

JaLC2 システム
外部接続インタフェース仕様書
第 2.3 版

2023 年 5 月 31 日

国立研究開発法人 科学技術振興機構
ジャパンリンクセンター（JaLC）事務局

改版履歴

#	版名	発行日	改訂内容
1	第 1.0 版	2015/2/3	初版作成
2	第 1.1 版	2015/2/17	初版の修正 2.2.2, 2.3.2, 2.4.2, 2.5.2, 2.6.2, および 2.7.2 におけるファイル送信時パラメータの記述 を修正。
3	第 1.2 版	2015/7/1	研究データのメタデータ拡張(メタデータ スキーマ v2.0)に合わせて 2.4 を修正。
4	第 1.3 版	2015/11/10	2.9 エラー発生例を追記。 3.2.6 を新設し、API の動作例を追記。
5	第 1.4 版	2018/2/6	DOI 名削除運用中止に伴う関連記述につい て取消線適用
6	第 1.5 版	2018/11/29	<ul style="list-style-type: none"> ・ 1.2. JaLC で提供するインタフェースに、 表 1-3 一般向けインタフェース一覧 (DOI コンテンツ検索) を追記。 ・ 1.4. 一般向けインタフェース (DOI コン テンツ検索) の利用者 を追記。 ・ 4. 一般向けインタフェース (DOI コン テンツ検索) を追記。 付録に、「別紙 4 DOI コンテンツ検索での レスポンス(RDF XML 形式)」、「別紙 5 DOI コンテンツ検索でのレスポンス (Citeproc JSON 形式)」を追加。
7	第 1.6 版	2018/12/28	非同期デポジット対応により別紙 1-7 レス ポンス_メッセージボディ (XML) 【登録】 を更新、別表 1-8 レスポンス_メッセージボ ディ (XML) 【照会】追加。 2.9 コンテンツ情報登録照会 HTTP 共通追 加。
8	第 1.7 版	2019/4/10	4.一般向けインタフェース (DOI コンテン ツ検索) の、4.2.1 リクエスト形式において、 HTTP リクエストヘッダを追加。 <ul style="list-style-type: none"> ・ 4.3.1 レスポンス形式 (1)RDF XML 形式のレスポンス仕様におい

			<p>て、多バイト文字をエスケープするように変更。</p> <p>(2)Citeproc JSON 形式のレスポンス仕様において、多バイト文字をエスケープしないように変更。</p>
9	第 1.8 版	2021/1/13	<p>2.8. レスポンスの例とエラー発生時の対処方法の表 2-1 エラー事象ごとに想定される原因に、詳細情報を追加</p>
10	第 1.9 版	2021/2/17	<ul style="list-style-type: none"> ・「2.8. レスポンスの例とエラー発生時の対処方法」記載の同期デポジットと非同期デポジットの内容を変更し、「2.1 共通事項」「(4) リクエストに対する処理方法について」に移動。 ・同期デポジット及び非同期デポジットフロー図を追加。 ・「1.3. 外部提供インタフェース（コンテンツ検索系）の想定利用者と利用条件」に認証キーと IP アドレスの設定方法を追記。
11	第 2.0 版	2021/10/13	<p>IF 仕様に関する特記事項に「BOM 無し」を追記</p>
12	第 2.1 版	2022/7/27	<ul style="list-style-type: none"> ・「2.1.共通事項」「(3) コンテンツの登録・更新・削除のリクエストチェックについて」のリクエストの具体的な指定方法を削除 ・「2.1.共通事項」「(5) HTTP リクエストメッセージボディ」に HTTP リクエストメッセージボディについての説明を追加 ・「2.1.共通事項」「(6) XML における HTML 修飾タグの利用」に XML での HTML 修飾タグについての説明を追加 ・「2.1.共通事項」「(7) XML における値が無い場合のタグの書き方」に XML での空要素タグについての説明を追加 ・以下の章を削除し、「2.2. コンテンツ情報登録 HTTP 共通」に統合 <p>2.2. コンテンツ情報登録 HTTP 共通（ジャーナルアーティクル）</p> <p>2.3. コンテンツ情報登録 HTTP 共通（書籍）</p>

			<p>2.4. コンテンツ情報登録 HTTP 共通（研究データ）</p> <p>2.5. コンテンツ情報登録 HTTP 共通（e-learning）</p> <p>2.6. コンテンツ情報登録 HTTP 共通（汎用データ）</p> <p>・「別紙 5 DOI コンテンツ検索でのレスポンス(Citeproc JSON 形式)」にエスケープする文字コードについて追加</p>
13	第 2.2 版	2023/1/25	<p>・ 図 2-8 レスポンスの例（<errcd>返却値発生時）に errmsg タグを追加</p> <p>・ 表 2-1 エラー事象ごとに想定される原因の#5 について下記の記載を変更 「UTF-8（BOM なし）」以外の形式 ↓ 「UTF-8」以外の形式</p> <p>・ 2.2.1. インタフェース概要 (3) IF 仕様に関する特記事項の記載内容から（BOM 無し）を削除</p> <p>・ 2.5.1. インタフェース概要 (3) IF 仕様に関する特記事項の記載内容から（BOM 無し）を削除</p> <p>・ 3.4.1. インタフェース概要 (3) IF 仕様に関する特記事項の記載内容から（BOM 無し）を削除</p> <p>・ 3.5.1. インタフェース概要 (3) IF 仕様に関する特記事項の記載内容から（BOM 無し）を削除</p>
14	第 2.3 版	2023/5/31	<p>・ 表 21 エラー事象ごとに想定される原因の#3 の「想定される原因」にアカウントロックに関する記載を追加</p> <p>・ 2.4. レスポンスの例とエラー発生時の対処方法に追記</p>

目次

1.	はじめに	7
1.1.	本資料について	7
1.2.	JaLC2 で提供するインタフェース	7
1.3.	外部提供インタフェース（コンテンツ検索系）の想定利用者と利用条件	9
1.4.	一般向けインタフェース(DOI コンテンツ検索)の利用者	9
2.	外部提供インタフェース（コンテンツ登録系）	10
2.1.	共通事項	10
2.2.	コンテンツ情報登録 HTTP 共通	15
2.2.1.	インタフェース概要	15
2.2.2.	リクエスト	15
2.2.3.	レスポンス	16
2.2.4.	サンプル	16
2.3.	HTTP リクエスト時の送信例	17
2.4.	レスポンスの例とエラー発生時の対処方法	19
2.4.1	正常に終了した場合のレスポンス	19
2.4.2	異常終了した場合のレスポンス	20
2.4.3	JaLC Web での履歴確認	22
2.4.4	コンテンツ情報登録照会 API での履歴確認	23
2.5.	コンテンツ情報登録照会 HTTP 共通	24
2.5.1.	インタフェース概要	24
2.5.2.	リクエスト	24
2.5.3.	レスポンス	24
3.	外部提供インタフェース（コンテンツ検索系）	25
3.1.	共通事項	25
3.2.	DOI 名_メタデータ検索 HTTP 共通	26
3.2.1.	インタフェース概要	26
3.2.2.	リクエスト	26
3.2.3.	レスポンス	28
3.2.4.	JaLC2 内での検索手順	28
3.2.5.	サンプル	30
3.2.6.	本 API と「DOI 名_タイトル著者名検索 HTTP 共通」の違い	31
3.3.	DOI 名_タイトル著者名検索 HTTP 共通	32
3.3.1.	インタフェース概要	32
3.3.2.	リクエスト	32

3.3.3.	レスポンス	33
3.3.4.	JaLC2 内での検索手順.....	33
3.3.5.	サンプル.....	34
3.4.	DOI 名_研究者 ID 検索 HTTP 共通	35
3.4.1.	インタフェース概要	35
3.4.2.	リクエスト.....	35
3.4.3.	レスポンス	36
3.4.4.	JaLC2 内での検索手順.....	36
3.4.5.	サンプル.....	37
3.5.	書誌情報 XML_DOI 名検索 HTTP 共通	38
3.5.1.	インタフェース概要	38
3.5.2.	リクエスト.....	38
3.5.3.	レスポンス	38
3.5.4.	JaLC2 内での検索手順.....	39
3.5.5.	サンプル.....	39
4.	一般向けインタフェース (DOI コンテンツ検索)	40
4.1.	概要 (全体イメージ)	40
4.2.	リクエスト	41
4.2.1.	リクエスト形式.....	41
4.2.2.	curl コマンドを利用した実行例.....	41
4.3.	レスポンス	42
4.3.1.	レスポンス形式.....	42
4.3.2.	RDF XML 形式と Citeproc JSON 形式のレスポンス相違点.....	42
4.3.3.	RDF XML 形式のレスポンスサンプル	43
4.3.4.	Citeproc JSON 形式のレスポンスサンプル	45

<付録>

- 別紙 1 コンテンツ登録系 API リクエスト・レスポンス
- 別紙 2 コンテンツ検索系 API リクエスト・レスポンス
- 別紙 3 コード一覧表
- 別紙 4 DOI コンテンツ検索でのレスポンス(RDF XML 形式)
- 別紙 5 DOI コンテンツ検索でのレスポンス(Citeproc JSON 形式)

1. はじめに

1.1. 本資料について

本資料は、JaLC2 システムに対して、コンテンツを登録・更新したり、メタデータから DOI を検索・取得したり、DOI から書誌 XML を取得する際の外部接続インタフェース仕様について記述したものである。

なお、JaLC2 では、OAI-PMH や LinkedOpenData による情報一括提供の仕組みも用意しているが、これらのインタフェース仕様については本資料の対象外とする。

