

Web API入門

中村 覚

(東京大学情報基盤センター)

もくじ

- Web APIとは
 - Web APIの事例
- Web APIの応用: マッシュアップ
 - マッシュアップ事例
- Web APIの課題

高久雅生. 第23回大図研オープンカレッジ「大学図書館員のための Web API入門」,
<https://www.slideshare.net/tmasao/web-api-49080729>. CC BY-SA.

Web APIとは

Web APIとは

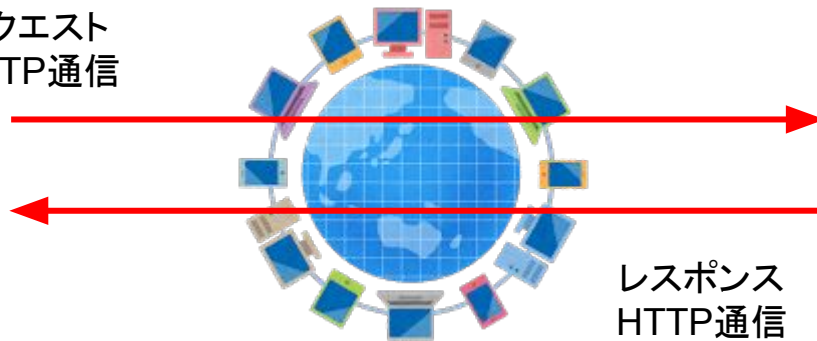
- ウェブ上で他のサービスを呼び出す方式、取り決め
 - 標準的なデータやサービスの提供を行う方法
- 具体的には:
 - ウェブを介して「要求(リクエスト)」を送り、サーバが要求された処理を実行し、「結果(レスポンス)」を返却する仕組み
 - リクエストはREST方式、レスポンスはXMLデータやJSONデータが一般的

Web APIとは(概念図)

例: ある著者が書いた書籍の一覧が欲しい



リクエスト
HTTP通信



Webサーバ



レスポンス
HTTP通信

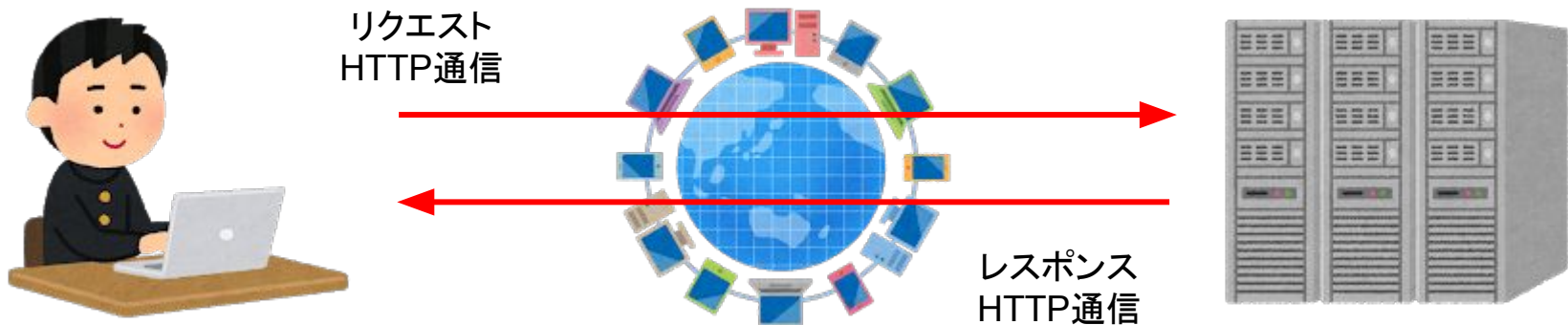
例: 検索結果として、書籍の
詳細リストを返す

Web APIの公開とは

- どのような形式の「要求(リクエスト)」か？
- どのような形式の「結果(レスポンス)」か？

を公開すること

=> そうすることで利用方法がわかるようになる



Webとは



World Wide Web

- WWW (World Wide Web)
 - ワールド・ワイド・ウェブ
 - ダブリュ・ダブリュ・ダブリュ
- または単にWeb(ウェブ)と称する
- 【web】(noun)
 - A network of silken thread spun especially by the larvae of various insects (as a tent caterpillar) and usually serving as a nest or shelter.



https://commons.wikimedia.org/wiki/File:Spider_web_Belgium_Luc_Viatour.jpg

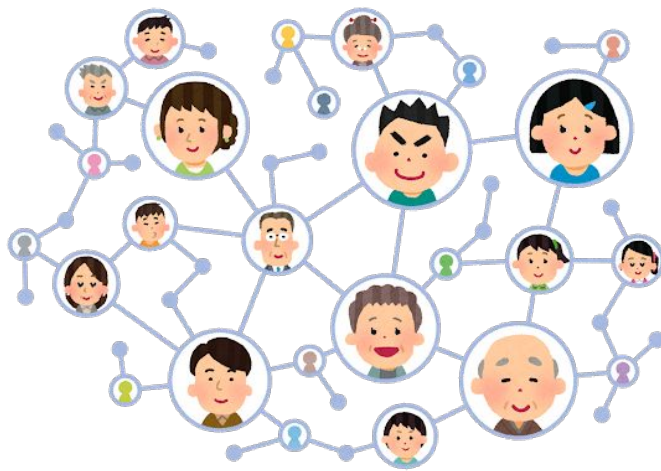
ティム・バーナード＝リー (Tim Berners-Lee)



<https://www.flickr.com/photos/knightfoundation/2467553359/>

Webの3要素

- HTTP, URL, HTMLの3つの枠組みを用いたWebの実現
- HTTPレイヤでのデータ転送とドキュメントフォーマット指定
- URI指定によるリンクを通じたハイパーメディアの実現



