

LearningVault.io Public API

Last Revised: 15 October 2022

Overview	1
Authentication	2
API Token.....	2
Environments.....	2
Swagger Documentation	2
Linking to hosted pages	3
Badge Assertion Page.....	3
API Reference.....	3
Badge Templates.....	3
Get Badge Templates.....	3
Get Badge Template By ID	7
Issued Badges.....	9
Issuing a badge	9
.....	9
Get The ID of the issued Badge	11
Getting a user's backpack.....	13
Revoking a Badge.....	15
Expire a Badge	16
Getting a Badge Item / Backpack Item / Issued Badge	17
Field Definitions	20

Overview

The Learning Vault badging website is where you can create and manage badges on the Learning Vault badging system: <https://www.learningvault.io>

Below is a list of common endpoints you can use to create and query badges. Each documented endpoint below will indicate whether Authentication I needed (see Authentication section below).

Authentication

API Token

Authentication requires an API Key and Token. These can be generated at www.learningvault.io under Issuing Badges > API. To generate a new token, click the green key icon next to the API key

To make calls against the Learning Vault API's, simply add the following headers to your request:

- **ApiKey:** API_KEY
- **Authorization:** "Bearer" + API_TOKEN

```
POST ../api/ActivityEvents
Content-Type: application/json
ApiKey: API_KEY
Authorization: Bearer API_TOKEN
```

Environments

There are 2 environments:

- Production
 - **Dashboard:** <https://www.learningvault.com.au/>
 - **Events:** <https://events.learningvault.io/>
 - **Badges:** <https://badges.learningvault.io>
- Sandbox
 - **Dashboard:** <http://sandbox.learningvault.com.au/>
 - **Events:** <https://events.sandbox.learningvault.com.au/>
 - **Badges:** <https://api.sandbox.learningvault.com.au/>

The API's below will reference either the Events or Badges endpoints. They will be shown as {{events}} and {{badges}}

Swagger Documentation

The endpoints have a minimal swagger integration to help with testing. You can find them at:

- <http://{{Events}}/swagger/index.html?url=/swagger/v1/swagger.json>
- <http://{{Badges}}/swagger/index.html?url=/swagger/v1/swagger.json>

Linking to hosted pages

This section documents specific links to our platform, and how they can be accessed.

Badge Assertion Page

The badge assertion page is the public page for a badge. This page can be linked to be shared on social media.

The format of the URL is:

`{{dashboard}}/view/#!/badge?badgeid={{BADGE_ID}}&includeSocials=true`

- **{{BADGE_ID}}** is the GUID ID of the issued badge to be loaded
- **includeSocials** is optional and can be true/false. If set to true, the social sharing buttons will be included on the page. Defaults to false (no social sharing buttons)

API Reference

Badge Templates

Get Badge Templates

Retrieves all badge templates and their details.

<i>Property</i>	<i>Description</i>
<i>URL</i>	<code>{{Badges}}/api/issuedbadgesexternal/badgeTemplate</code>
<i>Methods</i>	GET
<i>Authentication</i>	Required

Optional Querystring Fields:

<i>Field</i>	<i>Description</i>
<i>searchQuery</i> (string)	The search query will filter the results with ones that have a name that's contains the value or a tag that contains the value. <i>Note: The value should be URL encoded</i>
<i>skip</i> (integer)	The skip parameter will skip a given number or records from the results <i>Note: The Take parameter is required when using skip.</i>
<i>Take</i> (integer)	The take parameter limits the number of results by the given value. Skip / Take can be used for paging results.
<i>summaryOnly</i> (boolean)	Use summaryOnly=true in GET request to retrieve a smaller version of the response object.