※本資料に記載の機能については、予告無しに変更することがある。

1.2. JaLC2 で提供するインタフェース

JaLC2 システムでは、表 1-1、表 1-2 に示す外部提供インタフェースおよび、表 1-3 に示す一般向けインタフェースを用意している。

表 1-1 外部提供インタフェース一覧（コンテンツ登録、照会系）

#	インタフェース名称	概要	使用 プロトコル	入力 形式	出力 形式
1	コンテンツ情報登録 HTTP 共通(ジャーナルアーティク ル)	ジャーナルアーティクル形式の XML フォーマットのデータを JaLC2 に登録・更新するための インタフェース	HTTP POST	XML	XML
2	コンテンツ情報登録 HTTP 共通(書籍)	書籍形式の XML フォーマット のデータを JaLC2 に登録・更新 するためのインタフェース	HTTP POST	XML	XML
3	コンテンツ情報登録 HTTP 共通(研究データ)	研究データ形式の XML フォー マットのデータを JaLC2 に登 録・更新するためのインタフェ ース	HTTP POST	XML	XML
4	コンテンツ情報登録 HTTP 共通(e-learning)	e-learning 形式の XML フォー マットのデータを JaLC2 に登 録・更新するためのインタフェ ース	HTTP POST	XML	XML
5	コンテンツ情報登録 HTTP 共通(汎用データ)	汎用データ形式の XML フォー マットのデータを JaLC2 に登 録・更新するためのインタフェ ース	HTTP POST	XML	XML
6	コンテンツ情報登録照会 HTTP 共通	非同期デポジットの処理状況 を確認するインタフェース	HTTP POST		XML

表 1-2 外部提供インタフェース一覧（コンテンツ検索系）

#	インタフェース名称	概要	使用 プロトコル	入力 形式	出力 形式
1	DOI 名_メタデータ検索 HTTP 共通	コンテンツのタイトル、著者、ISSN、巻、号、開始頁、発行年などのメタデータを指定して検索を行い、該当するコンテンツの DOI を返却するインタフェース (メタデータは完全一致検索)	HTTP GET	URL	XML
2	DOI 名_タイトル著者名検索 HTTP 共通	コンテンツのタイトル、著者等を指定して検索を行い、該当するコンテンツの DOI を返却するインタフェース (メタデータは前方一致検索)	HTTP GET	URL	XML
3	DOI 名_研究者 ID 検索 HTTP 共通	研究者 ID を元にコンテンツ検索を行い、該当するコンテンツの DOI を返却するインタフェース	HTTP GET	URL	XML
4	書誌情報 XML_DOI 名検索 HTTP 共通	DOI で問い合わせを行い、該当するコンテンツの書誌情報を返却するインタフェース	HTTP GET	URL	XML

表 1-3 一般向けインタフェース一覧（DOI コンテンツ検索）

#	インタフェース名称	概要	使用 プロトコル	入力 形式	出力 形式
1	DOI コンテンツ検索	DOI で問い合わせを行い、該当するコンテンツのメタデータを返却するインタフェース	HTTP GET	URL	XML /JSON

1.3. 外部提供インタフェース（コンテンツ検索系）の想定利用者と利用条件

平成 26 年 12 月（JaLC2 リリース）時点では、表 1-2 に示す JaLC2 の外部提供インタフェース（コンテンツ検索系）の利用者は下記を想定している。（＝一般向けには開放していない）

- JaLC 正会員

また利用条件は下記となる。

- JaLC2 システム利用登録を行い、インタフェースの利用に必要な認証キー、ID、パスワードを知っていること
- 利用開始時に登録した IP アドレスの端末等からインタフェースへアクセスすること

※ 認証キー及び IP アドレスは JaLC システムにログイン後、「各種管理機能」・「利用機関情報管理」にてユーザ ID の検索後に表示される利用機関情報詳細画面で設定及び確認が可能である。

利用機関情報詳細画面の IP 認証フラグが未認証の場合、IP アドレスの登録は不要。

1.4. 一般向けインタフェース(DOI コンテンツ検索)の利用者

表 1-3 に示す 一般向けインタフェース（DOI コンテンツ検索）は、アクセス制限は掛けず、一般向けに開放する。

2. 外部提供インタフェース（コンテンツ登録系）

本章では、API を用いたコンテンツ登録や、登録されているコンテンツのメタデータ更新を行うためのインタフェース仕様について説明する。

2.1. 共通事項

(1) コンテンツの DOI 登録先の指定について

JaLC2 では、DOI の登録先として JaLC のほか、ジャーナルアーティクル・書籍については CrossRef を、研究データについては DataCite をそれぞれ登録先とすることが可能である。

ただし、登録先 DOI は、事前に DOI プレフィックス登録時に指定しておく必要がある。外部提供インタフェースを用いる時のリクエストメッセージで登録先を指定することはできない。

(2) コンテンツのメタデータ更新について

登録済コンテンツのメタデータ更新については、コンテンツ登録と同じインタフェースで実行可能である。

登録時に用いたリクエストに対し、変更のあるメタデータ項目を変えて再登録することで、JaLC データベース側のメタデータ更新となる。

(3) コンテンツの登録・更新のリクエストチェックについて

コンテンツの登録・更新リクエスト時は、ログイン ID に応じて、適切な ID からのリクエストかどうかチェックする。このチェックでは、ログイン ID、出典機関(サイト ID)、および DOI プレフィックスの関係を確認している。

例えば、あるログイン ID からのコンテンツ登録が、他の利用者に紐づく DOI プレフィックスである場合は、登録処理は行われずエラーとなる。また、コンテンツ更新リクエストが、他の利用者からコンテンツ登録されたものである場合は、更新処理は行われずエラーとなる。

(4) リクエストに対する処理方法について

リクエストに対する処理方法には「同期デポジット」と「非同期デポジット」がある。

- 同期デポジット

リクエスト受付と同時にシステムで登録処理を実行し処理結果をレスポンスする。

受付から登録までを一貫して行うので処理時間は長くなるが、レスポンスで登録結果を確認可能である。

登録用 XML の<result_method> (結果通知方法)に 0 を指定すると同期デポジットで処理する。

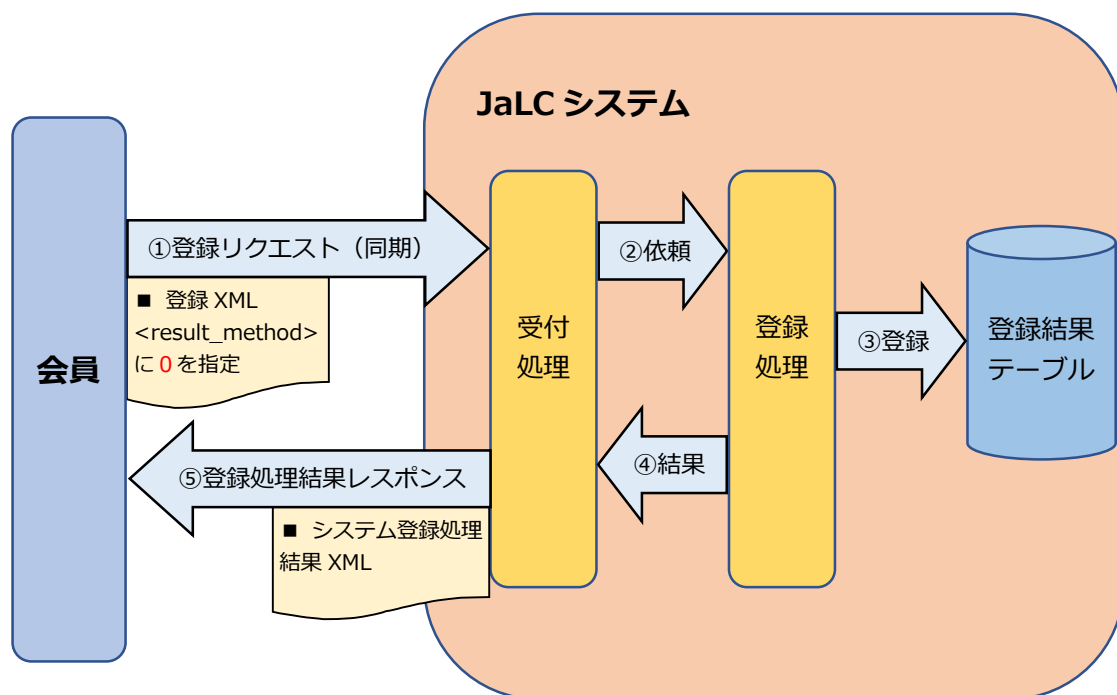


図 2-1 同期処理フロー

● 非同期デポジット

非同期デポジットはリクエスト受付処理と登録処理を分けて実行する。登録処理は 30 分ごとにバッチで実行する。

非同期デポジットでは受付処理結果をレスポンスするため処理時間は短い、登録結果は別途、照会を行う必要がある。

レスポンスには<exec_id>(実行 ID)の値が含まれており、この値を使用してデポジットの処理状況 (未デポジット、処理待ち、処理済み) を取得、確認可能である。(項番 2.4、2.5 参照)

