Entitlement API Specification

for Integrators

Document Version
2.4 (May 2024)

Audience
Integrators, consumers of the getFTR Entitlement API

Contributors
getFTR Technical Steering Group (TSG)

Member	Publisher
Gareth Wright	John Wiley
Mike Petras	American Chemical Society (ACS)
Artem Artemyev	Springer Nature
David Greenwald	Elsevier
Paul Smith	Taylor and Francis
Donatas Bacius	Equal Experts
Chris Shillum	GetFTR

Document Revision History

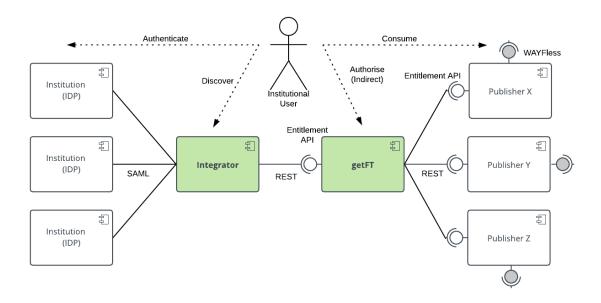| Version	Date	Comment	Author(s)
1.0	Oct 24 2019	First version	TSG
2.0	July 2020	Second version	TSG
2.1	July 2021	Introduced permFree	TSG
2.2	Feb 2023	Introduced Redirector Service	TSG
2.3	Jan 2024	Updated documentation for Hybrid API	TSG
2.4	May 2024	Introduced Retraction & Errata	TSG

Table of Contents

Overview

The Entitlement API establishes for a given document (DOI) and user affiliated institution the entitled level of access and appropriate content links, whether that be to the Version of Record or Alternate Version.

For brevity throughout the specification the following acronyms will be used:

Acronym	Definition
DOI	Digital Object Identifier
VoR	Version of Record
AV	Alternate Version
SAML	Security Assertion Markup Language
IdP	SAML Identity Provider
SP	SAML Service Provider

API version

The current API version is: v2.1

Endpoint

The URL for the API is: `https://entitlements.prod.getft.io`

Provides capability to establish entitlements for a single Organisation and a list of Documents (DOIs). The Organisation can be identified by a number of ID schemes.

Request

POST /v2.1/entitlements

```
{
  "org": {
    "ipv4": "...",
    "entityID": "...",
    "ringgoldID": "...",
  },
  "dois": [
    "...",
    "..."
  ]
}
```

Property	Required	Description
org	N	Describes a single Organisation and contains all the IDs available to identify it, with a minimum of at least one. Integrators are encouraged to share all known IDs.
ipv4	N	IPv4 address of the end user (not the Integrator).
ipv6	N	IPv6 address of the end user (not the Integrator).
entityID	N	The entityID of the Organisation's IdP.
openAthensOrgID	N	OpenAthens parameter. entityID must be present.
eduPersonScoped Affiliation	N	Shibboleth parameter. entityID must be present.
ringgoldID	N	The Organisation's Ringgold ID. For future proofing.

gridID	N	The Organisation's Grid ID. For future proofing.
rorID	N	The Organisation's Ror ID. For future proofing.
dois	Y	A list of 1 to 20 document identifiers (DOIs).

Response

For every document a corresponding Entitlement resource is returned, which establishes the level of entitlement; access type; document type (version of record or alternate); content type and ultimately a link to the actual resource if appropriate.

```
{
    "entitlements": [
        {
            "doi": "...",
            "statusCode": 200,
            "entitled": "yes",
            "accessType": "...",
            "org": {
                "ipv4": "..."
            },
            "vor": [
                {
                    "contentType": "...",
                    "url": "..."
                }
            ],
            "document": "..."
        }
    ]
}
```

Entitlements is a list of Entitlement resources, where the order of DOIs are preserved from the request.

For a complete set of scenarios refer to the "Entitlement API Scenarios" document.

Entitlement object

Property	Required	Description
doi	Y	The document identifier
statusCode	Y	Refer to appendix
entitled	Y	yes, no, maybe
accessType	N	open, free, permFree, paid
org	N	In the response, the "org" object contains the actual org ID(s) used to establish entitlement. If present, the org is known and was used to establish the level of entitlement.
vor	N	Version of Record object contains an array of Documents
av	N	Alternate Version object contains an array of Documents
document	Y	The URL for the document's landing page.
source	N	The origin of the entitlement within our service, this can be used for debugging purposes. Valid values are: unknown, publisher_not_supported, oa_platform, service_cache, centralised, service_request
updates	N	[Optional, if the feature is enabled for the integrator] Contains the updates to the current doi (if any). It is an array of Update Object

Document object

Property	Required	Description
contentType	Y	The declared MIME type of the document: ● application/epub+zip ● text/html ● application/pdf ● other
url	Y	A direct URL to the document. Will redirect to an IdP for authentication where appropriate (i.e. a "Smart Link").