Example Request:

```
GET {{Badges}}/api/issuedbadgesexternal/badgeTemplate?skip=10&take=10
Content-Type: application/json
ApiKey: API_KEY
Authorization: Bearer API_TOKEN
```

Example Responses:

Success (200): summaryOnly=true

```
[{
  "id": "OB-7cb2c7ed-4e1a-4e42-856a-bf35c3830cfd",
  "name": "Dev test",
  "description": "getting it up and running",
  "expires": null,
  "language": "en-US",
  "badgeImageUrl":
    "https://res.cloudinary.com/myprogress/image/upload/w_300,h_300,c_pad/v1664161819/uploaded_images/klwgkhj3nouyfg61rrej.png",
  "status": 1,
  "expiryUnit": 0,
  "expiryValue": 0
}]
```

Success (200): summaryOnly=false

```
[ {
  "id": "OB-9f68ed87-7e46-45be-8231-4af73aef0172",
  "name": "Digital Badge Fundamentals - Test",
  "description": "Badge for testing",
  "issuerId": "327649a7-8654-4fee-b117-09e009679636",
  "issuer": {
    "id": "327649a7-8654-4fee-b117-09e009679636",
    "name": "Credentialate stage issuer",
    "description": "Staging issuer for Credentialate",
    "url": "https://credentialate.com/",
    "logoUrl":
      https://res.cloudinary.com/myprogress/image/upload/w\_300/v1665293985/uploaded\_images/omomo5reutu5fgnxckzm.png
  },
  "graphicId": "759682d4-c7a7-405c-935f-8371fd8f34ad",
  "graphic": {
    "id": "759682d4-c7a7-405c-935f-8371fd8f34ad",
    "url":
      https://res.cloudinary.com/myprogress/image/upload/w\_300,h\_300,c\_pad/v1665294040/uploaded\_images/rmmyfdodxd3l5zizm7z9.png
  },
  "criteriaId": "b4706abf-746a-4859-97db-37ceb6deabde",
```

```

"criteria": {
  "id": "b4706abf-746a-4859-97db-37ceb6deabde",
  "type": 0,
  "value": "<p>This is a badge for testing purposes</p>",
  "url": http://mybadges-openbadges-test.mkmapps.com//app/#!/criteria?badgeid=9f68ed87-7e46-45be-8231-4af73aef0172
},

"category": 0,
"tags": [],
"skills": [
  {
    "text": "<p>Some skills!</p>",
    "ordinal": 0
  }
],
"familyId": "759682d4-c7a7-405c-935f-8371fd8f34ad",
"family": {
  "Id": "759682d4-c7a7-405c-935f-8371fd8f34ad ",
  "Name": "Family Name",
  "Description": "The family is used to categorise a badge template",
  "DateCreated": "2022-10-11T09:36:08.677Z",
  "DateUpdated": "2022-10-11T09:36:08.677Z",
  "DateDeleted": "2022-10-11T09:36:08.677Z"
},
"dateCreated": "2022-10-09T05:41:41.437",
"dateUpdated": "2022-10-09T05:42:14.053",
"dateDeleted": null,
"datePublished": "2022-10-09T05:42:16",
"totalIssued": 4,
"customData": [],
"expires": null,
"language": "en-GB",
"badgeImageUrl":
"https://res.cloudinary.com/myprogress/image/upload/w_300,h_300,c_pad/v1665294040/uploaded_images/rmmyfdodxd3l5izm7z9.png",
"status": 1,
"expiryUnit": 0,
"expiryValue": 0
}
]

```

Unauthorised (401)

```
{ "message": "Authorization has been denied for this request." }
```

Unknown Error (500)

```
{  
  "success": false,  
  "statusCode": 500,  
  "message": "An Unknown Error has Occurred",  
  "data": null  
}
```

Get Badge Template By ID

Retrieves a single badge template and its details.

Property	Description
URL	{{Badges}}/api/issuedbadgesexternal/badgeTemplate/{BadgeTemplateId}
Methods	GET
Authentication	Required

Required Fields:

Field	Description
BadgeTemplateId (string)	The Badge Template ID (AKA ActivityId). This ID should start with "OB-"

Example Request:

```
GET {{Badges}}/api/issuedbadgesexternal/badgeTemplate/OB-f22932c6-f3aa-4dd2-a81b-04897bcb764b
Content-Type: application/json
ApiKey: API_KEY
Authorization: Bearer API_TOKEN
```

Example Responses:

