」審査結果を発表, 2013.
http://www.recruit.jp/news_data/release/2013/1113_7355.html
- 63) CiNii ウェブ API コンテスト実施要項, 2009.
http://ci.nii.ac.jp/info/ja/web_api_contest_2009.html
- 64) 第 2 回 CiNii ウェブ API コンテスト実施要項, 2010.
http://ci.nii.ac.jp/info/ja/web_api_contest_2010.html
- 65) Linked Open Data Challenge Japan 2013.
<http://lod.sfc.keio.ac.jp/challenge2013/>
- 66) 鎌田篤慎. Web API 活用の現在～Hack for Japan の活動の事例から～. 情報の科学と技術, vol.64, no.5, 2014.
- 67) Björn Hartmann, Scott Doorley, and Scott R. Klemmer. Hacking, Mashing, Gluing: Understanding Opportunistic Design. IEEE Pervasive Computing, Vol.7, No.3, pp.46-54, 2008.
- 68) Ingbert R. Floyd, M. Cameron Jones, Dinesh Rathi, and Michael B. Twidale. Web Mash-ups and Patchwork Prototyping: User-driven technological innovation with Web 2.0 and Open Source Software. In Proceedings of HICSS 2007, pp.86-95, 2007.
- 69) Manu Sporny, Dave Longley, Gregg Kellogg, Markus Lanthaler, and Niklas Lindström. JSON-LD 1.0: A JSON-based Serialization for Linked Data, 2014.
<http://www.w3.org/TR/json-ld/>
- 70) Eran Hammer-Lahav. The OAuth 1.0 Protocol, 2010.
<http://tools.ietf.org/html/rfc5849>
- 71) Dick Hardt. The OAuth 2.0 Authorization Framework, 2012. <http://tools.ietf.org/html/rfc6749>
- 72) Barry Leiba. OAuth Web Authorization Protocol. IEEE Internet Computing, Vol.16, No.1, pp.74-77, 2012.
- 73) Open Definition, Version 1.1. <http://opendefinition.org/od/>
- 74) A well earned retirement for the SOAP Search API, 2009.
<http://googlecode.blogspot.jp/2009/08/well-earned-retirement-for-soap-search.html>
- 75) Twitter. API v1 Retirement is Complete - Use API v1.1, 2013. <https://blog.twitter.com/2013/api-v1-is-retired>
- 76) Thor Mitchell. Google Geo Developers Blog: Introduction of usage limits to the Maps API, 2011.
<http://googlegeodevelopers.blogspot.jp/2011/10/introduction-of-usage-limits-to-maps.html>
- 77) Daniel Jacobson, Greg Brail, and Dan Woods. APIs: A Strategy Guide. O'Reilly, 2011, 134p.
- 78) 清田陽司. つながれインフォプロ第 2 回. 情報管理, vol.56, no.8, pp.549-551, 2013.
- 79) 米田渉. Code4Lib JAPAN の活動と図書館. 図書館雑誌, vol.105, no.4, pp.215-17, 2011.
- 80) 国立情報学研究所. 教育研修事業—学術情報ウェブサービス担当者研修 (旧称: 学術ポータル担当者研修).
<http://www.nii.ac.jp/hrd/ja/portal/>

Special feature: How to use Web API. The Past, Present and Future of Web API. Masao TAKAKU (University of Tsukuba, 2-1 Kasuga, Tsukuba, Ibaraki, 305-8550 JAPAN)

Abstract: Web API has become essential role for information services, especially on the Web. Several categories of Web API are presented from the viewpoints of a provider and user as well as its functions and historical aspects. This paper demonstrates some examples from scholarly information fields, and discusses several issues such as objectives, data formats, licenses, and life cycles of Web APIs.

Keywords: Web services / Web 2.0 / Mashup / Linked Open Data (LOD) / REST / SOAP