HTTPとは

- HyperText Transfer Protocol
 - 基本的なWebサーバとのやり取りを行う方式

GET /index.html HTTP/1.0

URLとは

- URL (Universal Resource Locator)
- Web上でのアドレスを指し示すアドレス(識別子)として機能する

`http://ci.nii.ac.jp/search?q=web+api`

サーバの
アドレス

サーバ内
での場所

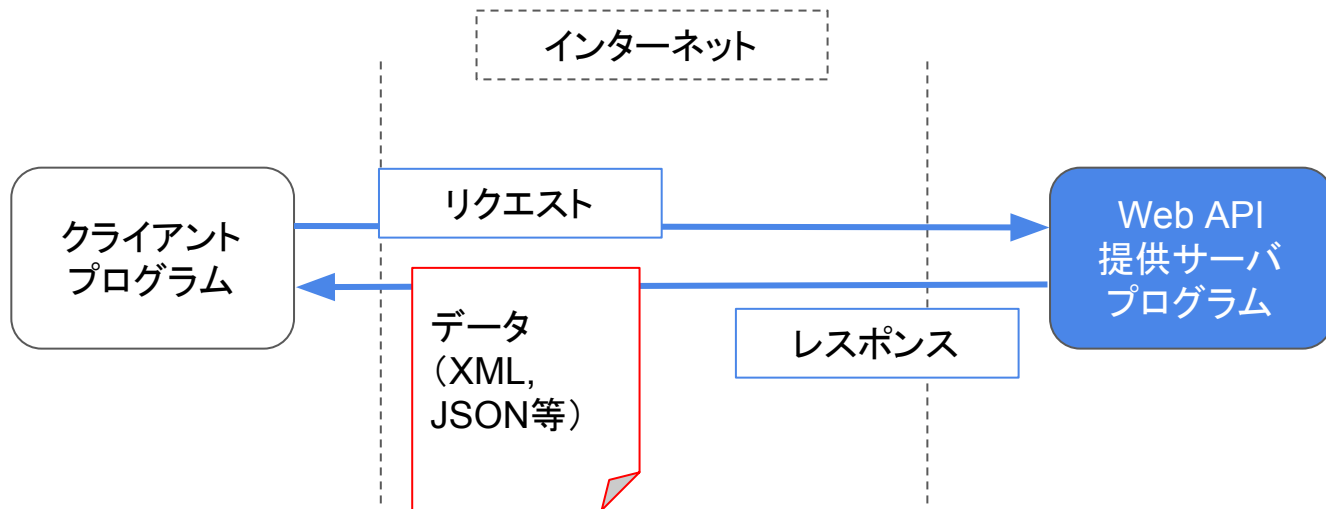
ページの表示に必要なパラメータ
パラメータ名: q
パラメータ値: web+api

APIとは

- Application Programming Interface
- アプリケーションソフトウェアの開発にあたって使われる、標準的なモジュール群、その利用法
 - API = 開発環境における標準
 - 典型的には、特定の OS環境のソフトウェアは共通する標準 APIを用いて開発される
 - Windows
 - Apple iOS

Web APIとは

- Web + API = Web API
- ウェブ上で他のサービスを呼び出す方式、取り決め
 - 標準的なデータやサービスの提供を行う方法



Web APIの歴史

- インターネット普及前には...
 - そもそも、同一環境(計算機や OS)同士で通信する仕組みはある程度必要
 - 遠隔のサービスを呼び出す仕掛け
 - ウェブ以前には: RPC(Remote Procedure Call)
 - オープン環境においても、通信コマンドやデータ形式の標準化(分野ごとに)
- これらの方式を受け継いで、Web上でも行うための手段
 - SOAP方式
 - ...一般的にはさほど普及しなかった

Web APIの歴史

- Web API: 別名 Web Servicesとも呼ばれる
- 2002年頃から一般的に使われ始める
 - Amazon, Googleなどが代表例
- きっかけはWeb 2.0
 - 2005/2006年頃に爆発的に流行
 - 当時流行っていて先進的なサービスをまとめた用語
 - その技術要素のひとつがAJAXという手法
 - その前提として必要なデータのやり取りの方式 = Web API
 - Google Mapsがその代表格
 - つまり、Google Maps APIが前提となって、AJAXを実現させた

実際のWeb APIのやり取り

The screenshot shows the CiNii Articles website in a web browser. The browser's address bar displays `https://ci.nii.ac.jp`. The website's header includes the CiNii logo and navigation links for "日本の論文をさがす", "大学図書館の本をさがす", and "日本の博士論文をさがす". On the right, there are links for "東大大学", "新規登録", "ログイン", and "English". A small notice at the top right states: "学協会向け論文電子化・公開サービス (NII-ELS) コンテンツの移行先について (更新)".

The main content area features the CiNii Articles logo and a search section with a teal background. The search section includes tabs for "論文検索", "著者検索", and "全文検索". Below these tabs is a search input field with the placeholder text "フリーワード" and a "検索" button. Under the input field are two buttons: "すべて" and "本文あり", with a "閉じる" link to the right.

Below the search section are several input fields for advanced search criteria:

- Title (タイトル): A single-line text input field.
- Author (著者名): A single-line text input field.
- Author ID (著者ID): A single-line text input field.
- Author Affiliation (著者所属): A single-line text input field.
- Publication Name (刊行物名): A single-line text input field.
- ISSN: A single-line text input field.
- Volume/Issue (巻号ページ): Three separate input fields labeled "巻", "号", and "ページ".
- Publisher (出版者): A single-line text input field.
- Reference (参考文献): A single-line text input field.
- Publication Year (出版年): Two input fields labeled "年" and "から", followed by a "年まで" label.

ci.nii.ac.jp/opensearch/search?q=図書館 X +

← → ↺ 🏠 ⓘ 🔒 https://ci.nii.ac.jp/opensearch/search?q=図書館&format=json

JSON 生データ ヘッダー

保存 コピー すべて折りたたむ すべて展開

```
@context:
  @vocab: "http://purl.org/rss/1.0/"
  rdf: "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  rdfs: "http://www.w3.org/2000/01/rdf-schema#"
  dc: "http://purl.org/dc/elements/1.1/"
  prism: "http://prismstandard.org/namespaces/basic/2.0/"
  opensearch: "http://a9.com/-/spec/opensearch/1.1/"
  @id: "https://ci.nii.ac.jp/opensearch/search?q=%E5%9B%B3%E6%9B%B8%E9%A4%A8&format=json&REMOTE_ADDR=130.69.96.13"
  @graph:
    0:
      @id: "https://ci.nii.ac.jp/opensearch/search?q=%E5%9B%B3%E6%9B%B8%E9%A4%A8&format=json&REMOTE_ADDR=130.69.96.13"
      @type: "channel"
      title: "CiNii OpenSearch - 図書館"
      description: "CiNii OpenSearch - 図書館"
      link:
        @id: "https://ci.nii.ac.jp/opensearch/search?q=%E5%9B%B3%E6%9B%B8%E9%A4%A8&format=json&REMOTE_ADDR=130.69.96.13"
        dc:date: "2018-10-25T13:28:43+09:00"
        opensearch:totalResults: "122160"
        opensearch:startIndex: "1"
        opensearch:itemsPerPage: "20"
      items:
        0:
          @id: "https://ci.nii.ac.jp/naid/120006404883"
          @type: "item"
          title: "新病院移転までの道のり～現在進行中～"
          link:
            @id: "https://ci.nii.ac.jp/naid/120006404883"
          rdfs:seeAlso:
            @id: "https://ci.nii.ac.jp/naid/120006404883.json"
          dc:creator:
```