登録用 XML の<result_method> (結果通知方法)に 2 を指定すると非同期デポジットで処理する。

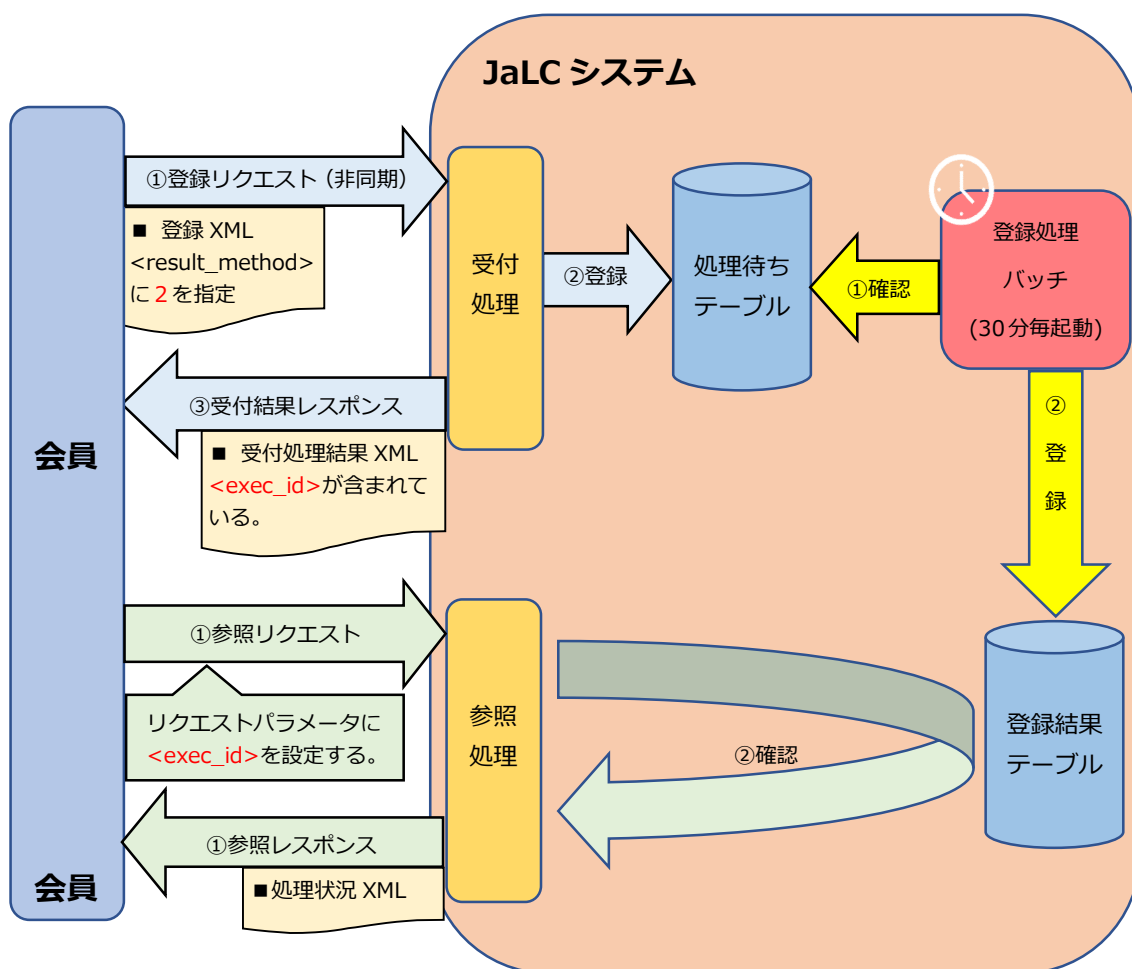


図 2-2 非同期処理フロー

(5) HTTP リクエストメッセージボディ

送信するパラメータについて、HTTP リクエストメッセージボディの利用が指示されている箇所について、以下のように指定する。

※<ログイン ID>、<パスワード>を例とする。

<HTML での例>

```
<html>
  <head></head>
  <body>
    <form name="form1" action="<送信先 URL>" method="POST">
      ログイン ID<input name="login_id" type="text" value="<ログイン ID>"><br/>
      パスワード<input name="login_passwd" type="password" value="<パスワード>"><br/>
      <input type="submit" value="送信">
    </form>
  </body>
</html>
```

<curl コマンドでの例>

```
curl -X POST <送信先 URL> ¥
-F 'login_id=<ログイン ID>' ¥
-F 'login_passwd=<パスワード>'
```

(6) XML における HTML 修飾タグの利用

XML ファイルで指定する値には、HTML における修飾タグ（<i><sup>等）をそのまま使用することはできない。

文字修飾が必要な場合には、エスケープするか CDATA で囲む方法を使用すること。

<タイトルとして「Sample <i>Value</i>」をエスケープした例>

```
<title>Sample&lt;i&gt;Value&lt;/i&gt;</title>
```

<タイトルとして「Sample <i>Value</i>」を CDATA で囲んだ例>

```
<title><![CDATA[Sample<i>Value</i>]]></title>
```

(7) XML における値が無い場合のタグの書き方

必須ではない要素において、指定する値がない場合は該当する要素自体の記載を避けること。

必須となる要素や、値がない要素を記載する必要がある場合は「<element></element>」と記載とすること。

<著者や作成者の所属機関として指定する値がない場合の例>

```
<affiliation></affiliation>
```

※空要素タグを表す「<element/>」という書き方は行わないこと。

2.2. コンテンツ情報登録 HTTP 共通

2.2.1. インタフェース概要

(1) 概要

ジャーナルアーク、書籍、研究データ、e-learning、汎用データの各種情報を XML 形式で JaLC2 に登録するためのインタフェースである。

(2) 送信先 URL

<https://japanlinkcenter.org/jalc/infoRegistry/registDataReceive/index>

(3) IF 仕様に関する特記事項

- 文字コードは、UTF-8 とする。
- 使用するフォーマットは XML 形式とする。
- 1 リクエストで送信するコンテンツは複数件を許容する。
- 使用するメソッドは POST とし、データの送信については、multipart/form-data 形式で行う。

2.2.2. リクエスト

HTTP リクエストメッセージボディに以下を指定する。

※URL パラメータに<ログイン ID>、<パスワード>を指定する方式は廃止を計画しております。

<HTTP リクエストメッセージボディに設定するパラメータ>

項番	パラメータ	必須	内容	備考
1	login_id	○	ログイン ID を設定する。	
2	login_passwd	○	パスワードを設定する。	
3	fname	○	デポジットするファイル (パス含む) を指定する。	

※HTTP リクエストメッセージボディの設定については 2.1 (5) を参照

デポジットするファイルは XML フォーマットである。

リクエスト時の XML フォーマット仕様を下記に示す。

- ジャーナルアーティクル情報
 - 別表 1-1-1 リクエスト_メッセージボディ(XML)【ジャーナルアーティクル・通常公開】
 - 別表 1-1-2 リクエスト_メッセージボディ(XML)【ジャーナルアーティクル・早期公開】
 - 別表 1-1-3 リクエスト_メッセージボディ(XML)【ジャーナルアーティクル・本公開】
- 書籍情報
 - 別表 1-2 リクエスト_メッセージボディ(XML)【書籍】
- 研究情報
 - 別表 1-3 リクエスト_メッセージボディ(XML)【研究データ】
- e-learning 情報
 - 別表 1-4 リクエスト_メッセージボディ(XML)【e-learning】
- 汎用データ情報
 - 別表 1-5 リクエスト_メッセージボディ(XML)【汎用データ】

リクエスト時の HTTP トランザクション例を 2.3 に示す。

2.2.3. レスポンス

別表 1-7 レスポンス_メッセージボディ (XML)【登録】を参照のこと。

2.2.4. サンプル

以下にある、zip ファイルを参照すること。

https://japanlinkcenter.org/top/service/service_technical.html#s003

2.3. HTTP リクエスト時の送信例

2.1 (5) および 2.2 で示したインタフェースの HTTP リクエスト送信例として、Apache JMeter(5.0)を利用して HTTP リクエストを送信する際の設定例を図 2-3 に、リクエスト時の HTTP トランザクション例（ボディ部）を図 2-4 に示す。

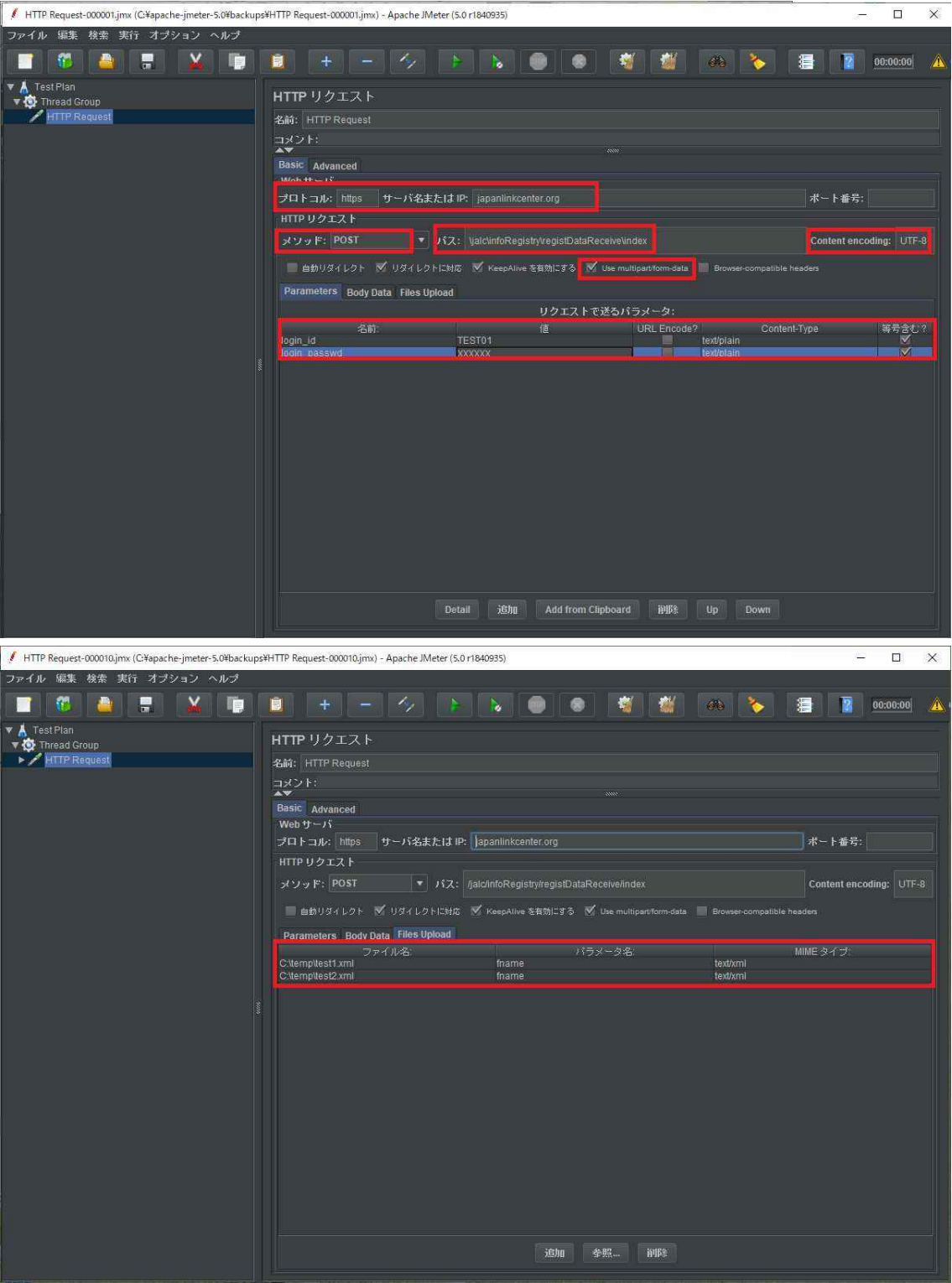


図 2-3 Apache JMeter 使用時の HTTP リクエスト送信設定例

```
POST (送信先 URL)

POST data:

(ログイン ID)
--wN-hB-ZFGMt3HojRT85Oy3WrJ7Ns0H6RaAZhoCc
Content-Disposition: form-data; name="login_id"
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 8bit

(パスワード)
--wN-hB-ZFGMt3HojRT85Oy3WrJ7Ns0H6RaAZhoCc
Content-Disposition: form-data; name="login_passwd"
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 8bit

--wN-hB-ZFGMt3HojRT85Oy3WrJ7Ns0H6RaAZhoCc
Content-Disposition: form-data; name="fname"; filename="research_data_v1.21.xml"
Content-Type: text/xml
Content-Transfer-Encoding: binary

<actual file content, not shown here>
--wN-hB-ZFGMt3HojRT85Oy3WrJ7Ns0H6RaAZhoCc--

[no cookies]

Request Headers:
Connection: keep-alive
Content-Length: 10076
Content-Type: multipart/form-data;
boundary=wN-hB-ZFGMt3HojRT85Oy3WrJ7Ns0H6RaAZhoCc; charset=UTF-8
Host: (送信先の IP アドレス)
User-Agent: Apache-HttpClient/4.2.6 (java 1.5)
```

