

# 人間文化研究機構の新ポータルサイト nihuBridge の API Application Programming Interface (API) of nihuBridge, a new portal site of the National Institute for the Humanities

鈴木 卓治<sup>†</sup> 後藤 真<sup>†</sup> 関野 樹<sup>††</sup> 北岡 タマ子<sup>†††</sup> 堀 浩一<sup>†††</sup>

Takuzi SUZUKI<sup>†</sup> Makoto GOTO<sup>†</sup> Tatsuki SEKINO<sup>††</sup>  
Tamako KITAOKA<sup>†††</sup> and Koichi HORI<sup>†††</sup>

<sup>†</sup> 国立歴史民俗博物館

<sup>†</sup> National Museum of Japanese History

<sup>††</sup> 国際日本文化研究センター

<sup>††</sup> International Research Center for Japanese Studies

<sup>†††</sup> 人間文化研究機構

<sup>†††</sup> National Institutes for the Humanities

E-mail: <sup>†</sup>{suzuki, m-goto}@rekihaku.ac.jp <sup>††</sup>sekino@nichibun.ac.jp <sup>†††</sup>{t-kitaoka, hori}@nihu.jp

## 1. はじめに

人間文化研究機構（以下、機構）は、2022年6月に nihuBridge[1]を公開した。 nihuBridge は、機構が2008年以降公開・開発してきた研究資源共有化システム（nihuINT）[2]による統合検索機能を一部に引き継ぎつつ、新たな機能として、研究成果データの蓄積機能や DH (Digital Humanities) 関係リンク情報集約機能などを備えており、機構および連携諸機関が発信する多様な研究資源情報を共有・活用するためのポータルサイトと位置付けられている。

nihuBridge に新たに備えられた機能のひとつに nihuBridge API (Application Programming Interface) がある。本稿では、 nihuBridge API の位置付け、設計の概要、ならびに利用例を示す。 nihuBridge API は、機構から発信する様々なデータの機械的取得を可能にし、これまでに公開されてきた多くの研究データから新たな研究や計量分析を行うことを支援する。大学院生や新たに DH の研究に取り組もうとする研究者をはじめ、人文学、情報学等の様々な分野で nihuBridge が活用されることを希望している。

## 2. nihuBridge の構成

nihuBridge のトップページ（図1）には、機構のデジタル資源全体の“入り口”（ポータル）として、以下の5つのサービスが示されている：

1. 6機関データベースリンク：機構所属6機関が各機関で公開するデータベースの一覧およびアクセスリンク；
2. NIHU データカタログ：機構本部が公開するデータセットの一覧およびダウンロード機能；
3. NIHU 研究資源リンク：機構本部が提供する研究資源情報の一覧およびアクセスリンク；
4. 大型デジタル資源リンク：機構所属6機関が各

機関で運用する大型デジタル情報資源の一覧およびアクセスリンク；

5. 統合検索：機構本部および機構所属6機関が公開するデータベースの統合検索機能。

この5つのサービスはおおきく2つに大別できる。ひとつは、ポータル機能を実現するための、機構本部および所属6機関の公開情報へのリンク集（1., 3., 4.）であり、もうひとつは、直接的にデータの検索と提供をサービスする情報システム（2., 5.）である。

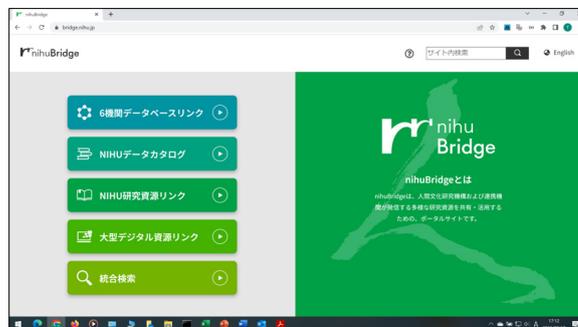


図1 nihuBridge のトップページ

## 3. nihuBridge API

nihuBridge API は、 nihuBridge の情報システムにおける情報資源、すなわち、(1)NIHU データカタログの登録データセット、ならびに、(2)統合検索の登録データベース、の検索・管理・運用を行うための“計算機向けインターフェイス”である。 Web ページとして用意された“人間向けインターフェイス”は、プログラム中で nihuBridge API を呼び出すことで実現されている。

この2つの情報資源は、それぞれ性格が異なる。 NIHU データカタログの登録データセットは、データセットごとに、その内容、フォーマット、利用条件、クラウド等に保存されたデータセットの実体（ファイ