<https://ci.nii.ac.jp/opensearch/search?q=%E5%9B%B3%E6%9B%B8%E9%A4%A8&format=json>

2つのやり取りの方式

- SOAP方式

- 呼び出すパラメータ類をXML形式のデータにまとめ、それを受け渡す方式
- W3C (World Wide Web Consortium) が規格化
- 企業による初期のWeb API提供形態

- REST方式

- 呼び出すパラメータ類はURL内のクエリとして発行し、データ結果のみを受け取る方式
- ブラウザからも簡単に呼び出したり、テストしたりできるため、今ではこちらが主流に

RESTとは

- Webの基本的なやりとり、コンテンツの取得(GET)、更新(POST)といったやりとりをまとめたお約束
- Web APIとの関連では:
 - コンテンツの取得はGETメソッドを使う
 - コンテンツの更新はPOSTメソッドを使う
 - といった、ブラウザとウェブサーバのやり取りの仕方を指す
- RESTの範囲を外れる例:
 - データの取得にPOSTを使ったり、クッキーを前提としたページ遷移を必要とするサービス
- GETリクエストとパラメータによって、必要なデータを取得する仕組み

URLとWeb APIリクエストの関係

- Web APIにおいてはURLが基本的な単位となって、それに対する「操作」(処理)がやり取りを決める
 - URLに様々なパラメータを与える
- 例:

`http://ci.nii.ac.jp/opensearch/search?`

`q=図書館&`

`format=json&`

`sortorder=7`

URLとパラメータ

- Web APIのサービス単位で異なる
- CiNii Articlesの場合
 - q=クエリ
 - format=データ形式
 - sortorder=ソート方式
 - 詳細はヘルプを参照 ...

「CiNii Articles論文検索のOpenSearch」ヘルプより

NII 学術コンテンツサービス サポート

CiNii全般 CiNii Articles CiNii Books CiNii Dissertations KAKEN NII-REO 日本語

CiNii Articles

CiNii Articles - メタデータ・API - CiNii Articles 論文検索のOpenSearch

ヘルプ

- CiNii Articlesについて
- ご利用区分について
- マニュアル
- よくある質問
- 旧学協会向け論文電子化・公開サービス (NII-ELS)
- メタデータ・API**

- CiNii Articles のOpenURL 受信機能
- CiNii Articles のOpenURL 送信機能
- CiNii Articles 論文検索のOpenSearch
- CiNii Articles 著者検索のOpenSearch
- CiNii Articles 全文検索のOpenSearch

目次

- クエリ仕様
- レスポンス仕様
 - RSS1.0 レスポンスフォーマット仕様
 - Atom1.0 レスポンスフォーマット仕様
 - JSON-LD レスポンスフォーマット仕様
- 改訂履歴

クエリ仕様

CiNii Articles論文検索のOpenSearchクエリは以下の形式です。
`http://ci.nii.ac.jp/openserach/search?(パラメータ=値)&(パラメータ=値)&…&(パラメータ=値)`です。

クエリを構成する各パラメータは以下のとおりです。

N	パラメータ	内容
1	q	フリーワードを指定するパラメータ（項6～18が指定されない場合は必須） このパラメータはURLエンコードされている必要がある。
2	count	ページあたりの検索結果件数を指定するパラメータ（任意） デフォルトは20。0以下もしくは200を超える値が指定された場合、もしくは指定されない場合はデフォルト設定とみなす。（200：論文検索結果画面の最大表示件数）

Web APIの事例

Web APIの種類

- 様々なサービスがWeb APIとして提供
 - 例えば、手元で使ったことがあるものだけでも、書籍検索、論文検索、画像検索、地図情報、音声認識、顔認識、など
 - Web APIの紹介サイトProgrammable-Webには、600近いカテゴリが用意され、13,000を超えるWeb APIサービスが紹介されている
 - <https://www.programmableweb.com/category>

Web APIの事例紹介

- CiNii (Articles/Books)
- NDLサーチ
- カーリル
- (Google Maps)
- (Amazon)

CiNii Articles

- 国立情報学研究所が提供する日本の論文検索サービス
 - <https://ci.nii.ac.jp/>
- 検索: OpenSearch
 - <https://ci.nii.ac.jp/opensearch/search?q=omeka&format=json>
 - 検索条件をパラメータとしてやり取りする
 - 検索結果はAtom形式(XML)やJSON形式等を選択可能
- 詳細レコード: RDF
 - <https://ci.nii.ac.jp/naid/130006407004.rdf>
 - 詳細レコードのデータをRDF形式で返す

ci.nii.ac.jp/opensearch/search?q=flickr X

https://ci.nii.ac.jp/opensearch/search?q=flickr&format=json 110% 検索

JSON 生データ ヘッダー

保存 コピー すべて折りたたむ すべて展開

```
@context:
  @vocab: "http://purl.org/rss/1.0/"
  rdf: "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  rdfs: "http://www.w3.org/2000/01/rdf-schema#"
  dc: "http://purl.org/dc/elements/1.1/"
  prism: "http://prismstandard.org/namespaces/basic/2.0/"
  opensearch: "http://a9.com/~spec/opensearch/1.1/"
@id: "https://ci.nii.ac.jp/opensearch/search?q=flickr&format=json&REMOTE_ADDR=126.167.161.41"
@graph:
  0:
    @id: "https://ci.nii.ac.jp/opensearch/search?q=flickr&format=json&REMOTE_ADDR=126.167.161.41"
    @type: "channel"
    title: "CiNii OpenSearch - flickr"
    description: "CiNii OpenSearch - flickr"
    link:
      @id: "https://ci.nii.ac.jp/opensearch/search?q=flickr&format=json&REMOTE_ADDR=126.167.161.41"
      dc:date: "2018-10-25T21:03:17+09:00"
      opensearch:totalResults: "68"
      opensearch:startIndex: "1"
      opensearch:itemsPerPage: "20"
    items:
      0:
        @id: "https://ci.nii.ac.jp/naid/130006407004"
        @type: "item"
        title: "デジタルアーカイブの「裾野のモデル」を求めて - 東京大学附属図書館U-PARL「古典籍on flickr! ~漢籍・法帖を写真サイトでオープンしてみると〜」報告"
        link:
          @id: "https://ci.nii.ac.jp/naid/130006407004"
        rdfs:seeAlso:
          @id: "https://ci.nii.ac.jp/naid/130006407004.json"
        dc:creator:
          0:
            @value: "富澤 かな"
          1:
            @value: "木村 拓"
          2:
            @value: "成田 健太郎"
          3:
            @value: "永井 正勝"
          4:
            @value: "中村 寛"
          5:
            @value: "福島 幸宏"
```