図 2-4 リクエスト時の HTTP トランザクション (ボディ部) の例
(Apache JMeter 利用時)

注：JMeter は例示するために使用したツールであり、JMeter を使用してシステム構築することは想定していない (例: 複数 XML ファイルの連続処理は不可能などの仕様のため)。

2.4. レスポンスの例とエラー発生時の対処方法

2.1 共通事項で説明した通り、リクエストに対する処理方法には「同期デポジット」と「非同期デポジット」があり、レスポンスが異なる。

2.4.1 正常に終了した場合のレスポンス

同期デポジットの登録・更新が正常に終了した場合のレスポンスの例を図 2-5 に示す。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<root>
  <head>
    <totalcnt>1</totalcnt>
    <okcnt>1</okcnt>
    <ngcnt>0</ngcnt>
  </head>
  <body>
    <result>
      <seqno>0000000000000001</seqno>
      <resultstatus>1</resultstatus>
      <doi>test001/test201</doi>
    </result>
  </body>
</root>
```

図 2-5 同期デポジット レスポンスの例（正常終了時）

登録・更新が正常に終了した場合は、<head>内の<totalcnt>（＝コンテンツの総数）と<okcnt>（＝正常終了したコンテンツの件数）が一致し、かつ<ngcnt>（＝異常終了したコンテンツの件数）が 0 件、となる。また、<body>内には、リクエスト時に指定したコンテンツのシーケンス番号<seqno>ごとに、<resultstatus>に 1（新規登録）、2（更新）いずれかの値が格納される。

非同期デポジットの登録・更新が正常に終了した場合のレスポンスの例を図 2-6 に示す。

```
<root>
  <head>
    <totalcnt>0</totalcnt>
    <okcnt>0</okcnt>
    <ngcnt>0</ngcnt>
    <exec_id>12345 </exec_id>
  </head>
</root>
```

図 2-6 非同期デポジット レスポンスの例（正常終了時）

2.4.2 異常終了した場合のレスポンス

コンテンツの登録・更新が異常終了した場合は、レスポンスの HTTP ステータスコードのエラー、レスポンスの XML の<head>内のエラー、レスポンスの XML の<body>内のエラー、などが返却される。

レスポンスのエラー事象ごとに想定される原因の例を表 2-1 に示す

表 2-1 エラー事象ごとに想定される原因

#	API 利用時のレスポンスのエラー事象	想定される原因
1	HTTP ステータスコードが 200 以外	<ul style="list-style-type: none"> ● 送信先 URL が間違っている ● HTTP クライアントで定義したメソッドが間違っている ● fname に指定したファイルが存在しない
2	HTTP ステータスコードが 200	
3	レスポンスの<head>内の<errcd>の返却値が”*” (認証エラー)	<ul style="list-style-type: none"> ● リクエストパラメータの ID とパスワードの組合せが間違っている ● アクセス元の IP アドレスが JaLC に登録されている IP アドレスと異なっている ● 認証操作の複数回失敗によりアカウントがロックされている
4	レスポンスの<head>内の<errcd>の返却値が”#” (フォーマットエラー)	<ul style="list-style-type: none"> ● fname で指定した XML ファイルの構造に誤りがある (具体的なケース) <ul style="list-style-type: none"> ・ <head>, <body>, <error_process>, <result_method>, <content_classification>, <request_kind>, <site_id> タグのいずれかが無い ・ <error_process>, <result_method>, <content_classification>, <request_kind>, <site_id> タグのいずれかに値が無い ・ <error_process>, <result_method>, <content_classification>, <request_kind> タグのいずれかの値が定義外 ● login_id, login_passwd, fname のパラメータに未設定のものがある
5	レスポンスの<head>内の<errcd>の返却値が”+” (その他エラー)	<ul style="list-style-type: none"> ● fname で指定したファイルが XML ファイルでない ● XML ファイルの保存形式が「UTF-8」以外の形式になっている
6	レスポンスの<body>内の<resultstatus>の返却値が”4” (エラー)	<ul style="list-style-type: none"> ● 本来あるべきタグが存在しない ● 文字数制限を超えている

(1) 受付時エラー

HTTP ステータスコードが 200 以外または、レスポンスの<head>内に<errcd>が返された場合は、表 2-1 の#1 から 5 を参照する。

図 2-7 にエラーのためレスポンスの<head>内に<errcd>が発生した例を示す。

※この場合は JaLC Web ログイン後の「一括 DOI メタデータ情報登録」の「登録情報履歴」には残らない

```
<root>
  <head>
    <totalcnt>1</totalcnt>
    <okcnt>0</okcnt>
    <ngcnt>1</ngcnt>
    <errcd>*</errcd>
    <errmsg>ID またはパスワードが正しくありません。</errmsg>
  </head>
  <body/>
</root>
```

図 2-7 レスポンスの例（受付時エラー 発生時）

(2) 受付後のエラー

受付後のチェックにおいて、本来あるべきタグが存在しない、文字数制限を超えている等の理由で登録エラーとなった場合、レスポンスの<body>内の<resultstatus>として「4」が返される。

図 2-8 に 2 件のコンテンツを登録しようとして、1 件は正常終了、もう 1 件は異常終了した場合のレスポンスの例を示す。

```
<root>
  <head>
    <totalcnt>2</totalcnt>
    <okcnt>1</okcnt>
    <ngcnt>1</ngcnt>
  </head>
  <body>
    <result>
      <seqno>0000000000000001</seqno>
      <resultstatus>2</resultstatus>
      <doi>test001/test201</doi>
    </result>
    <result>
      <seqno>0000000000000002</seqno>
      <resultstatus>4</resultstatus>
      <doi>test001/test202</doi>
    </result>
  </body>
</root>
```

図 2-8 レスポンスの例（2 件目が異常終了）

<resultstatus>として「4」が返された場合に、そのエラーの詳細を確認するには、以下の方法がある。

①JaLC Web での履歴確認

②コンテンツ情報登録照会 API での確認（非同期デポジットの場合のみ）

それぞれの確認方法を 2.4.3、2.4.4 に示す。

2.4.3 JaLC Web での履歴確認

JaLC Web ログイン後の「一括 DOI メタデータ情報登録」の「登録情報履歴」でエラー件数の確認と登録結果情報ファイル（res.xml）、エラーデータファイル（err.xml）のダウンロードが可能である。

※同期デポジットでタイムアウトした場合にも確認可能

登録結果情報ファイルには登録・更新時のレスポンスが、エラーデータファイルにはエラーの詳細が記載されている。

res.xml と err.xml の紐づけはシーケンス番号で実施する。

「登録情報履歴」画面のイメージは以下。



2.4.4 コンテンツ情報登録照会 API での履歴確認

非同期デポジットの場合、実行 ID を指定してエラー件数、エラー詳細が確認可能。レスポンス例を図 2-9 に示す。

コンテンツ情報登録照会 API の説明は 2.5 を参照のこと。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<root>
  <head>
    <exec_id>999999999999</exec_id>
    <status>2</status>
    <exec_time>20990101223610</exec_time>
    <totalcnt>2</totalcnt>
    <okcnt>1</okcnt>
    <ngcnt>1</ngcnt>
  </head>
  <body>
    <result>
      <seqno>0000000000000001</seqno>
      <resultstatus>2</resultstatus>
      <doi> test001/test201</doi>
    </result>
    <result>
      <seqno>0000000000000002</seqno>
      <resultstatus>4</resultstatus>
      <doi> test001/test202</doi>
      <errinfo>
        <id>EC0501</id>
        <message>タイトルを設定して下さい。</message>
      </errinfo>
      <errinfo>
        <id>EC0506</id>
        <message>設定された出版地の値が不正です。</message>
      </errinfo>
    </result>
  </body>
</root>
```

図 2-9 照会 API レスポンスの例（2 件目が異常終了）

2.5. コンテンツ情報登録照会 HTTP 共通

2.5.1. インタフェース概要

(1) 概要

非同期デポジットの処理状況を確認するインタフェースである。

(2) 送信先 URL

<https://japanlinkcenter.org/jalc/infoRegistry/registDataResult/index>

(3) IF 仕様に関する特記事項

- 文字コードは、UTF-8 とする。
- 使用するフォーマットは XML 形式とする。
- 1 リクエストで送信する実行 ID は 1 件のみとする。
- 使用するメソッドは POST で行う。

2.5.2. リクエスト

HTTP リクエストメッセージボディに以下を指定する。

※URL パラメータに<ログイン ID>、<パスワード>を指定する方式は廃止を計画しております。

<HTTP リクエストメッセージボディに設定するパラメータ>

項番	パラメータ	必須	内容	備考
1	login_id	○	ログイン ID を設定する。	
2	login_passwd	○	パスワードを設定する。	
3	exec_id	○	実行 ID を設定する。	非同期デポジット時に返却される ID を設定する。