表 1 nihuBridge のメタデータスキーマ

フィールド	項目名	データ型	Dublin Core に対応する基本記述要素	Schema.org に対応するタイプ
researchResourceId	研究資源 ID	文字列	identifier	identifier
doi	doi	文字列	identifier	identifier
originalId	オリジナル ID	文字列	identifier	identifier
status	状態	整数		creativeWorkStatus
title	タイトル	文字列	title	name
alternativeTitle	別タイトル	文字列	title	name
creator	作成者	文字列	creator	creator
contributor	編者・監修者	文字列	contributor	contributor
publisher	所蔵者・組織	文字列	publisher	publisher
subject	主題	文字列	subject	about
keyword	キーワード	文字列	subject	about
description	概要・説明	文字列	description	description
dateCreated	研究資源が作られた日	文字列	date	dateCreated
datePublished	登録日	文字列	date	datePublished
dateModified	最終更新日	文字列	date	dateModified
futureUpdates	今後の更新の有無	真偽		(creativeWorkStatus)
license	利用ライセンス	文字列	rights	license
type	研究データのタイプ	文字列	type	genre
archiveFormat	研究データのフォーマット(全体)	文字列	format	encodingFormat
contentSize	研究データのサイズ	文字列	format	contentSize
encodingFormat	研究データのフォーマット(個別)	文字列	format	encodingFormat
inLanguage	研究データの言語	文字列	language	inLanguage
isBasedOn	元になった資源	文字列	source	isBasedOn
link	研究データ	文字列	relation	url
sampleLink	サンプルデータ	文字列	relation	
metadataFields	メタデータの項目	文字列		
metadataLanguage	メタデータの記述言語	文字列		
relatedLink	関連データ	文字列		relatedLink
resources	上位の研究資源	文字列	relation	relatedLink
spatial	対象空間範囲	緯度経度範囲	coverage	spatial
temporal	対象時間範囲	日時範囲	coverage	temporal
comment	備考	文字列		
useCases	利用事例	文字列		
operationHistories	作業履歴	文字列		updateAction
accessCount	アクセス状況	整数		interactionStatistics

表 2 nihuBridge API 一覧 (一般利用者用)

名称	機能	GET/POST
<b>メタデータ検索・閲覧 (metadata)</b>		
メタデータ検索 (search)	メタデータを検索し、該当する研究資源の一覧を取得	POST
メタデータ検索(HIT数) (search-hits)	メタデータを検索し、該当する研究資源の集計一覧を取得	POST
メタデータ取得 (機能名省略, 研究資源 ID を与える)	メタデータの 1 つのレコードを取得	GET
メタデータ取得 (機能名省略)	メタデータの 1 つまたは複数のレコードを取得	POST
<b>データベース情報【統合検索のみ】 (databases)</b>		
データベース一覧取得 (list)	データベースの一覧を取得	POST
データベースグループ一覧取得 (group)	データベースグループの一覧を取得	POST

とがわかる。

「メタデータ検索(HIT数)」の API は、POST 通信による呼び出しであり、検索文字列等のパラメタは、JSON 形式で記述し、HTTP リクエストのメッセージボディに含めて送信する。

nihuBridge API は、API としての利便性を高める観点から、一般利用者用 API については CORS (Cross-

Origin Resource Sharing ; オリジン間リソース共有) を許可しており、利用者が自分で作成した Web ページから nihuBridge API の呼出しが可能となっている。

一般利用者向け nihuBridge API の利用法についての詳細は、nihuBridge のページ[4]にある API 仕様書, API リクエストサンプル, API チュートリアル, python プログラムソース集を参照されたい。

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="utf-8" />
</head>
<body>
  <H3>検索 JSON</H3>
  <pre id="query"></pre>
  <H3>結果 JSON</H3>
  <pre id="result"></pre>
  <script language="javascript" type="text/javascript">
    // API「メタデータ検索(HIT数)」の URL
    let targetURL = "https://api.bridge.nihu.jp/v1/integratedsearch/metadata/search-hits";
    // 検索 JSON 文字列
    let query_json = {
      "query": {
        "conditions": [
          {
            "query": {
              "term": "源義経"
            }
          }
        ],
      }
    };
    // 検索 JSON 文字列を表示
    let query_json_text = JSON.stringify(query_json, null, 2);
    document.getElementById("query").innerText = query_json_text;
    // console.log(json_text);
    // API 呼び出しオブジェクトの生成
    let testRequest = new XMLHttpRequest();
    // 呼び出し成功時の処理
    testRequest.onload = function () {
      // 結果 JSON 文字列を表示
      let result_json = JSON.parse(this.response);
      let result_json_text = JSON.stringify(result_json, null, 2);
      document.getElementById("result").innerText = result_json_text;
    }
    // 呼び出し失敗時の処理
    testRequest.onerror = function () {
      document.getElementById("result").innerText = 'Could not post';
    }
    // API 呼び出しの実行
    testRequest.open("POST", targetURL);
    testRequest.setRequestHeader('content-type', 'application/json');
    testRequest.send(query_json_text);
  </script>
</body>
</html>
```

図 4 Web ページからの nihuBridge API の呼出しの例