<https://ci.nii.ac.jp/opensearch/search?q=flickr&format=json>

この XML ファイルにはスタイル情報が関連づけられていないようです。以下にドキュメントツリーを表示します。

```
-<rdf:RDF>
  -<rdf:Description rdf:about="https://ci.nii.ac.jp/naid/130006407004#article">
    <rdf:type rdf:resource="http://purl.org/ontology/bibo/Article"/>
    <foaf:isPrimaryTopicOf rdf:resource="https://ci.nii.ac.jp/naid/130006407004.rdf"/>
  -<dc:title>
    デジタルアーカイブの「裾野のモデル」を求めて -東京大学附属図書館U-PARL「古典籍on flickr!～漢籍・法帖を写真サイトでオープンしてみると～」報告
  </dc:title>
  <dc:creator>富澤 かな</dc:creator>
  <dc:creator>木村 拓</dc:creator>
  <dc:creator>成田 健太郎</dc:creator>
  <dc:creator>永井 正勝</dc:creator>
  <dc:creator>中村 寛</dc:creator>
  <dc:creator>福島 幸宏</dc:creator>
  <dc:publisher>一般社団法人 情報科学技術協会</dc:publisher>
  <prism:publicationName>情報の科学と技術</prism:publicationName>
  <prism:issn>0913-3801</prism:issn>
  <prism:volume>68</prism:volume>
  <prism:number>3</prism:number>
  <prism:startingPage>129</prism:startingPage>
  <prism:endingPage>134</prism:endingPage>
  <prism:publicationDate>2018</prism:publicationDate>
  -<dc:description>
```

<https://ci.nii.ac.jp/naid/130006407004.rdf>

NDLサーチ

- 国立国会図書館が提供する検索サービス

- <http://iss.ndl.go.jp/>

- 検索: OpenSearch、SRU/SRW

- <http://iss.ndl.go.jp/api/openserach?title=%e3%83%9e%e3%83%aa%e3%83%bc%e3%82%a2%e3%83%b3%e3%83%88%e3%83%af%e3%83%8d%e3%83%83%e3%83%88&ndc=2>

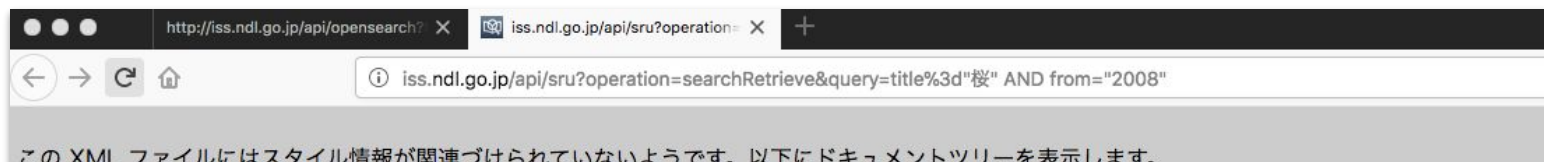
- <http://iss.ndl.go.jp/api/sru?operation=searchRetrieve&query=title%3d%22%e6%a1%9c%22%20AND%20from=%222008%22>

- データ取得系: OAI-PMH、SRU/SRW

- DCNL (NDL版Dublin Core (XML))

