

Entitlement API Specification

Publishers

Version

1.0 (Oct 2019)

Audience

Publishers, providers of an Entitlement API.

Contributors

Technical Steering Group (TSG)

Member	Publisher
Gareth Wright	John Wiley
Mike Petras	American Chemical Society (ACS)
Josep Prat	Springer Nature
Terence Clifton	Elsevier
Matt Kemp	Taylor and Francis
Paul Smith	Taylor and Francis
Kevin Hinde	Elsevier
Artem Artemyev	Springer Nature

Document Revision History

Version	Date	Comment	Author(s)
1.0	Oct 24 2019	First version	TSG

Table of Contents

[Entitlement API Specification](#)

[Version](#)

[Audience](#)

[Contributors](#)

[Document Revision History](#)

[Overview](#)

[Entitlement Endpoint](#)

[Request](#)

[Response](#)

[Entitlement object](#)

[Document object](#)

[Entitlement truth table](#)

[Scenarios](#)

[#1 Version of Record \(VoR\)](#)

[Request](#)

[Response](#)

[#2 Best Available Version \(BAV\)](#)

[Request](#)

[Response](#)

[#3 No Best Available Version](#)

[Request](#)

[Response](#)

[#4 Maybe entitled](#)

[Request](#)

[Response](#)

[#5 Unknown Institution + Open/Free](#)

[Request](#)

[Response](#)

[Security](#)

[Transport \(TLS\)](#)

[Authentication \(JWT\)](#)

[Example](#)

[Shared Secret](#)

[JWT Token Header](#)

[JWT Token Payload](#)

[JWT Auth Header](#)

[Resources](#)

[Tracing](#)

[Caching](#)

[Caching enabled](#)

[Caching disabled](#)

[Versioning](#)

[Major Versions](#)

[Minor Versions](#)

[Robustness](#)

[Response Codes](#)

[Service Health](#)

[Build](#)

[Appendix](#)

[Encoding & Formatting](#)

[URL Encoding](#)

[Single line JSON](#)

[White space](#)

[Character Encoding](#)

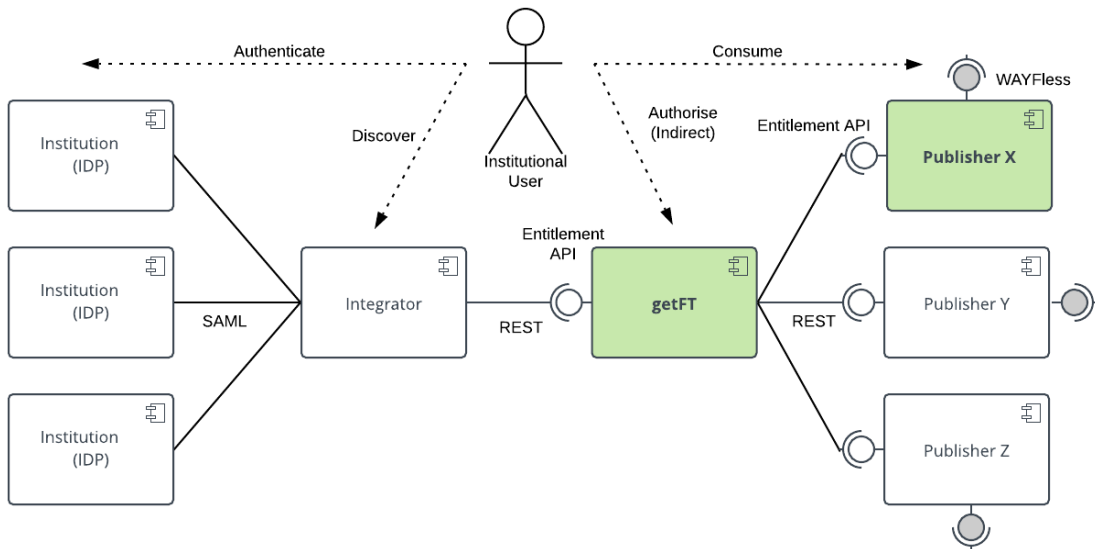
[Readability](#)

[Adopted Standards](#)

[Schema](#)

Overview

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



Entitlement Endpoint

The Entitlement resource is the only resource declared on the Entitlement API.