※HTTP リクエストメッセージボディの設定については 2.1 (5) を参照

2.5.3. レスポンス

別表 1-8 レスポンス_メッセージボディ (XML)【照会】を参照のこと。

3. 外部提供インタフェース（コンテンツ検索系）

本章では、JaLC2 に対して、DOI やメタデータ等で検索・問合せを行い、該当する情報を取得するインタフェース仕様について説明する。

3.1. 共通事項

(1) 検索時に指定するパラメータの正規化

検索時に指定するパラメータのうち、下記のものについては、英数文字を下記形式に変換（正規化）して検索処理を行う。

表 3-1 正規化の内容

項番	変換対象	変換前	変換後	備考
1	英数記号、スペース	全角	半角	
2		小文字	大文字	

表 3-2 パラメータごとの正規化する／しないの区分

項番	パラメータ	内容	正規化	備考
1	key	認証キー	しない	
2	contentsClassification	コンテンツ区分	しない	
3	journalTitle	資料名	しない	
4	title	タイトル	する	
5	creator	著者名(作成者名)	する	
6	researcherIdType	研究者 ID タイプ	しない	
7	researcherIdCode	研究者 ID コード	しない	
8	issn	ISSN	しない	
9	volume	巻	する	
10	issue	号	する	
11	page	開始ページ	する	
12	year	発行年	する	
13	isbn	ISBN	しない	
14	seriesTitle	シリーズタイトル	する	
15	chapterTitle	チャプタータイトル	する	
16	subject	主題	する	
17	publisher	出版社(者)	する	
18	multipleHits	要求区分	しない	
19	doi	DOI	しない	

3.2. DOI 名_メタデータ検索 HTTP 共通

3.2.1. インタフェース概要

(1) 概要

コンテンツのタイトル、著者、ISSN、巻、号、開始頁、発行年などのメタデータを指定して検索を行い、該当するコンテンツの DOI を返却するインタフェースである。

本 API にてリクエストパラメータで指定する各種メタデータは、一部正規化を行った上で、パラメータごとに完全一致検索で対象コンテンツの有無を判定する。一方、「3.3 DOI 名_タイトル著者名検索 HTTP 共通」（後述）でもリクエストパラメータにタイトル・著者を指定して検索できるが、「DOI 名_タイトル著者名検索 HTTP 共通」は前方一致検索で対象コンテンツの有無を判定する点が本 API とは異なる。（3.2.6 参照）

(2) 送信先 URL

<https://japanlinkcenter.org/jalc/linkSakiAnnaiOto/doiBookMetadataSearch/index>

(3) IF 仕様に関する特記事項

- プロトコルは、HTTP GET。
- 文字コードは、UTF-8 とし、日本語を URL パラメータに設定する際はエンコードを実施する。
- 1 リクエストのクエリ数は 1 件とする。
- 上限は HTTP GET の文字数制限 2083 文字。問合せ、回答ともに同じ。
- 結果が複数存在する場合、改行コード(¥n)で 1 件の区切りとする。

3.2.2. リクエスト

(1) リクエスト形式

<送信先 URL>?key=<key>&contentsClassification=<contentsClassification>
&journalTitle=<journalTitle>&title=<title>&creator=<creator>
&researcherIdType=<researcherIdType>&researcherIdCode=<researcherIdCode>
&issn=<issn>&volume=<volume>&issue=<issue>&page=<page>&year=<year>
&isbn=<isbn>&seriesTitle=<seriesTitle>&chapterTitle=<chapterTitle>
&subject=<subject>&publisher=<publisher>&multipleHits=<multipleHits>

(2) リクエストパラメータ

表 3-3 リクエストパラメーター一覧 を参照のこと。

表 3-3 リクエストパラメーター一覧 (DOI 名_メタデータ検索)

項番	パラメータ	ジャーナル アーティクル	書籍	研究データ	e-learning	汎用データ	内容	備考
1	key	○	○	○	○	○	認証キーを記述する。	
2	contentsClassification	●	●	●	●	●	コンテンツ区分を設定する。	01:ジャーナル アーティクル 02:書籍 03:研究データ 04:e-learning 99:汎用データ
3	journalTitle	●	-	-	-	-	資料名を記述する。	
4	title	●	●	●	●	●	タイトルを記述する。	
5	creator	●	●	●	●	●	著者名(作成者名)を記述する。	
6	researcherIdType	●	●	●	●	●	研究者 ID タイプを設定する。	
7	researcherIdCode	●	●	●	●	●	研究者 ID コードを設定する。	
8	issn	●	-	-	-	-	ISSN を記述する。	
9	volume	●	-	-	-	-	巻名を記述する。	
10	issue	●	-	-	-	-	号名を記述する。	
11	page	●	-	-	-	-	開始ページを記述する。	
12	year	●	●	●	●	●	発行年を記述する。	
13	isbn	●	●	-	-	-	ISBN を記述する。	
14	seriesTitle	-	●	-	●	-	シリーズタイトルを記述する。	
15	chapterTitle	-	●	-	-	-	チャプタータイトルを記述する。	
16	subject	-	-	●	-	-	主題を記述する。	
17	publisher	-	●	●	●	●	出版社(者)を記述する。	※1
18	multipleHits	○	○	○	○	○	要求区分を記述する。	"01"(1 件) "02"(複数件)

○…必須 ●…少なくとも●印のパラメータのうちいずれか1つ以上必須

※1 ジャーナルアーティクルのリクエストに publisher を入れても検索条件として使用しない。

※2 パラメータごとの複数值指定は不可。

※ 項番 18 について：

要求区分を"01 (1 件)"と指定した場合で該当するコンテンツが複数ある場合は、エラーメッセージ「EI0130:条件に該当するデータが最大表示件数を超過しています。条件を変更し、再度検索してください。」となる。(どの DOI も返却しない)。

要求区分を"02 (複数件)"と指定した場合は、該当する 1 件または複数件の DOI を返却する。

3.2.3. レスポンス

別表 2-1 レスポンス_メッセージボディ (XML) 【DOI 名_メタデータ検索】を参照のこと。

3.2.4. JaLC2 内での検索手順

- IP アドレス認証

リクエスト送信元の IP アドレスが JaLC に登録されているものと一致しない場合は、エラーメッセージ「EC0087 : IP アドレスが一致しません」を返す。

- 項目チェック

リクエストパラメータの項目をチェックする。

- 認証キー（必須）が指定されていない場合は、エラーメッセージ「認証キーを入力してください」を返す。
- 認証キーが不正（認証に失敗）のときは、エラーメッセージ「EC0085 : ID またはパスワードが正しくありません」を返す。
- 要求区分（必須）が指定されていない場合は、エラーメッセージ「EC0010 : 要求区分を入力してください」を返す。また、要求区分の指定値が不適当な場合は、エラーメッセージ「EC0160 : ○○は範囲外の値です」を返す。
- それ以外の項目は規定の文字数を超えた場合、エラーメッセージ「EC0040 : ○○は XX 文字以内で入力してください」を返す。

- リクエストパラメータの正規化

パラメータの英数文字・スペースは、全角を半角に、小文字を大文字に変換（正規化）する。（3.1 共通事項参照）

- ISSN, ISBN, 資料名(journalTitle)による資料の特定

リクエストパラメータに ISSN、ISBN または、資料名の何れかがある場合、ISSN、ISBN または、資料名より JaLC に存在する資料の ID（JaLC 内部で保持している値）を取得する。

リクエストパラメータに ISSN、ISBN、資料名がある場合は、ISSN・ISBN、資料名の順に資料が 1 件に特定されるまで検索を行う。

本処理にて資料の ID が特定できなかった場合、資料情報存在チェックエラーとし、エラーメッセージ「条件に該当するデータがありません」を返す。

ただし、ISSN、ISBN、資料名の内、ISBN のみ設定されていた場合はエラーとせず、引き続き後続の処理を実施する。

- 資料 ID、書誌メタデータによるコンテンツの特定

上記で取得した資料 ID（値が存在する場合）、及びリクエストパラメータの書誌メタデータを元にコンテンツ検索テーブル等を検索し、コンテンツの ID（JaLC 内部で保持している値）を取得する。

以下のコンテンツ特定処理フローにしたがって、コンテンツ特定を行う。

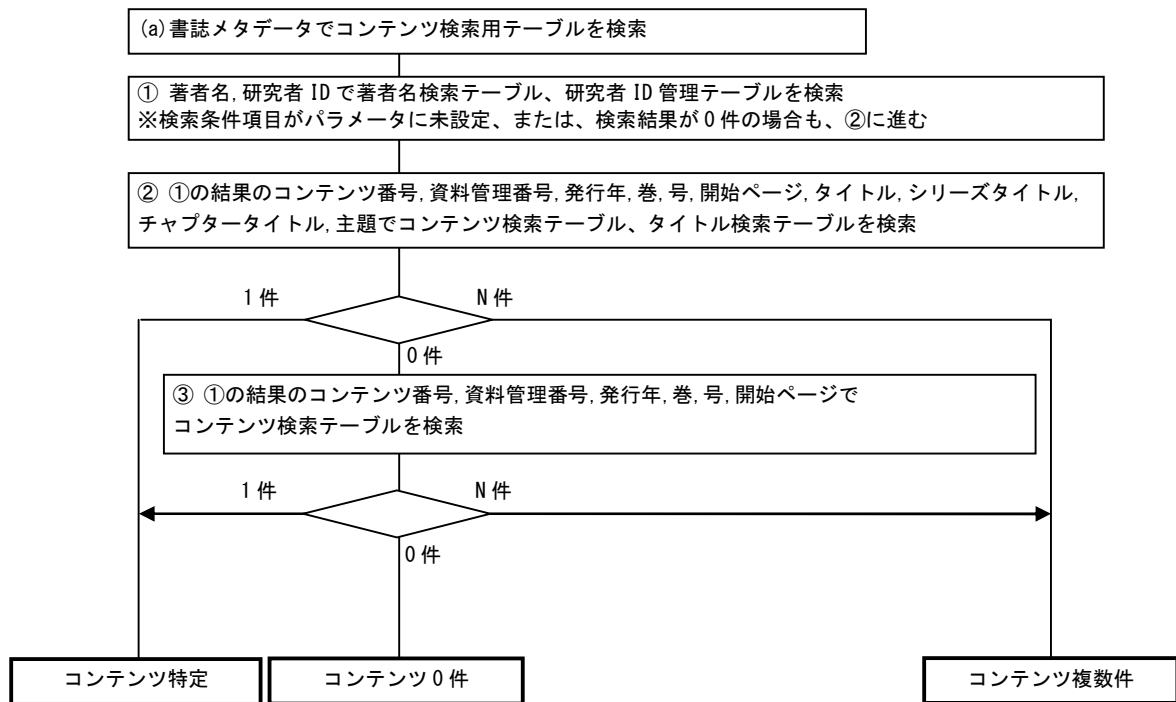


図 3-10 資料 ID、書誌メタデータによるコンテンツの特定

➤ 資料 ID（値が存在する場合）、発行年、巻、号、開始ページ、タイトル、シリーズタイトル、チャプタータイトル、主題、著者名、研究者 ID、出版社、ISBN で検索用テーブル等を検索する。