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <rss version="2.0" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3 <channel>
4 <title>2 マリーアントワネット - 国立国会図書館サーチ OpenSearch</title>
5 <link>http://iss.ndl.go.jp/api/openserach?ndc=2&title=%E3%83%9E%E3%83%AA%E3%83%BC%E3%82%A2%E3%83%B3%E3%83%88%E3%83%AF%E3%83%8D%E3%83%88&ndc=2
6 <description>Search results for ndc=2 title=マリーアントワネット</description>
7 <language>ja</language>
8 <openSearch:totalResults>115</openSearch:totalResults>
9 <openSearch:startIndex>1</openSearch:startIndex>
10 <openSearch:itemsPerPage></openSearch:itemsPerPage>
11 <item>
12 <title>10分で読める伝記</title>
13 <link>http://iss.ndl.go.jp/books/R100000002-I000011230787-00</link>
14 <description>
15 <![CDATA[<p>4年生,学研教育出版,9784052033865</p>
16 <ul><li>タイトル : 10分で読める伝記</li>
17 <li>タイトル (読み) : 10ブン デ ヨメル デンキ</li>
18 <li>責任表示 : 塩谷京子 監修,</li>
19 <li>NDC(9) : 280</li>
20 </ul>]]>
21 </description>
22 <author>塩谷京子 監修,塩谷, 京子,</author>
23 <category>児童書</category>
24 <guid isPermaLink="true">http://iss.ndl.go.jp/books/R100000002-I000011230787-00</guid>
25 <pubDate>Fri, 29 Jul 2011 09:00:00 +0900</pubDate>
26 <dc:title>10分で読める伝記</dc:title>
27 <dcndl:titleTranscription>10ブン デ ヨメル デンキ</dcndl:titleTranscription>
28 <dc:creator>塩谷, 京子</dc:creator>
29 <dcndl:creatorTranscription>シオヤ, キョウコ</dcndl:creatorTranscription>
30 <dcndl:volume>4年生</dcndl:volume>
31 <dc:publisher>学研教育出版</dc:publisher>
32 <dc:publisher>学研マーケティング</dc:publisher>
33 <dcterms:issued xsi:type="dcterms:W3CDTF">2011</dcterms:issued>
34 <dcndl:price>800円</dcndl:price>
35 <dc:extent>195p ; 21cm</dc:extent>
36 <dc:identifier xsi:type="dcndl:ISBN">9784052033865</dc:identifier>
37 <dc:identifier xsi:type="dcndl:NDLBibID">000011230787</dc:identifier>
38 <dc:identifier xsi:type="dcndl:JPN">21958058</dc:identifier>
39 <dc:identifier xsi:type="dcndl:NSMARCNO">123450400</dc:identifier>
```

<http://iss.ndl.go.jp/api/openserach?title=%E3%83%9E%E3%83%AA%E3%83%BC%E3%82%A2%E3%83%B3%E3%83%88%E3%83%AF%E3%83%8D%E3%83%88&ndc=2>



—<searchRetrieveResponse>

<version>1.2</version>

<numberOfRecords>8713</numberOfRecords>

<nextRecordPosition>201</nextRecordPosition>

—<extraResponseData>

<facets> <lst name="REPOSITORY_NO"> <int name="R100000002">4707</int> <int name="R100000001">2396</int> <int name="R000000004">220</int> <int name="R100000049">214</int> <int name="R000000024">96</int> <int name="R100000073">5</int> <int name="R100000040">26</int> <int name="R100000067">13</int> <int name="R100000094">9</int> <int name="R100000039">6</int> <int name="0">33</int> <int name="1">42</int> <int name="2">490</int> <int name="3">179</int> <int name="4">78</int> <int name="5">40</int> <int name="9">2337</int> </lst> <lst name="ISSUED_DATE"> <int name="1200">1</int> <int name="1915">1</int> <int name="1916">1</int> <int name="1968">1</int> <int name="1973">2</int> <int name="1974">1</int> <int name="1979">2</int> <int name="1980">1</int> <int name="1989">2</int> <int name="1990">1</int> <int name="1991">4</int> <int name="1992">1</int> <int name="1993">1</int> <int name="1998">1</int> <int name="1999">4</int> <int name="2000">4</int> <int name="2001">2</int> <int name="2002">3</int> <int name="2007">1</int> <int name="2008">873</int> <int name="2009">912</int> <int name="2010">844</int> <int name="2011">7</int> <int name="2015">792</int> <int name="2016">800</int> <int name="2017">656</int> <int name="2018">464</int> </lst> <lst name="LIBRARY"> <int name="ピエ図書館">752</int> <int name="岡山県立図書館">701</int> <int name="さいたま市立中央図書館">667</int> <int name="横浜市中央図書館">667</int> <int name="長崎県立長崎図書館">553</int> <int name="千葉市中央図書館">470</int> <int name="福岡市総合図書館">452</int> <int name="川崎市立図書館">394</int> <int name="滋賀県立図書館">388</int> <int name="福井県立図書館">342</int> <int name="福岡県立図書館">339</int> <int name="静岡県立中央図書館">290</int> <int name="静岡市立中央図書館">275</int> <int name="鹿児島県立図書館">264</int> <int name="名古屋市鶴舞中央図書館">264</int> <int name="府立図書館">253</int> <int name="北海道立図書館">253</int> <int name="福島県立図書館">246</int> <int name="愛知県図書館">242</int>

<http://iss.ndl.go.jp/api/sru?operation=searchRetrieve&query=title%3d桜 AND from=2008>

カーリル

- カーリル社が提供する図書館横断検索サービス
 - 独自のWeb API
- 検索・所蔵検索
 - http://api.calil.jp/check?isbn=4877511172&systemid=Tokyo_Ome&format=json
- データ形式
 - JSON形式(独自)

カーリル