```
検索JSON
{
  "query": {
    "conditions": [
      {
        "query": {
          "term": "源義経"
        }
      }
    ]
  }
}

結果JSON
{
  "info": {
    "statusCode": 0,
    "total": 382,
    "databases": {
      "nijl_rekisi_jinbutu": 66,
      "nijl_kojiruien": 50,
      "nijl_kokubungakuronbun": 46,
      "nijl_mei_jisyuppankoukoku": 41,
      "nijl_kotensakutougouhyakka_hagajimeijiten": 30,
      "nmjh_kanzousiryuu": 23,
      "nijl_nihonkotenseki": 22,
      "rme_lib_books": 17,
      "nmjh_tosyomokuroku": 17,
      "nmjh_kanzousisakie": 15,
      "nijl_suzumakagami": 7,
      "rme_mofull": 6,
      "rme_trackinfo": 6,
      "rme_mcd-literature": 5,
      "rme_audcat": 4,
      "nmjh_nihonsyouden": 4,
      "nmjh_nihoninzokugakubunken": 3,
      "ircjs_heiankyoutomeisyozue": 2,
      "ircjs_zagainihonbijutu": 2,
      "nijl_archiv": 2,
      "nijl_kindeisyosi": 2,
      "nijl_syuzou_archive": 2,
      "rihn_syozoutosyo": 2,
      "ircjs_ir": 1,
      "nijl_ir": 1,
      "nijl_kohitugire": 1,
      "nijl_oususyozai": 1,
      "ninjal_nihongokenkyuu_nihongokoiku": 1,
      "ninjal_toshokanmokuroku": 1,
      "rme_mocat": 1,
      "nmjh_minzokugoi": 1
    }
  }
}
```

図 5 図 4 の実行結果

## 5. おわりに

人間文化研究機構の新しい DH ポータルサイト nihuBridge の“計算機向けインターフェイス”である nihuBridge API について紹介した。

研究活動における IT 技術の導入は、大量のデータを、人手を介さず、誤りなく処理することによる信頼性の向上、ならびにプログラム化などの手続きの可視化による客観性の確保に寄与する。伝統的に人間の介在する要因が大きい人文学研究の分野においても、これらの利点を生かしたデジタル人文学 (Digital Humanities; DH) の進展が強く期待されている。そのためには現時点では基礎的な部分、すなわち、機械的処理に耐える“きれいな”データを大量かつ低コストで確保すること、あるいは、一つの検索の結果を使って別の検索を自動的・連鎖的に実行して必要な情報を収集することなど、人文学研究に資するデジタル研究資源・研究環境の確立を急がねばならない。nihuBridge はその一翼となるべく設計・実現されたものであり、たゆまぬ検討と改良を経て、一歩ずつ目標に近づいていきたい。

なお、本稿の内容は、じんもんこん 2022 のプレイベントとして開催された第 1 回人間文化研究機構 DH 研究会における発表[5]をもとに再構成したものである。

## 文 献

- [1] <https://bridge.nihu.jp/> (2023 年 7 月 24 日確認)。
- [2] [https://www.nihu.jp/ja/research/archive/cip/source\\_system/](https://www.nihu.jp/ja/research/archive/cip/source_system/) (2023 年 7 月 24 日確認)。
- [3] 鈴木卓治：歴史系博物館とメタデータ，T3-2，画像電子学会第 49 回年次大会予稿集(2021)。
- [4] <https://www.nihu.jp/ja/database/nihubridge/> (2023 年 7 月 24 日確認)。
- [5] <http://jinmoncom.jp/sympo2022/program.html#P1>，DH プラットフォーム nihuBridge の研究活用の可能性 ～API 機能を一例として (2022 年 12 月 9 日；パネラー：北岡タマ子，後藤真，関野樹，鈴木卓治，宮川創，菊池信彦)，第 1 回人間文化研究機構 DH 研究会 (プレイベント 1)，情報処理学会人文科学とコンピュータシンポジウム (じんもんこん 2022)，人文科学とコンピュータ研究会 Web サイト (2023 年 7 月 24 日確認)。