➤ 検索の結果、コンテンツの ID を取得できなかった場合、コンテンツ情報存在チェックエラー(メッセージ ID : EI0100「条件に該当するデータがありません。」)とし、CrossRef に対して検索処理を行う。

① 著者名検索テーブル、研究者 ID 管理テーブルを検索

著者名、研究者 ID で著者名検索テーブル、研究者 ID 管理テーブルを検索する。

② コンテンツ検索テーブル、タイトル検索テーブルを検索

①の検索結果のコンテンツの ID、資料管理番号（値が存在する場合）、発行年、巻、号、開始ページ、出版社、タイトル、シリーズタイトル、チャプタータイトル、主題でコンテンツ検索テーブル、タイトル検索テーブルを検索する。

検索結果が 0 件の場合、③の処理を実行する。

検索結果が 1 件、または、複数件の場合、リクエストパラメータより取得した要求区分を元に、回答判定（全て回答／エラー）を行う。エラーの場合は、エラーメッセージ「EI0130：条件に該当するデータが最大表示件数を超過しています」を返す。

③ タイトル検索テーブルをはずして、コンテンツ検索テーブルを検索

①の検索結果のコンテンツの ID、資料の ID (値が存在する場合)、発行年、巻、号、開始ページでコンテンツ検索テーブルを検索する。

検索結果が 0 件の場合、コンテンツ情報存在チェックエラー(メッセージ ID : EI0100「条件に該当するデータがありません。」)とし、CrossRef に対して検索処理を行う。

➤ 検索結果が 1 件、または、複数件の場合、リクエストパラメータより取得した要求区分を元に、回答判定 (全て回答/エラー) を行う。

- レスポンス作成

取得した情報からレスポンスメッセージボディ(XML)を作成する。検索結果全件分、処理を繰り返す。

3.2.5. サンプル

- タイトル : Frictional Coefficient under Banana Skin

https://japanlinkcenter.org/jalc/linkSakiAnnaiOto/doiBookMetadataSearch/index?key=*****&title=Frictional%20Coefficient%20under%20Banana%20Skin&multipleHits=02

⇒ 1 件ヒット

- コンテンツ区分 : 01、ジャーナルタイトル : Tribology Online、巻 : 7、号 : 3

https://japanlinkcenter.org/jalc/linkSakiAnnaiOto/doiBookMetadataSearch/index?key=*****&contentsClassification=01&journalTitle=Tribology%20Online&volume=7&issue=3&multipleHits=02

⇒ 17 件ヒット (同一の巻・号の記事が 17 件あるため)

3.2.6. 本 API と「DOI 名_タイトル著者名検索 HTTP 共通」の違い

タイトルと著者名のみを検索時に指定した場合の挙動について補足する。

本 API では、まず著者名で完全一致検索を行う。検索時に指定する著者名は「姓 名」もしくは「名 姓」（姓と名の間に半角スペースを入れる）とする。ここで該当するコンテンツが見つかったら、次にタイトルで完全一致検索を行う。タイトルが一致するコンテンツがある場合はそのコンテンツを回答する。一方一致するコンテンツがない場合は著者名が一致したコンテンツを全件回答する。

一方、「DOI 名_タイトル著者名検索 HTTP 共通」では、指定したタイトルの前方一致、かつ指定した著者名の前方一致で、該当するコンテンツがあるか検索を行う。検索時に指定する著者名は「姓」のみ、「名」のみ、「姓 名」、「名 姓」（姓と名の間に半角スペースを入れる）のいずれかでよい。この条件で該当するコンテンツがある場合は、そのコンテンツを全件回答する。

3.3. DOI 名_タイトル著者名検索 HTTP 共通

3.3.1. インタフェース概要

(1) 概要

コンテンツのタイトル、著者を指定して検索を行い、該当するコンテンツの DOI を返却するインタフェースである。

本 API にてリクエストパラメータで指定するタイトル、著者は、前方一致検索で対象コンテンツの有無を判定する。一方、3.2DOI 名_メタデータ検索 HTTP 共通（先述）でもリクエストパラメータにタイトル・著者を指定して検索できるが、「DOI 名_タイトル著者名検索 HTTP 共通」は前方一致検索で対象コンテンツの有無を判定する点が本 API とは異なる。タイトル、著者に関して曖昧性のある検索を行う場合には本 API を用いる。

(2) 送信先 URL

<https://japanlinkcenter.org/jalc/linkSakiAnnaiOto/doiTitleAuthorSearch/index>

(3) IF 仕様に関する特記事項

- プロトコルは、HTTP GET。
- 文字コードは、UTF-8 とし、日本語を URL パラメータに設定する際はエンコードを実施する。
- 1 リクエストのクエリ数は 1 件とする。
- 上限は HTTP GET の文字数制限 2083 文字。問合せ、回答ともに同じ。
- 結果が複数存在する場合、改行コード(¥n)で 1 件の区切りとする。

3.3.2. リクエスト

(1) リクエスト形式

<送信先 URL>?key=<key>&creator=<creator>&title=<title>

&contentsClassification=<contentsClassification>&multipleHits=<multipleHits>

(2) リクエストパラメータ

表 3-4 リクエストパラメーター一覧 を参照のこと。

表 3-4 リクエストパラメーター一覧 (DOI 名_タイトル著者名検索)

項番	パラメタ	必須	内容	備考
1	key	○	認証キーを記述する。	
2	creator	○	著者名を記述する。	
3	title	○	タイトルを記述する。	
4	contentsClassification		コンテンツ区分を設定する。	01:ジャーナルアーティクル, 02:書籍, 03:研究データ, 04:e-learning, 99:汎用データ
5	multipleHits	○	要求区分を記述する。	"01"(1件),"02"(複数件)

3.3.3. レスポンス

本インタフェースのレスポンスは、3.2DOI 名_メタデータ検索 HTTP 共通 のレスポンスと同じである。別表 2-1 レスポンス_メッセージボディ (XML) 【DOI 名_メタデータ検索】を参照のこと。

3.3.4. JaLC2 内での検索手順

● IP アドレス認証

リクエスト送信元の IP アドレスが JaLC に登録されているものと一致しない場合は、エラーメッセージ「EC0087: IP アドレスが一致しません」を返す。

● 項目チェック

リクエストパラメータの項目をチェックする。

- 認証キー（必須）が指定されていない場合は、エラーメッセージ「認証キーを入力してください」を返す。
- 認証キーが不正（認証に失敗）のときは、エラーメッセージ「EC0085: ID またはパスワードが正しくありません」を返す。
- 要求区分（必須）が指定されていない場合は、エラーメッセージ「EC0010: 要求区分を入力してください」を返す。また、要求区分の指定値が不適当な場合は、エラーメッセージ「EC0160: ○○は範囲外の値です」を返す。
- それ以外の項目は規定の文字数を超えた場合、エラーメッセージ「EC0040: ○○は XX 文字以内で入力してください」を返す。

● リクエストパラメータの正規化

パラメータの英数文字・スペースは、全角を半角に、小文字を大文字に変換（正規化）

する。(3.1 共通事項参照)

- 書誌メタデータによるコンテンツの特定

リクエストパラメータの書誌メタデータを元にタイトル検索テーブル、著者名検索テーブルを検索し、コンテンツの ID を取得する。

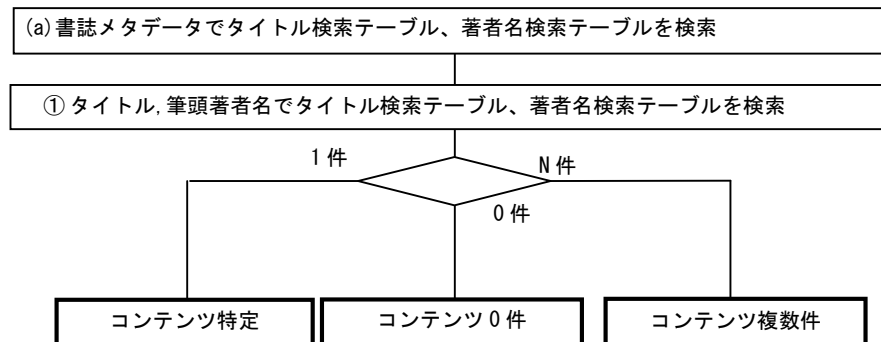


図 3-11 書誌メタデータによるコンテンツの特定

➤ 書誌メタデータ（タイトル、筆頭著者名）でタイトル検索テーブル、著者名検索テーブルを検索する。

➤ 0 件だった場合、コンテンツ情報存在チェックエラー(メッセージ ID : EI0100「条件に該当するデータがありません。」)とし、エラーメッセージを回答する。コンテンツの件数が複数件ある場合、リクエストパラメータより取得した要求区分を元に、回答判定（全て回答／エラー）を行う。

- レスpons作成

取得した情報から、レスポンスメッセージボディ(XML)を作成する。検索結果全件分、処理を繰り返す。

3.3.5. サンプル

- タイトル : Frictional Coefficient under Banana Skin、著書 : Kiyoshi Mabuchi
https://japanlinkcenter.org/jalc/linkSakiAnnaiOto/doiTitleAuthorSearch/index?key=*****&title=Frictional%20Coefficient%20under%20Banana%20Skin&creator=Kiyoshi%20Mabuchi&multipleHits=02