```
callback({  
  "session": "ae6318869192394d0d2a62164ff4a1b3",  
  "books": {  
    "4877511172": {  
      "Tokyo_Ome": {  
        "status": "Cache",  
        "reserveurl": "https://www.lics-saas.nexs-service.jp/ome/webopac/searchdetail.do?biblioid=44433",  
        "libkey": {  
          "沢井": "貸出可",  
          "小曾木": "貸出可",  
          "中央": "貸出可",  
          "今井": "貸出可",  
          "大門": "貸出可"  
        }  
      }  
    }  
  },  
  "continue": 0  
});
```

その他のWeb API事例

- 興味深い事例としてはAmazon Web ServicesやTwitterなどのように主要な商用サービスにおけるWeb APIの展開例がある
- これらでは、自身のサービスをWeb API経由で取得、操作できる
 - Twitterクライアントの操作: ツイートの投稿、編集、リプライなど(BOT)
 - Amazonにおけるインフラ系ツールを Web APIから操作: サーバインフラの増強、ツールの追加

マッシュアップの事例

Web APIの応用例（マッシュアップ）

- Web APIの典型的な応用例（アプリケーション）
- マッシュアップとは？（Mashup）
 - 複数の機能を組み合わせて、可視化したり、異なる機能同士を有機的につなげて利用する
 - Web APIの応用アプリケーションは、Web API機能をほかの何かと組みわせて扱うことが多いことから「マッシュアップ」と呼ばれる

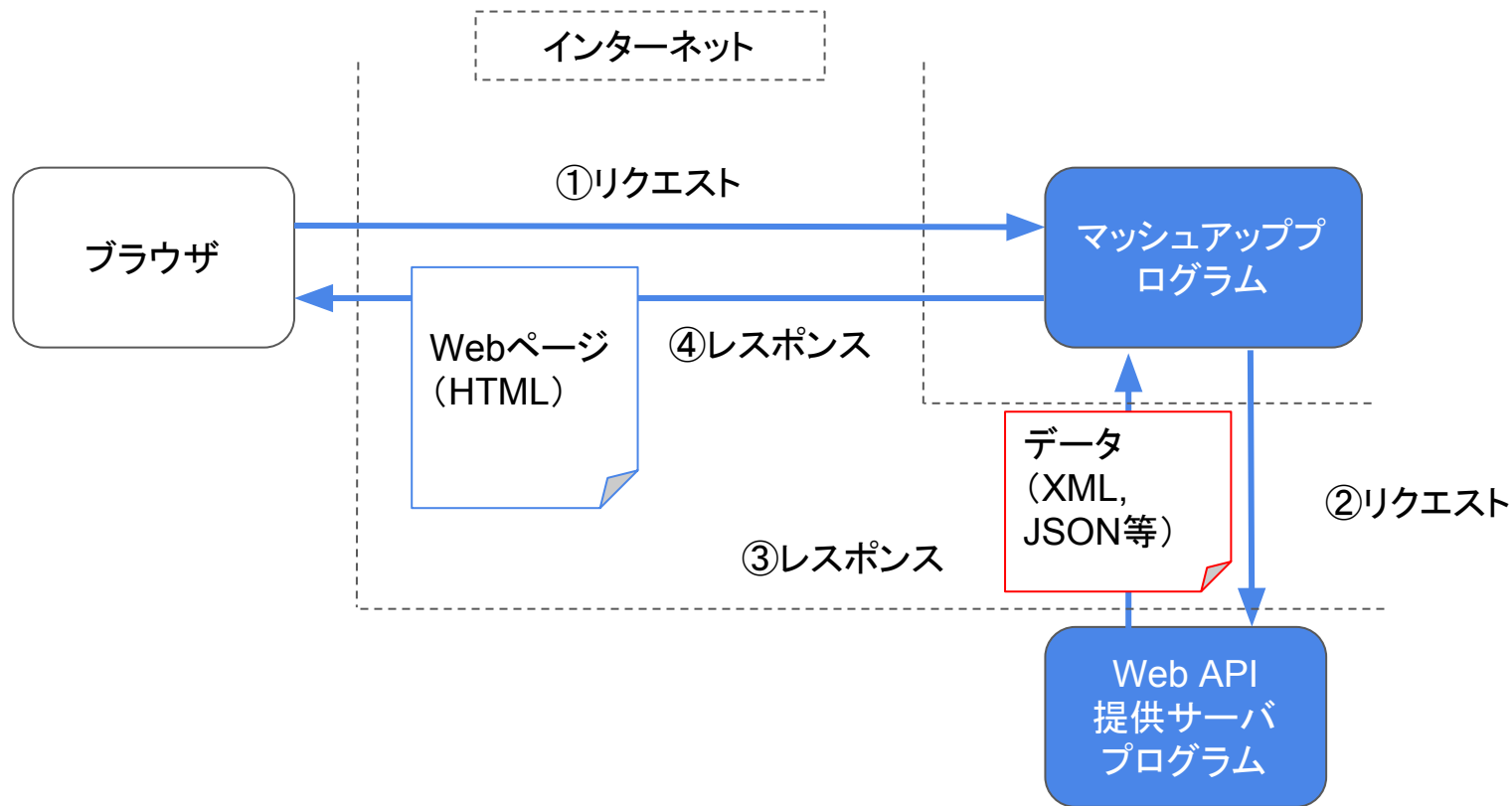
マッシュアップの種類

- (付加価値サービス)
- マッシュアップ(Mashup)
 - サーバサイド型マッシュアップ
 - Ajax型マッシュアップ
 - クライアントサイド
 - データ集約型マッシュアップ
 - ハーベスト型

サーバサイド型マッシュアップ

- 最も古典的なタイプの応用例
- 様々な類型がある
 - ひとつだけでなく、複数のサービスを組み合わせる
 - Ajax型との組み合わせも

サーバサイド型マッシュアップ(概念図)



論文ったー

<https://twitter.com/ronbuntter>

ホーム

モーメント

キーワード検索

アカウントをお持ちの場合 ログイン



論文ったー

@ronbuntter

空気を読んで論文を紹介するtwitter-botです。CiNii ウェブAPIとTwitterの日本トレンド（以前はbuzztterのRSS）を利用しています。詳細は
d.hatena.ne.jp/YamadaToshiyuk...
d.hatena.ne.jp/YamadaToshiyuk...
2009年9月に登録

ツイート

196,344

フォロー

6

フォロワー

7,611

いいね

872

フォローする

ツイート

ツイートと返信

メディア



論文ったー @ronbuntter · 2分
こんな論文どうですか？「観光の文脈」と「本来の文脈」：ケニア・サンプルの観光地への出稼ぎの事例より(中村 香子),2008



CiNii 論文 - 「観光の文脈」と「本来の文脈」：ケニア...
「観光の文脈」と「本来の文脈」：ケニア・サンプルの観光地への出稼ぎの事例より 中村 香子 日本文化人類学会研究大会発表要旨集 2008(0), 31-31, 2008
cl.nii.ac.jp

🗨️

🔄

❤️



論文ったー @ronbuntter · 17分
こんな論文どうですか？ テクノスコープ Technology File(046)ダイナミックプライシング ビッグデータで最適値付け：ヤフー 福岡ソフトバンクホークス リクルートホー-,2016



