

Entitlement API Specification

Integrators

Version

2.2 (Feb 2023)

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

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

Table of Contents

[Entitlement API Specification](#)

[Version](#)

[Audience](#)

[Contributors](#)

[Revision History](#)

[Table of Contents](#)

[Overview](#)

[Endpoint](#)

[Request](#)

[Response](#)

[Entitlement object](#)

[Document object](#)

[Entitlement truth table](#)

[Security](#)

[Transport \(TLS\)](#)

[Integrator ID](#)

[Authentication \(JWT\)](#)

[Example](#)

[Shared Secret](#)

[JWT Token Header](#)

[JWT Token Payload](#)

[JWT Auth Header](#)

[Resources](#)

[Quality of Service \(QoS\)](#)

[Tracing](#)

[Versioning](#)

[Major Versions](#)

[Minor Versions](#)

[Robustness](#)

[Status Codes](#)

[Batch Level \(HTTP\)](#)

[Item Level \(JSON\)](#)

[Appendix](#)

[Encoding & Formatting](#)

[Single line JSON](#)

[Whitespace](#)

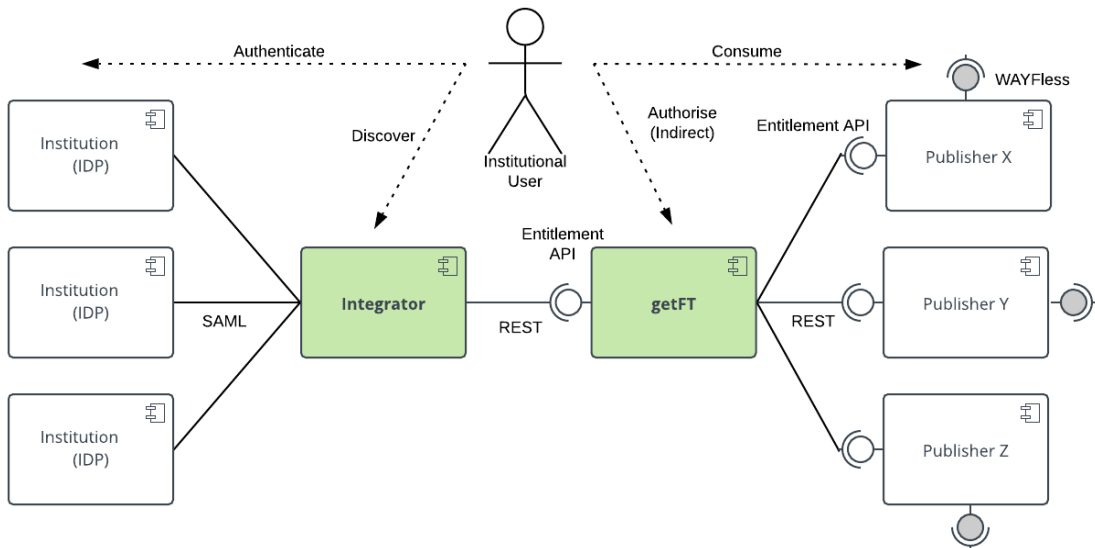
[Character Encoding](#)

[Adopted Standards](#)

[HTTP Headers](#)

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

Endpoint

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/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 “Entitlement API - Scenarios - vs 2.0” 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 <u>actual</u> org ID(s) used to establish entitlement. If present, the org is known <u>and</u> 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.

Document object

Property	Required	Description
contentType	Y	The declared MIME type of the document: <ul style="list-style-type: none">• 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”).

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

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 our effectiveness and measure the added value of our 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.

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 only 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+qCsffsv7lH0JaSAFo0HdynYNkeKgXASKyj3HSNNifCYv7nftAKGBbrT8t  
CfZknoqPbRxcha+JFnTRWtF4CtapW47K/WRg6BzMvtL7NL3/6okbbLdmuYvbPYuZza/HcBJ  
nqRy1QB4kz5D+fEwPZPaEYU3B1DIoNw9rbd+buE52wAD21nQ5Rmk4Y66yNYnOUkU8GHwxZ0  
303ghDFPvsr9Tl8dhC8I14ahfPPUe23EUkCBiYjys2121vfB2m/07GqWQSpTx9hZfQvN+a/  
d+cagTRtduDhLq13wBD98V0w/jXgwtB5GvEw6MrYUGcxo+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.eyJpc3MiOiJtZW5kZWxleSI6ImF1dGZ0ciIsImhdCI6MTU2ODExMDUxOCwianRpIjoieWVlZDYxNTAtYzgzNS00NmFmLTlkZDEtOGVkaMDRiZTgwYjBmIiwiaXNjaWkiOiJjoiMTIuMzQ1LzIwMTgzZz11MjIzMyJ9.7j1C0-3pzXDRX3M1LCrV38EVA1P-pb7Y0MPLm4VIBaw
```

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

This specification is at version 2.0 and is reflected in the path as follows:

/v2/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	<ul style="list-style-type: none">• More than 20 DOIs• Payload is not valid
401	Unauthorized	<ul style="list-style-type: none">• 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 .