⇒ 1 件ヒット

- タイトル : Frictional Coefficient、著者 : Kiyoshi
https://japanlinkcenter.org/jalc/linkSakiAnnaiOto/doiTitleAuthorSearch/index?key=*****&title=Frictional%20Coefficient&creator=Kiyoshi&multipleHits=02

⇒ 1 件ヒット (タイトル、著者の前方一致検索でヒット)

3.4. DOI 名_研究者 ID 検索 HTTP 共通

3.4.1. インタフェース概要

(1) 概要

研究者 ID を元にコンテンツ検索を行い、該当するコンテンツの DOI を返却するインタフェースである。

(2) 送信先 URL

<https://japanlinkcenter.org/jalc/linkSakiAnnaiOto/doiResearcherIdSearch/index>

(3) IF 仕様に関する特記事項

- 文字コードは、UTF-8 とする。
- 使用する回答フォーマットは XML 形式とする。
- 1 リクエストで送信する研究者 ID は複数 1 件を許容する。
- プロトコルは、HTTP GET。
- 上限は HTTP GET の文字数制限 2083 文字。問合せ、回答ともに同じ。

3.4.2. リクエスト

(1) リクエスト形式

<送信先 URL>?key=<key>&researcherIdType=<researcherIdType>
&researcherIdCode=<researcherIdCode>
&contentsClassification=<contentsClassification>

(2) リクエストパラメータ

表 3-5 リクエストパラメーター一覧 を参照のこと。

表 3-5 リクエストパラメーター一覧 (DOI 名_研究者 ID 検索)

項番	パラメータ	必須	内容	備考
1	key	○	認証キーを記述する。	
2	researcherIdType	○	研究者 ID タイプを記述する。	記述例 (ORCID、READ、ERAD、JGLOBAL、NDL、KAKENHI)
3	researcherIdCode	○	研究者 ID コードを記述する。	
4	contentsClassification		コンテンツ区分を設定する。	01: ジャーナルアーティクル、02: 書籍、03: 研究データ、04: e-learning、99: 汎用データ

3.4.3. レスポンス

表 3-6 レスポンスメッセージボディ(XML) を参照のこと。

表 3-6 レスポンスメッセージボディ(XML)

項番	項目	属性	必須	内容	備考
1	root		○	ルートエレメント 配下にヘッダ要素・ボディ要素を保持するエレメント。	
2	head		○	配下に共通情報を保持するエレメント。	
3	errCode			エラーコードを設定する。	
4	body		※1	配下に元論文要素を複数(繰り返し)保持するエレメント。	
5	doi			DOIを設定する。	繰り返し可能。

※1:エラーコード未設定時のみ設定。

3.4.4. JaLC2 内での検索手順

● IP アドレス認証

リクエスト送信元の IP アドレスが JaLC に登録されているものと一致しない場合は、エラーメッセージ「EC0087 : IP アドレスが一致しません」を返す。

● 項目チェック

リクエストパラメータの項目をチェックする。

- 認証キー（必須）が指定されていない場合は、エラーメッセージ「認証キーを入力してください」を返す。
- 認証キーが不正（認証に失敗）のときは、エラーメッセージ ID : EC0087 「ID またはパスワードが正しくありません」を返す。
- 要求区分（必須）が指定されていない場合は、エラーメッセージ ID : EC0010 「要求区分を入力してください」を返す。また、要求区分の指定値が不適当な場合は、エラーメッセージ ID : EC0160 「〇〇は範囲外の値です」を返す。
- それ以外の項目は規定の文字数を超えた場合、エラーメッセージ ID : EC0040 「〇〇は XX 文字以内で入力してください」を返す。

● リクエストパラメータの正規化

パラメータの英数文字・スペースは、全角を半角に、小文字を大文字に変換（正規化）する。

● 研究者 ID タイプ、研究者 ID コードによる検索

研究者 ID タイプと研究者 ID コードの組合せに該当するコンテンツがあるか検索を行う。この検索では、通常公開・本公開の状態のコンテンツだけでなく、ステータスが早期公開のコンテンツも対象とする。検索の結果、該当するコンテンツが 0 件の場合、研究者 ID 該当データなしエラー(メッセージ ID : EI0340 「該当データがありません。」 またはエラーメッセージ ID : EI0100 「条件に該当するデータがありません。」)を返す。

- DOI の取得

上記検索でヒットしたコンテンツの DOI を取得する。

- レスpons作成

取得した情報から、レスポンスメッセージボディ(XML)を作成する。検索結果全件分、処理を繰り返す。

3.4.5. サンプル

- 研究者 ID タイプ: ORCID、研究者 ID コード: <http://orcid.org/0000-0001-9796-1163>、
コンテンツ区分: 01

https://japanlinkcenter.org/jalc/linkSakiAnnaiOto/doiResearcherIdSearch/index?key=*****&researcherIdType=ORCID&researcherIdCode=http://orcid.org/0000-0001-9796-1163&contentsClassification=01

⇒ 該当する DOI がヒット。

3.5. 書誌情報 XML_DOI 名検索 HTTP 共通

3.5.1. インタフェース概要

(1) 概要

DOI で問い合わせを行い、該当するコンテンツの書誌情報を返却するインタフェースである。

(2) 送信先 URL

<https://japanlinkcenter.org/jalc/linkSakiAnnaiOto/bookMetaDoiSearch/index>

(3) IF 仕様に関する特記事項

- 文字コードは、UTF-8 とする。日本語を URL パラメータに設定する際はエンコードを実施する。
- 使用する回答フォーマットは XML 形式とする。
- プロトコルは、HTTP GET。
- 上限は HTTP GET の文字数制限 2083 文字。問合せ、回答ともに同じ。

3.5.2. リクエスト

(1) リクエスト形式

<送信先 URL>?key=<key>&doi=<doi>

(2) リクエストパラメータ

表 3-7 リクエストパラメーター一覧 を参照のこと。

表 3-7 リクエストパラメーター一覧（書誌情報 XML_DOI 名検索）

項番	パラメータ	必須	内容	備考
1	key	○	認証キーを記述する。	
2	doi	○	DOI 名を記述する。	

3.5.3. レスポンス

別表 2-2 レスポンス【書誌情報 XML_DOI 名検索】 および 別表 2-3～2-7 レスポンス_メッセージボディ（XML）（各コンテンツ）【書誌情報 XML_DOI 名検索】を参照のこと。

3.5.4. JaLC2 内での検索手順

- IP アドレス認証

リクエスト送信元の IP アドレスが JaLC に登録されているものと一致しない場合は、エラーメッセージ「EC0087 : IP アドレスが一致しません」を返す。

- 項目チェック

リクエストパラメータの項目をチェックする。

- 認証キー（必須）が指定されていない場合は、エラーメッセージ ID:EC0010「認証キーを入力してください」を返す。
- 認証キーが不正（認証に失敗）のときは、エラーメッセージ ID : EC0087「ID またはパスワードが正しくありません」を返す。
- DOI（必須）が指定されていない場合は、エラーメッセージ ID:EC0010「認証キーを入力してください」を返す。
- 認証キー、DOI で指定された文字が規定の文字数を越えた場合、エラーメッセージ ID : EC0040「〇〇は XX 文字以内で入力してください」を返す。

- コンテンツの検索

パラメータで指定された DOI に対応するコンテンツを検索する。

検索結果が 0 件の場合は、エラーメッセージ ID : EI0100「条件に該当するデータがありません。」を返す。一方、検索結果が複数件の場合（同一 DOI のレコードが複数件ある＝マルチプルレゾリューション）は、マルチプル優先度の高い 1 件を回答するコンテンツとする。

- レスポンス作成

取得した情報から、レスポンスメッセージボディ(XML)を作成する。

3.5.5. サンプル

- DOI : 10.1241/johokanri.56.728

https://japanlinkcenter.org/jalc/linkSakiAnnaiOto/bookMetaDoiSearch/index?key=***&&doi=10.1241/johokanri.56.728

⇒ 該当する DOI の書誌 XML 情報を返却

4. 一般向けインタフェース（DOI コンテンツ検索）

本章では、DOI で検索・問合せを行い、該当するメタデータを取得するインタフェース仕様について説明する。本機能を「DOI コンテンツ検索」と呼ぶ。

※なお、本機能の仕様については、予告なしに変更することがある。

4.1. 概要（全体イメージ）

「DOI コンテンツ検索」とは、利用者が DOI のメタデータを取得しようとしたときに、DOI を指定してリクエストを投げると、システム側がリクエストで指定された形式に合わせてメタデータを返却する仕組みのことを言う。

レスポンス形式は、「RDF XML 形式」、「Citeproc JSON 形式」をサポートしており、HTTP リクエストヘッダで指定する。

DOI コンテンツ検索の流れを下記に示す。

図中の①→②→③は利用者が RDF XML 形式でのレスポンスを要求したときの流れであり、一方、図中の①'→②'→③'は Citeproc JSON 形式でのレスポンスを要求したときの流れである。

DOI コンテンツ検索のイメージを図 4-1 に示す。

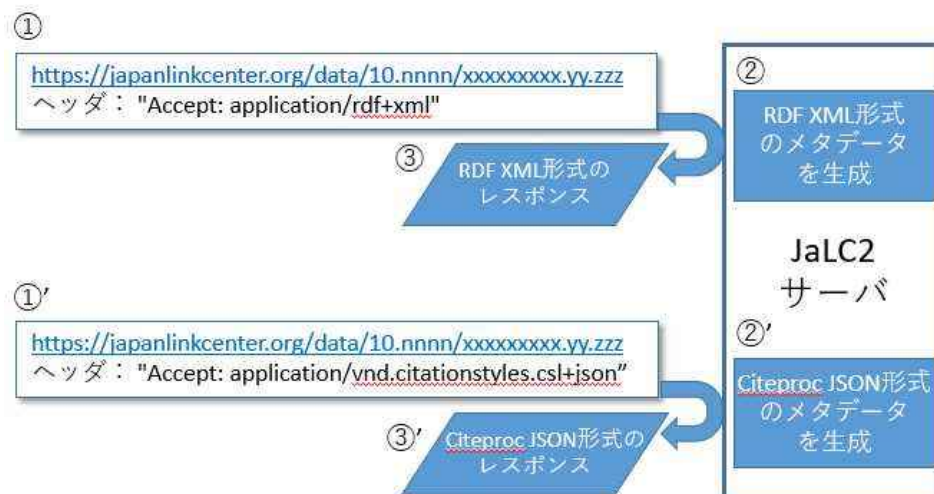


図 4-1 DOI コンテンツ検索のイメージ

DOI コンテンツ検索の流れを説明する。

まず、利用者が JaLC2 に対してリクエストを送信する（図中の①や①'）。リクエスト送信時には、メタデータを取得したい DOI を URL に記載するとともに、HTTP リクエストヘッダにレスポンス受信時に受信したい形式を指定する。図中では①は RDF XML 形式、①'は Citeproc JSON 形式を要求したリクエストである。