CiNii 論文 - テクノスコープ Technology File(046)ダイ...
テクノスコープ Technology File(046)ダイナミックプライシング ビッグデータで最適値付け：ヤフー 福岡ソフトバンクホークス リクルートホールディングス Airbnb...
cl.nii.ac.jp

🗨️

🔄

❤️

Twitterを使ってみよう

登録してあなたのタイムラインを作りましょう

アカウント作成

こちらもおすすめです · 更新



レファレンッター
@referentter



CiNii (さいにい)
@cinii_jp



国立情報学研究所(NII)
@johouken



岩波書店
@Iwanamishoten

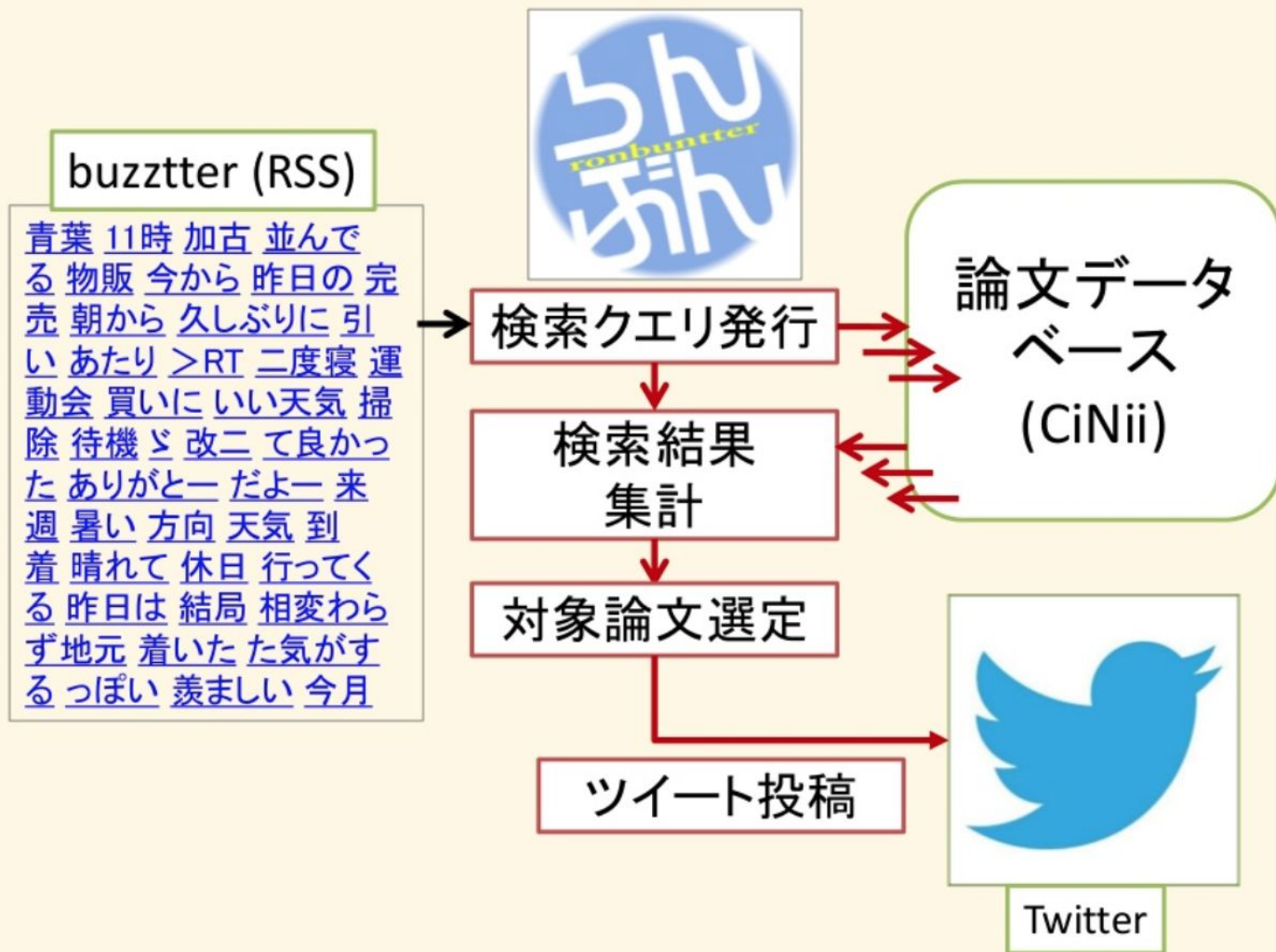


スドー
@stdaux

世界中のトレンド

41

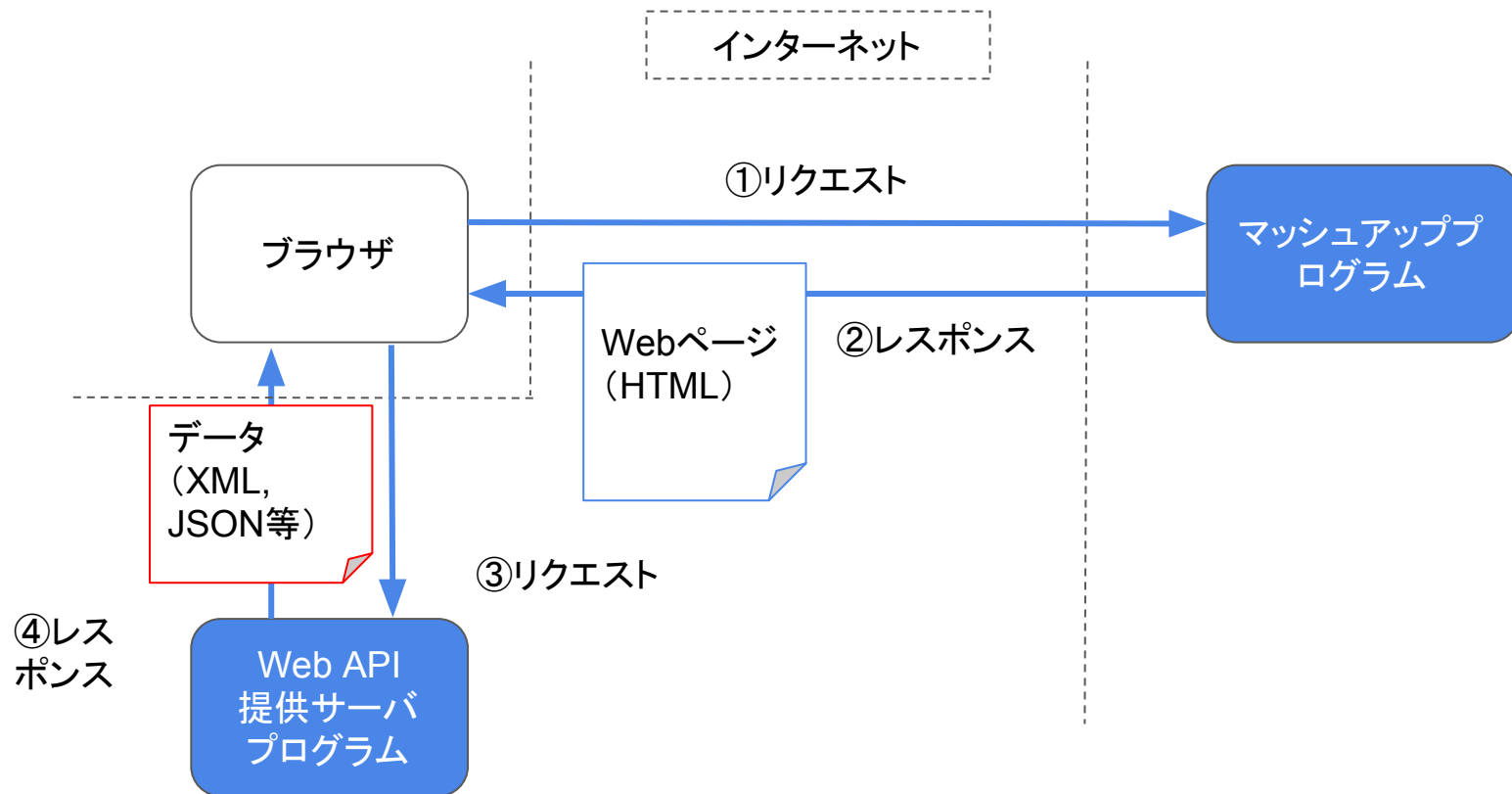
処理の 流れ



Ajax型マッシュアップ

- ブラウザ上で実行されるJavaScriptを主体としたアプリケーション
- サーバ側でまとめて実行された結果が返ってくるのではなく、個別のWeb APIへのリクエストがひとつずつ実行された分ごとに返ってくるため、見た目の実行時間を早める利点がある

Ajax型マッシュアップ(概念図)



日本最大の図書館検索

カーリル™

情報科学技術協会

さがす

千葉県立図書館を検索しています 変更する

▼図書館を絞り込み



検索スキルをみがく:検索
技術者検定3級 公式
テキスト

原田 智子/著、吉井
隆明/著、森 美由紀

蔵書なし

読みたい 読んだ



プロの検索テクニック:
検索技術者検定2級 公
式推奨参考書

原田 智子/著、小河
邦雄/著、清水 美都

蔵書なし

読みたい 読んだ



ひとりでできる特許調
査—特許のしくみと情
報検索虎の巻

情報科学技術協会OIG
特許分科会/著

蔵書なし

読みたい 読んだ



情報検索の基礎知識

原田 智子/著、小山
恵司/著、岸田 和明

蔵書なし

読みたい 読んだ



新サーチャー入門〈基
礎編〉

INFOSTA/編集、情報
科学技術協会/編集、

蔵書あり(貸出可)

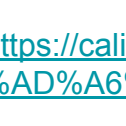
読みたい 読んだ



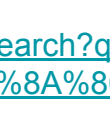
ビブリオバトル入門—
本を通して人を知る—



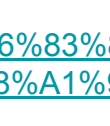
情報の管理と検索



インパクトファクター
を解き明かす



情報検索の知識と技術



情報検索の基礎知識

<https://calil.jp/search?q=%E6%83%85%E5%A0%B1%E7%A7%91%E5%AD%A6%E6%8A%80%E8%A1%93%E5%8D%94%E4%BC%9A>

データ集約型マッシュアップ

- Web APIを通じてデータ収集し活用できるようにしたマッシュアップサービス
- RSSリーダーなどはこの形態が多い
- 機関リポジトリなどにおけるメタデータハーベスティング
 - 複数のRSSサイト => RSSフィード閲覧サービス
 - 複数の機関リポジトリ => 論文の横断検索サービス

すべて

図書館

検索エリアの追加

すべて

本文あり

検索

クリア

検索結果56892 件 1 ~ 10 件を表示

1 | 2 | 3 | 4 | 5 | 6

次へ

最後

操作を選択

実行

全件選択

10件ずつ表示

出版年：新しい順に表示

☒

金戸文庫仮目録 / 金沢大学附属図書館
 9999-01-01, 金沢大学附属図書館
[金沢大学学術情報リポジトリKURA](#)
[金沢大学](#)
 概要
 出版年不明26cm ; 24p

☒

用和堂文庫[目録] / [第四高等学校図書館]
 9999-01-01, [第四高等学校図書館]
[金沢大学学術情報リポジトリKURA](#)
[金沢大学](#)
 概要
 用和堂文庫の手書き目録。作成年不明。28p

☒

[北之庄城郭図]