Property	Required	Description
source	Y	The source of the update information. (e.g: retractionwatch, crossref)
updateDoi	Y	The DOI of the document, which holds the updated information about the original document.
updateDate	Y	Update date (e.g.: 2000-01-01)
updateType	Y	The type of the update (e.g: retraction, correction, expression-of-concern, reinstatement)
reasons	N	An array of strings that contain the detailed reasons for the update (if specified by the source)
urls	N	An array of external URLs regarding the update.

Entitlement truth table

The following truth table captures the legal combinations of VoR and AV in the Entitlement resource:

entitled	accessType	vor	av (alternate vs.)
yes	free|permFree|open|paid	true	false
maybe	paid	true	false
no	N/A	false	true|false

Source field

The source field inside the entitlement response object is primarily used for debugging and investigatory purposes. It indicates the source of the entitlement response on our platform.

- `unknown` - Represents an unknown source. Such that we were unable to retrieve the DOI (ie. it does not exist).

- `publisher_not_supported` - The DOI belongs to a publisher we don't currently support.

- `oa_platform` - The entitlement information was retrieved from the GetFTR OA & Free service.

- `service_cache` - The entitlement information was retrieved from the Cache.

- `centralised` - The entitlement was retrieved from the centralised entitlements API, where publishers have shared entitlements with GetFTR.

- `service_request` - The entitlement information was retrieved by contacting the publisher's API.

Links & Redirector service

By default, GetFTR collects clickthrough metrics for every link that was returned by the entitlement service. We are doing this to monitor effectiveness and measure the added value of the service.

This service was designed as a privacy-first solution, and we are only storing service level information about the clickthrough, like the integrator, the publisher of the DOI, the time of the click, when the click (or more specifically the redirect) happened. We do not store any data about the user, such as the IP Address, nor the DOI or link that was clicked on.
If you have concerns about this and would like to know more, please contact us.

GetFTR OA & Free service

When we receive a request for a DOI, we check to determine if it is an OA/Free DOI, using internal checks and logic. If a DOI satisfies one or more of these checks, we skip sending the request to the publisher API and respond on behalf of the publisher. For entitlements that have satisfied one of the following checks, the "source" key is set to `oa_platform`.

- Has a previous publisher response indicated this DOI is OA/Free/Permfree?
- Does the DOI have one or more licences that we know are OA?
- Is the publisher this DOI belongs to OA only?
- Does the DOI begin with a prefix that the owning publisher has designated as OA?
- Does the DOI belong to an OA Journal?
- Does the DOI belong to a journal registered with DOAJ?

Retraction & Updates

The GetFTR Entitlement API provides a summary of updates to the Document for every DOI where information is available at Crossref or Retraction Watch.

GetFTR provides an additional service, Document Status Page, which shows the history of changes to the document in timeline format. To access this page for a specific document add

the DOI after the following prefix "https://document-status.getft.io/doi/". For example, the link to the Document Status Page for "10.3389/fenvs.2022.983136" DOI is the following:

https://document-status.getft.io/doi/10.3389/fenvs.2022.983136

Security

Transport (TLS)

All communications are encrypted over a TLS 1.2, or above, connection. The TLS handshake will exchange server certificates only.

Integrator ID

getFTR issues each Integrator with a unique "Integrator ID" String. Integrators present this identifier in every request as the HTTP header "X-INTEGRATOR-ID". getFTR will use this to lookup the "shared secret" (below).

Authentication (JWT)

Integrators must sign all API requests with a JWT bearer token (rfc7519), which getFTR is responsible for verifying.

getFTR issues each Integrator with a unique shared secret, a pseudo random generated 256 bit long number encoded in Base64. This must be decoded into a raw byte array when signing and verifying requests.

The following JWT properties, exhaustive list, have been adopted:

Property	Container	Value	Usage Comments
alg	Header	HS256	
typ	Header	JWT	
iss	Payload	*	The Integrator's ID in lowercase.
aud	Payload	getftr	
iat	Payload	*	Unix time used to expire stale requests (10 mins)
jti	Payload	*	Standard usage. Nonce used to avoid replay attacks.
doi	Payload	*	DOI of the **first item** in the batch in lowercase.

`doi` is the <u>only</u> business property extension to the JWT payload, deemed enough to confirm the authenticity of the Integrator and integrity of the request.

Integrators are responsible for generating valid Bearer tokens. Refer to [Status Codes](#).