次に、外部からのリクエストを受信した JaLC2 サーバは、リクエストの中にある「DOI」と「形式」の情報を元に、DB から必要な情報を抽出し指定された形式のメタデータを生成する（図中の②や②'）。そして最後に、利用者へレスポンスを返却する（図中の③は RDF XML 形式、図中の③'は Citeproc JSON 形式）。

4.2. リクエスト

4.2.1. リクエスト形式

`https://japanlinkcenter.org/data/[DOI]`

HTTP リクエストヘッダ: "Accept:application/[形式]"

(1) DOI

メタデータを取得したい DOI を指定する。

(2) 形式

項番	形式	HTTP リクエストヘッダの内容
1	RDF XML 形式	Accept:application/xml
2		Accept:application/rdf+xml
3	Citeproc JSON 形式	Accept:application/json
4		Accept:application/vnd.citationstyles.csl+json

※2019 年 4 月から、項番 1 および、項番 3 の HTTP リクエストヘッダも使用可能

HTTP リクエストで形式の指定が無い場合は、Citeproc JSON 形式でレスポンスを返す。

4.2.2. curl コマンドを利用した実行例

＜RDF XML 形式＞

```
curl -H "Accept:application/xml" https://japanlinkcenter.org/data/10.nnnn/xxxxxxxxx.yy.zzzz
```

もしくは、

```
curl -H "Accept:application/rdf+xml" https://japanlinkcenter.org/data/10.nnnn/xxxxxxxxx.yy.zzzz
```

＜Citeproc JSON 形式＞

```
curl -H "Accept:application/json" https://japanlinkcenter.org/data/10.nnnn/xxxxxxxxx.yy.zzzz
```

もしくは、

```
curl -H "Accept: application/vnd.citationstyles.csl+json"
```

```
https://japanlinkcenter.org/data/10.nnnn/xxxxxxxxx.yy.zzzz
```

4.3. レスポンス

4.3.1. レスポンス形式

レスポンス形式は、「RDF XML 形式」、「Citeproc JSON 形式」の2種類。

(1) RDF XML 形式のレスポンス仕様

「別紙 4_DOI コンテンツ検索でのレスポンス(RDF XML 形式)」を参照のこと。

なお、RDF XML 形式は、現行使用しているプログラムの仕様により、多バイト文字はエスケープする仕様としている。

そのため、「4.3.3.RDF XML 形式のレスポンスサンプル」のようなレスポンスが返る。

(2) Citeproc JSON 形式のレスポンス仕様

「別紙 5_DOI コンテンツ検索でのレスポンス(Citeproc JSON 形式)」を参照のこと。

4.3.2. RDF XML 形式と Citeproc JSON 形式のレスポンス相違点

RDF XML 形式と Citeproc JSON 形式は、使用目的の違いから、以下のようなレスポンスの相違点がある。

- ・ Citeproc JSON 形式は、「DOI Citation Formatter※」と連携するために、本文言語(language)を参照して、一致するタイトル(title)、著者名(author)、出版者名(publisher)、出版者タイトル(container-title)等をレスポンスしている。(本文言語が不明の場合は“日本語”を優先。)

- ・ 一方、RDF XML 形式は、全ての言語をレスポンスしている。(例えば著者名が日本語と英語の両方がデータベースに登録されている場合、両方の情報をレスポンスする。)

※「DOI Citation Formatter」とは、DOI を画面に入力すると任意の引用形式で書誌情報を表示するサービスであり、DataCite, Crossref, mEDRA、ISTIC と JaLC が参加している。(2018 年 11 月時点)

4.3.3. RDF XML 形式のレスポンスサンプル

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:prism="http://prismstandard.org/namespaces/basic/2.0/">
  <rdf:Description rdf:about="https://doi.org/10.nnnn/xxxxxxx.yy.zzzz">
    <prism:doi>10.nnnn/xxxxxxx.yy.zzzz</prism:doi>
    <dcterms:title>2099 JST Summer Meeting : Making Research better</dcterms:title>
    <dcterms:title>2099&#x5e74;JST&#x590f;&#x5b63;&#x4f1a;&#x8b70;&#x3000;&#x7814;&#x7a76;&#x3092;&#x3088;&#x308a;&#x3088;&#x304f;&#x3059;&#x308b;&#x305f;&#x3081;&#x306b;</dcterms:title>
  </dcterms:title>
    <dcterms:creator>
      <foaf:Person>
        <foaf:name>JaLC Tarou</foaf:name>
        <foaf:familyName>JaLC</foaf:familyName>
        <foaf:givenName>Tarou</foaf:givenName>
      </foaf:Person>
    </dcterms:creator>
    <dcterms:creator>
      <foaf:Person>
        <foaf:name>&#xFF2A;&#xFF41;&#xFF2C;&#xFF23;&#x20;&#x592A;&#x90CE;</foaf:name>
        <foaf:familyName>&#xFF2A;&#xFF41;&#xFF2C;&#xFF23;&#xA;</foaf:familyName>
        <foaf:givenName>&#x592A;&#x90CE;</foaf:givenName>
      </foaf:Person>
    </dcterms:creator>
    <dc:creator>JaLC Tarou</dc:creator>
    <dc:creator>&#xFF2A;&#xFF41;&#xFF2C;&#xFF23;&#x20;&#x592A;&#x90CE;</dc:creator>
    <dcterms:publisher>&#x56fd;&#x7acb;&#x7814;&#x7a76;&#x958b;&#x767a;&#x6cd5;&#x4eba;&#x79d1;&#x5b66;&#x6280;&#x8853;&#x632f;&#x8208;&#x6a5f;&#x69cb;</dcterms:publisher>
    <dcterms:publisher>Japan Science and Technology Agency</dcterms:publisher>
    <dcterms:date rdf:resource="http://www.w3.org/2001/XMLSchema#date">2014</dcterms:date>
    <prism:volume>56</prism:volume>
    <prism:number>10</prism:number>
    <prism:startingPage>728</prism:startingPage>
    <prism:endingPage>729</prism:endingPage>
    <prism:issn>00219999</prism:issn>
    <prism:issn>13999999</prism:issn>
    <dcterms:publicationName>Journal of Information Processing and
    Management</dcterms:publicationName>
    <dcterms:publicationName>&#x60c5;&#x5831;&#x7b11;&#x7406;</dcterms:publicationName>
  </rdf:Description>
</rdf:RDF>

<?xml version="1.0" encoding="UTF-8"?>
```

※上記サンプルは、多バイト文字がエスケープされているレスポンス例で、架空の「DOI」を指定している。

```

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:prism="http://prismstandard.org/namespaces/basic/2.0/">
<rdf:Description rdf:about="http://doi.org/10.nnnn/xxxxxxx.yy.zzzz">
  <prism:doi>10.nnnn/xxxxxxx.yy.zzzz</prism:doi>
  <dcterms:title>2099 JST Summer Meeting : Making Research better</dcterms:title>
  <dcterms:title>2099 年 JST 夏季会議 研究をよりよくするために</dcterms:title>
  <dcterms:creator>
    <foaf:Person>
      <foaf:name>JaLC Tarou</foaf:name>
      <foaf:familyName>JaLC</foaf:familyName>
      <foaf:givenName>Tarou</foaf:givenName>
    </foaf:Person>
  </dcterms:creator>
  <dcterms:creator>
    <foaf:Person>
      <foaf:name>J a L C 太郎</foaf:name>
      <foaf:familyName>J a L C</foaf:familyName>
      <foaf:givenName>太郎</foaf:givenName>
    </foaf:Person>
  </dcterms:creator>
  <dc:creator>JaLC Tarou</dc:creator>
  <dc:creator>J a L C 太郎</dc:creator>
  <dcterms:publisher>国立研究開発法人 科学技術振興機構</dcterms:publisher>
  <dcterms:publisher>Japan Science and Technology Agency</dcterms:publisher>
  <dcterms:date rdf:resource="http://www.w3.org/2001/XMLSchema#date">2014</dcterms:date>
  <prism:volume>56</prism:volume>
  <prism:number>10</prism:number>
  <prism:startingPage>728</prism:startingPage>
  <prism:endingPage>729</prism:endingPage>
  <prism:issn>00219999</prism:issn>
  <prism:issn>13999999</prism:issn>
  <dcterms:publicationName>Journal of Information Processing and Management
  </dcterms:publicationName>
  <dcterms:publicationName>情報管理</dcterms:publicationName>
</rdf:Description>
</rdf:RDF>

```

※上記サンプルは、P45 のサンプルのレスポンス例の多バイト文字を復元したもので、架空の「DOI」を指定している。

4.3.4. Citeproc JSON 形式のレスポンスサンプル

```
{
  "DOI": "10.nnnn/johokanri.aa.bbb",
  "URL": "https://doi.org/10.nnnn/johokanri.aa.bbb",
  "language": "ja",
  "title": "2099 年 JST 夏季会議 研究をよりよくするために",
  "author":
  [
    {
      "given": "太郎",
      "family": "J a L C"
    },
    {
      "given": "次郎",
      "family": "J a L C"
    },
    {
      "given": "三郎",
      "family": "J a L C"
    },
    {
      "given": "四郎",
      "family": "J a L C"
    }
  ],
  "publisher": "国立研究開発法人 科学技術振興機構",
  "issued":
  {
    "date-parts":
    [
      [
        2014
      ]
    ]
  },
  "volume": "56",
  "number": "10",
  "page": "728-729",
  "ISSN":
  [
    "00217298",
    "13471597"
  ],
  "container-title": "情報管理"
}
```

※上記サンプルは、架空の「DOI」を指定している。

以上