 9999
[福井県地域共同リポジトリ](#)
[福井県立](#)

<http://jairo.nii.ac.jp/>

Web APIとデータ形式

Web APIと標準化

- 様々な標準規格化への欲望
- 多様なサービス vs ニーズ vs 標準化
- 標準化の難しさ vs 標準の乱立
- 検索・データ取得用Web API標準の例
 - SRU, OpenSearch, OAI-PMH, OpenURL, SPARQL, ...

データ形式と標準化

- Web APIにおいては、基本的には、リクエストとレスポンスの2つの標準化がある
- レスポンス: 機械可読なデータ形式
- データ形式:
 - XML
 - JSON
 - TSV

XML (Extensible Markup Language) とは

- タグ要素とその内容テキストを指定し、機械的に読み取り可能とする形式
 - <親要素>
 - <要素 属性="値"></要素>
 - </親要素>
- なぜXMLか
 - 階層関係(木構造が入れ子になったデータ)を扱いやすい
 - 多言語対応(Unicode) : 日本語はじめ、多くの言語に対応
 - ほとんどのプログラミング環境に対応するライブラリが完備されていて、簡単に利用可能
- XMLの欠点
 - データ量が増える(タグ要素を入れ込む分)
- XMLの利点
 - メタ言語: 様々な形式の新しい種類の言語を規定できる
 - XHTML, HTML5, SVG, RSS, Atom, ...

JSON (JavaScript Object Notation) とは

- ブラウザ上で動作するJavaScript言語が標準対応する形式
 - XMLの複雑でデータ量が多くなる欠点を、タグを使わず、より簡易に書けるようにしたデータ形式

{

“著者” : [“中村覚”, “永崎研宣”],

“タイトル” : “日本国内のIIIF準拠画像に対する横断検索システムの構築

}

Web APIの課題

Web APIのご利益

- サービスの「プラットフォーム化」
 - 第三者にサービス提供する方式を標準化し、利用を促進できる
 - 近年の多デバイス環境下（PC、タブレット、スマートフォン）では、自サービスのプラットフォーム化が必然になってしまうという側面も ...
- サービスの多様化
 - Web APIにより、通常の利用形態を超えた利用を作り出すことができる

Web APIの寿命

- Web APIの永続性はそれほど長くないことに注意が必要
 - GoogleサーチAPI(2010年に終了)
 - Yahoo! Web検索API(2013年に終了)
 - Twitter(2013年に大幅な更新)
- 必然的に、マッシュアップ側の寿命に影響を与える！
- 利用する場合には
 - ある程度、代替サービスを念頭におく必要がある
 - Youtube vs ニコニコ動画
- 提供する場合には
 - 例えば、検索サービスを単純に更新するだけでも、Web APIにおいては大きな影響を与えることとなる
 - 最初に公開する際のリクエスト形式、データモデルの設計

Web APIの制限、認証

- Web API利用は機械可読(プログラムによるやり取り)が前提であるため、過負荷状態になる危険もある
 - 人手アクセスの数倍から数十倍
- 同時利用に制限をかけることが一般的
 - 1時間に1,000アクセスまで
 - 1時間に3,000アクセスまで
 - 1日に100,000アクセスまで、等
- 加えて、アクセス元のユーザを識別するために認証をつけることもある
 - Twitterなど

Web APIの提供にあたって

- URLとデータモデルの決定が非常に重要
 - どのURLにリクエストを送ると、どんなデータが戻ってくるか？
 - 「クールなURIは変わらない」
 - cf. <https://www.kanzaki.com/docs/Style/URI>
- Web API vs データファイルの提供
 - 数万件くらいの単純なデータなら、データファイル全件をダウンロードできるようにしておくだけでも十分
 - 逆に、データ量が膨大だったり、複雑な条件指定を行う必要があるものでは Web API化する恩恵も

ご静聴ありがとうございました。