Entitlement API Specification

for Publishers

Document Version

2.1 (July 2021)

Audience

Publishers, providers of an 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

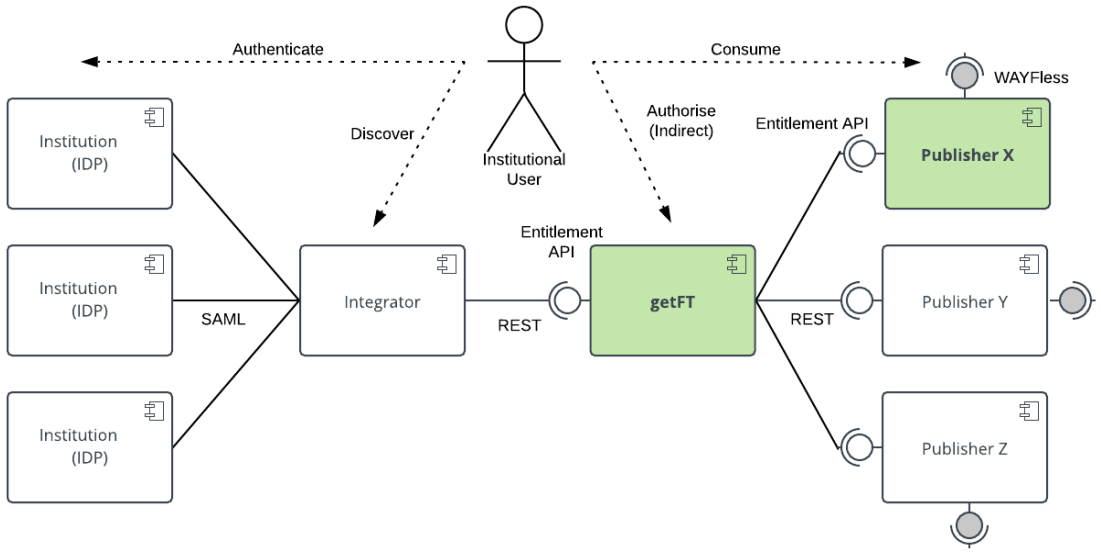
Document Revision History

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

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

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” 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.

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”).

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

Security

Transport (TLS)

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

Authentication (JWT)

getFTR signs all API requests with a JWT bearer token (rfc7519), which Publishers are responsible for verifying.

Publishers must issue getFTR 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	getftr	
sub	Payload	*	The Integrator's name in lowercase.
aud	Payload	*	The Publisher's name in lowercase.
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.

Publisher's are responsible for validating the Bearer token. Refer to Status Codes.

Example

Shared Secret

```
VGkAdj00elk1sCY9X5sMV61/Vo3S2S5HvZxbcsRjSH6mMSbedPCQ630LmQjflPhLtBKxUx1
vaHQlq5JZUu297/8yNwdZbhqbKSVw8SYkaRss5zBOs2aGaqQejvwIbBmGFLusQrt7yycPws
/N4P1PYIpan21ROiFtzPvkiwE2ZpICfgfiHcJ+J9yOqnSgRQ8SJhBlJZ7IHnoqD+svi8jer
qdrmeafdJW1roCKgCcrOpi7lwrl6KeEJ+IASMoVw0ODo3G9KeHmwed/Y821IAAnJFkOBzYz
7269fH716mkoM16GxGfY7HTa9JcKHTat3vwWub0G+wY4jJbKHCRNr/dd9A==
```

JWT Token Header

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

JWT Token Payload

```
{
  "iss": "getftr",
  "sub": "mendeley",
  "aud": "wiley",
  "iat": 1568110518,
  "jti": "83d4f63a-6486-4653-ac0a-1bf3c82183af",
  "doi": "12.345/2018zz112233"
}
```

JWT Auth Header

```
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnZXRmdHIiLCJzdWIiOiJtZW
5kZWxleSIsImF1ZCI6IndpbGV5IiwiaWF0IjoxNTY4MTEwNTE4LCJqdGkiOiI4M2Q0ZjYzY
S02NDg2LTQ2NTMtYWMwYS0xYmYzYzgyMTgzYWYiLCJkb2kiOiIxMi4zNDUvMjAxOHp6MTEy
MjMzIn0.MbcX8rToNlaeLmGBjnsE4qlHMtj4B_GV3kzpetgXvSg
```

Resources

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

Tracing

Each request will include a unique `X-REQUEST-ID` HTTP header which can be observed:

```
X-REQUEST-ID:
02690813-9d09-4b76-a068-e064c8ce1a1e:3e5980ba-ceae-4976-a9d4-c7e6ac49a20b
```

```
X-REQUEST-ID = INTEGRATOR-REQUEST-ID:GETFTR-REQUEST-ID
```

This request id encoding provides a simple tracing strategy.

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

Publishers will conform completely to the specification, but must be able to accept input with any non-breaking changes (e.g. new query params, introduced in a more recent minor version release, should be gracefully ignored) . 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	• Request is not authorized • Request is a replay • Request is over 10 mins old
403	Forbidden	• Client is blocked • 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
404	Not Found	Unknown DOI
504	Gateway Timeout	Entitlement check timed out

Value a partial response over a complete timeout: some entitlements can still be cached.

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.