Request

GET

/v1/entitlement?doi={doi}&entityID={entityID}

The full list of supported request parameters are described here:

Parameter	Required	Description
doi	Y	The DOI of the target document.
entityID	N	The entityID of the IdP which authenticated the user.
orgID	N	OpenAthens parameter from SAML auth response.
eduPersonScopedAffiliation	N	Shibboleth parameter from SAML auth response.
prettyPrint	N	Format the response to be human readable, with line breaks and spacing where appropriate (as shown in all examples).

Response

The Entitlement resource is returned in JSON form, refer to [schema](#).

Entitlement object

Property	Required	Description
entitled	Y	yes, no, maybe
doi	Y	The DOI of the document.
entityID	N	The entityID of the IdP which authenticated the user.
accessType	N	open, free, paid
vor	N	Version of Record object contains an array of Documents
bav	N	Best Available Version object contains an array of Documents
document	Y	The URL for the document's landing page.

Document object

The document object is returned in JSON form, refer to [schema](#).

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	The url to the resource

Entitlement truth table

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

entitled	accessType	vor	bav	document
yes	free open paid	true	false	true
maybe	paid	true	false	true
no	N/A	false	true false	true

Scenarios

#1 Version of Record (VoR)

Institution is entitled to the document. DOI and Institution are known and combined have entitlement rights. Response will contain a VoR (i.e. "vor" field). Example:

Request

```
/v1/entitlement?doi=12.345/2018zz112233&entityID=https://example.idp.org&prettyPrint=true
```

Response

```
{
  "entitled": "yes",
  "doi": "12.345/2018zz112233",
  "entityID": "https://example.idp.org",
  "accessType": "open",
  "vor": [
    {
      "contentType": "application/pdf",
      "url":
        "https://example.publisher.com/doi/pdf/12.345/2018zz112233?entityID=https://example.idp.org"
    },
    {
      "contentType": "application/epub+zip",
      "url":
        "https://example.publisher.com/doi/epub/12.345/2018zz112233?entityID=https://idp.example.org"
    },
    {
      "contentType": "text/html",
      "url":
        "https://example.publisher.com/doi/full/12.345/2018zz112233?entityID=https://idp.example.org"
    }
  ],
  "document":
    "https://example.publisher.com/doi/abs/12.345/2018zz112233?entityID=https://idp.example.org"
}
```


#2 Best Available Version (BAV)

Institution is not entitled to the document. DOI and Institution are known and combined do not have entitlement rights. Response will be missing a VoR but a BAV is available (i.e. "bav" field). Example:

Request

```
/v1/entitlement?doi=12.345/2018zz445566&entityID=https://example.idp.org&prettyPrint=true
```

Response

```
{
  "entitled": "no",
  "doi": "12.345/2018zz445566",
  "entityID": "https://idp.example.org",
  "bav": [
    {
      "contentType": "application/pdf",
      "url":
"https://example.publisher.com/doi/pdf/12.345/2018zz445566"
    }
  ],
  "document":
"https://example.publisher.com/doi/abs/12.345/2018zz445566"
}
```

#3 No Best Available Version

Institution is not entitled to the document. DOI and Institution are known and combined do not have entitlement rights. Furthermore a BAV is not available. Example:

Request

```
/v1/entitlement?doi=12.345/2019zz778899&entityID=https://idp.example.org&prettyPrint=true
```

Response

```
{  
  "entitled": "no",  
  "doi": "12.345/2019zz778899",  
  "entityID": "https://idp.example.org",  
  "document":  
  "https://example.publisher.com/doi/abs/12.345/2019zz778899"  
}
```