Example

Shared Secret

ggQ+DfA3DJrh+qCsfFfsv7lH0JaSAFoOHdynYNkeKgXASKyj3HSNNifCYv7nftAKGBbrT8t
CfZknoqPbRxcha+JFnTRWtF4CtapW47K/WRg6BzMvtL7NL3/6okbbLdmuYvbPYuZza/HcBJ
nqRy1QB4kz5D+fEwPZPaEYu3B1DIoNw9rbd+buE52wAD21nQ5Rmk4Y66yNYnOUkU8GHwxZ0
3O3ghDFPvsr9Tl8dhC8I14ahfPPUe23EUkCBiYjys212lvfB2m/O7GqWQSpTx9hZfQvN+a/
d+cagTRtduDhLq13wBD98V0w/jXgwtB5GvEw6MrYUgGcxo+qT3355pyUNA==

JWT Token Header

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

JWT Token Payload

```
{
  "iss": "mendeley",
  "aud": "getftr",
  "iat": 1568110518,
  "jti": "aa2d6150-c89a-46af-9dd1-8ed04be80b0f",
  "doi": "12.345/2018zz112233"
}
```

JWT Auth Header

Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJtZW5kZWxleSIsImF1ZCI6Im
dldGZ0ciIsImlhdCI6MTU2ODExMDUxOCwianRpIjoiYWEyZDYxNTAtYzg5YS00NmFmLTlkZ
DEtOGVkMDRiZTgwYjBmIiwiZG9pIjoiMTIuMzQ1LzIwMTh6ejExMjIzMyJ9.7j1C0-3pzXD
RX3MlLCrV38EVA1P-pb7YOMPLm4VIBaw

Resources

- https://jwt.io/#debugger-io
- https://mkjwk.org/

- https://www.unixtimeconverter.io/

Quality of Service (QoS)

Each request must include a X-API-KEY HTTP header, supplied by getFTR . This is to ensure all Integrators get the same Quality of Service, and no one Integrator overloads getFTR. Integrators should gracefully handle a 429 Response Code (Too Many Requests)

Tracing

Each request must include a unique `X-REQUEST-ID` HTTP header which will be carried forward:

`X-REQUEST-ID: 3e5980ba-ceae-4976-a9d4-c7e6ac49a20b (GUID)`

Versioning

Major Versions

Major version changes result in a breaking change to the interface contract. The major version number (X below) is declared in the path as follows:

`/vX/entitlements`

The API is at version 2.1 and is reflected in the path as follows:

`/v2.1/entitlements`

Minor Versions

Minor version changes result in a non-breaking change (e.g. additional SAML attribute support). The minor version is not reflected in the path.

Robustness

getFTR will conform completely to the specification, but Integrators must be able to accept input with any non-breaking changes. In other words:

"Be liberal in what you accept, be conservative in what you send"

(Postel's Law, aka The Robustness Principle)

Status Codes

Batch Level (HTTP)

The following status codes are returned via the standard HTTP Status Code Header:

Code	Definition	Scenario
200	OK	The request has succeeded
400	Bad request	- More than 20 DOIs - Payload is not valid
401	Unauthorized	- Integrator is not known - Integrator is not authorized - Request is a replay - Request is over 10 mins old
403	Forbidden	Integrator is blocked
404	Not Found	The endpoint is not known
405	Method Not Allowed	HTTP method is not known
429	Too Many Requests	Request quota exceeded
500	Internal Server Error	The Entitlement API has thrown an error
504	Gateway Timeout	Upstream services timed out

Item Level (JSON)

The following status codes are returned via the JSON response body:

Code	Definition	Scenario
200	OK	Entitlement determined
403	Forbidden	Access to content is restricted
404	Not Found	Unknown DOI
500	Internal Server Error	Error handling entitlement result
502	Bad Gateway	Publisher API rate limit reached
503	Service Unavailable	Entitlement cannot be checked

504	Gateway Timeout	Entitlement check timed out

Appendix

Encoding & Formatting

The following rules apply to Entitlement API request and response structures:

Single line JSON

The JSON body response must be single line, with no line feeds or carriage returns.

Whitespace

No white space between properties and values in the JSON response.

Character Encoding

UTF-8 is the adopted character encoding standard.

Adopted Standards

The Entitlement API specification adopts a number of open standards and patterns, outlined below:

Standard	Version	Definition
REST	N/A	Representational state transfer pattern.
JWT	RFC 7519	Open standard auth token.
HMAC	SHA256	Hash algorithm used to digitally sign messages.
HTTPS	N/A	Secure HTTP communications using latest TLS standard. See versions below.
HTTP	2.0	Hypertext Transfer Protocol.
TLS	1.3	Transport Layer Security.
JSON Schema	7	JSON Schema.
URI	RFC 3986	Uniform Resource Identifier.

Unix Time	N/A	Unix Epoch Time.

HTTP Headers

All adopted HTTP headers are listed here:

Header	Use
X-REQUEST-ID	Used for tracing requests.
X-INTEGRATOR-ID	Can be used for logging.
AUTHORIZATION: Bearer	Used for authentication.
X-API-KEY	Used for QoS.