#4 Maybe entitled

Institution/Department cannot be uniquely identified (additional SAML attributes required). The DOI is known. The scenario can arise in deferred authentication where only the entityID is supplied. A VoR is returned with a “maybe” entitled status. Example:

Request

```
/v1/entitlement?doi=12.345/2018zz112233&entityID=https://example.idp.org&prettyPrint=true
```

Response

```
{
  "entitled": "maybe",
  "doi": "12.345/2018zz112233",
  "entityID": "https://example.idp.org",
  "accessType": "paid",
  "vor": [
    {
      "contentType": "application/pdf",
      "url":
        "https://example.publisher.com/doi/pdf/12.345/2018zz112233?entityID=https://example.idp.org"
    }
  ],
  "document":
    "https://example.publisher.com/doi/abs/12.345/2018zz112233?entityID=https://idp.example.org"
}
```

#5 Unknown Institution + Open/Free

DOI is known but the Institution is unknown (EntityID is missing or unknown). If the document is free or open-access a VoR is returned. Example:

Request

```
/v1/entitlement?doi=12.345/2018zz998877&prettyPrint=true
```

Response

```
{
  "entitled": "yes",
  "doi": "12.345/2018zz998877",
  "accessType": "open",
  "vor": [
    {
      "contentType": "application/pdf",
      "url":
"https://example.publisher.com/doi/pdf/12.345/2018zz998877"
    }
  ],
  "document":
"https://example.publisher.com/doi/abs/12.345/2018zz998877"
}
```

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	getft	
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	*	The document's DOI in lowercase.
idp	Payload	*	The IDP's EntityID in lowercase. null if omitted.

doi and idp are the only business property extensions 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 [Response Codes](#).

Tracing

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

```
X-REQUEST-ID:  
02690813-9d09-4b76-a068-e064c8ce1a1e:3e5980ba-ceae-4976-a9d4-c7e6ac49a2  
0b
```

```
X-REQUEST-ID = INTEGRATOR-REQUEST-ID:GET-FT-REQUEST-ID
```

This request id encoding provides a simple tracing strategy.

Caching

The Publisher's Entitlement API must declare cache control behaviour using standard HTTP Header: "cache-control". Legal directive options:

Caching enabled

To direct getFTR to cache responses, implement the following directives:

```
cache-control: private  
cache-control: max-age
```

Example (shorthand):

```
cache-control: private, 1800
```

Caching disabled

To direct getFTR not to cache responses, implement the following directive:

```
cache-control: no-store
```


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/entitlement?doi={doi}
```

This specification is at version 1.0 and is reflected in the path as follows:

```
/v1/entitlement?doi={doi}
```

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](#))

Response Codes

Response Code	Definition	Scenario
200	OK	All legitimate scenarios .
400	Bad request	Generic catch all, if can't map to a specific error defined here.
401	Unauthorized	Authorization header is missing; Authorization header is invalid/wrongly computed; The request is a replay; The request is stale (over 10 mins old)
403	Forbidden	The Entitlement API can restrict access by IP and return a 403 if illegally accessed
404	Not Found	The DOI is not found on publisher side. The endpoint is unknown (in the unlikely situation of manually calling API).
405	Method Not Allowed	Attempting to use another method other than GET on the entitlement endpoint
429	Too Many Requests	Too many requests to the Entitlement API. Have exceeded quota limit for the endpoint
500	Internal Server Error	The Entitlement API has thrown an error, which must be logged internally appropriately.

Service Health

`/v1/entitlement/status`

This endpoint must return a HTTP response code 200 if the service is up and healthy and a 503 if unhealthy/unavailable. The endpoint is not behind Authentication and Rate Limited.

Build

Publishers should return an X-BUILD-NUMBER header in the response, to uniquely identify the implemented build.

Appendix

Encoding & Formatting

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

URL Encoding

All API URLs are constructed in compliance with [RFC 3986](#).

Single line JSON

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

White space

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

Character Encoding

UTF-8 is the adopted character encoding standard.

Readability

To improve readability, in all worked examples, all responses are shown with prettyPrint enabled.

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.

Schema

Adopted JSON schema format: <http://json-schema.org>

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id":
"http://getfulltextresearch.com/schema/entitlement-schema-1-0.json",
  "type": "object",
  "title": "Entitlement API Schema 1.0",
  "oneOf": [
    {
      "$ref": "#/entitledResponse"
    },
    {
      "$ref": "#/bavResponse"
    },
    {
      "$ref": "#/unentitledResponse"
    }
  ],
  "definitions": {
    "doi": {
      "$id": "#/definitions/doi",
      "type": "string"
    },
    "entitled": {
      "$id": "#/definitions/entitled",
      "type": "string",
      "enum": [
        "yes",
        "no",
        "maybe"
      ]
    },
    "accessType": {
      "$id": "#/definitions/accessType",
      "type": "string",
      "enum": [
        "open",
        "free",
        "paid"
      ]
    },
    "contentType": {
```

```

"$id": "#/definitions/contentType",
"type": "string",
"enum": [
  "application/pdf",
  "text/html",
  "application/epub+zip",
  "other"
]
},
"url": {
  "$id": "#/definitions/url",
  "type": "string",
  "format": "uri",
  "pattern": "^(https?|http?|ftp?|ftp?)://"
},
"entityID": {
  "$id": "#/definitions/entityID",
  "$ref": "#/definitions/url"
},
"document": {
  "$id": "#/definitions/document",
  "type": "object",
  "properties": {
    "contentType": {
      "$ref": "#/definitions/contentType"
    },
    "url": {
      "$ref": "#/definitions/url"
    }
  },
  "additionalProperties": false,
  "required": [
    "contentType",
    "url"
  ]
},
"documentArray": {
  "$id": "#/definitions/documentArray",
  "type": "array",
  "items": {
    "$ref": "#/definitions/document"
  }
},
"entitledResponse": {
  "$id": "#/entitledResponse",
  "properties": {

```

```

    "entitled": {
      "enum": [
        "yes",
        "maybe"
      ]
    },
    "doi": {
      "$ref": "#/definitions/doi"
    },
    "entityID": {
      "$ref": "#/definitions/entityID"
    },
    "accessType": {
      "$ref": "#/definitions/accessType"
    },
    "vor": {
      "$ref": "#/definitions/documentArray"
    },
    "document": {
      "$ref": "#/definitions/url"
    }
  },
  "additionalProperties": false,
  "required": [
    "entitled",
    "doi",
    "accessType",
    "vor",
    "document"
  ]
},
"bavResponse": {
  "$id": "#/bavResponse",
  "properties": {
    "entitled": {
      "const": "no"
    },
    "doi": {
      "$ref": "#/definitions/doi"
    },
    "entityID": {
      "$ref": "#/definitions/entityID"
    },
    "bav": {
      "$ref": "#/definitions/documentArray"
    }
  },

```



```

        "document": {
          "$ref": "#/definitions/url"
        }
      },
      "additionalProperties": false,
      "required": [
        "entitled",
        "doi",
        "bav",
        "document"
      ]
    },
    "unentitledResponse": {
      "$id": "#/unentitledResponse",
      "properties": {
        "entitled": {
          "const": "no"
        },
        "doi": {
          "$ref": "#/definitions/doi"
        },
        "entityID": {
          "$ref": "#/definitions/entityID"
        },
        "document": {
          "$ref": "#/definitions/url"
        }
      },
      "additionalProperties": false,
      "required": [
        "entitled",
        "doi",
        "document"
      ]
    }
  }
}

```

<https://jsonschemalint.com>