Success (200):

```
[ {
  "id": "OB-9f68ed87-7e46-45be-8231-4af73aef0172",
  "name": "Digital Badge Fundamentals - Test",
  "description": "Badge for testing",
  "issuerId": "327649a7-8654-4fee-b117-09e009679636",
  "issuer": {
    "id": "327649a7-8654-4fee-b117-09e009679636",
    "name": "Credentialate stage issuer",
    "description": "Staging issuer for Credentialate",
    "url": "https://credentialate.com/",
    "logoUrl":
https://res.cloudinary.com/myprogress/image/upload/w\_300/v1665293985/uploaded\_images/omomo5reutu5fgnxckzm.png
  },
  "graphicId": "759682d4-c7a7-405c-935f-8371fd8f34ad",
  "graphic": {
    "id": "759682d4-c7a7-405c-935f-8371fd8f34ad",
    "url":
https://res.cloudinary.com/myprogress/image/upload/w\_300,h\_300,c\_pad/v1665294040/uploaded\_images/rmmyfdodxd3l5zizm7z9.png
  }
}
```

```

},
"criteriaId": "b4706abf-746a-4859-97db-37ceb6deabde",
"criteria": {
  "id": "b4706abf-746a-4859-97db-37ceb6deabde",
  "type": 0,
  "value": "<p>This is a badge for testing purposes</p>",
  "url": http://mybadges-openbadges-test.mkmapps.com//app/#!/criteria?badgeid=9f68ed87-7e46-45be-8231-4af73aef0172
},

"category": 0,
"tags": [],
"skills": [
  {
    "text": "<p>Some skills!</p>",
    "ordinal": 0
  }
],
"familyId": "759682d4-c7a7-405c-935f-8371fd8f34ad",
"family": {
  "id": "759682d4-c7a7-405c-935f-8371fd8f34ad ",
  "Name": "Family Name",
  "Description": "The family is used to categorise a badge template",
  "DateCreated": "2022-10-11T09:36:08.677Z",
  "DateUpdated": "2022-10-11T09:36:08.677Z",
  "DateDeleted": "2022-10-11T09:36:08.677Z"
},
"dateCreated": "2022-10-09T05:41:41.437",
"dateUpdated": "2022-10-09T05:42:14.053",
"dateDeleted": null,
"datePublished": "2022-10-09T05:42:16",
"totalIssued": 4,
"customData": [],
"expires": null,
"language": "en-GB",
"badgeImageUrl":
"https://res.cloudinary.com/myprogress/image/upload/w_300,h_300,c_pad/v1665294040/uploaded_images/rmmyfdodxd3l5zizm7z9.png",
"status": 1,
"expiryUnit": 0,
"expiryValue": 0
}
]

```

Unauthorised (401)

```
{ "message": "Authorization has been denied for this request." }
```

Unknown Error (500)

```
{
  "success": false,
  "statusCode": 500,
  "message": "An Unknown Error has Occured",
  "data": null
}
```

Issued Badges

Issuing a badge

Issues a badge to a specified recipient

<i>Property</i>	<i>Description</i>
<i>URL</i>	{{Events}}/api/ActivityEvents
<i>Methods</i>	POST
<i>Authentication</i>	Required

Required Fields:

<i>Field</i>	<i>Description</i>
<i>ActivityId</i> (string:100)	The id of the badge template. This can be in the learning vault dashboard > issuing badges > manual entry > summary > badge id The id will always start with "ob-"
<i>ActivityTime</i> (string:datetime)	The date/Time for when the badge was valid from. Must be a UTC date/time in the "Round trip" date/time format (which is an ISO 8601 format). For example: "2020-11-11T08:51:02.465Z".
<i>UserId</i> (string:255)	The unique identifier of the user the event relates to. Duplicate badges cannot be created for the same User Id. If a request is created for the same badge (Activity Id) and User Id a duplicate badge will not be created. <i>We strongly recommend using the Email Address as the Unique Identifier</i>
<i>Email</i> (string)	The email address of the user the event relates to

Optional Fields:

Field	Description
<i>FirstName</i> (string)	The first name of the user the event relates to
<i>LastName</i> (string)	The last name of the user the event relates to
<i>Evidence</i> (array)	An Array of evidence with the following fields: • URL (string): A URL pointing towards the evidence • Name (string): The name of the evidence (e.g. Results) • Description (string): A short description describing the type of evidence supplied • Public (boolean): Set to true to allow the evidence to be publicly available
<i>Int1</i> (integer)	Can be used for rule processing
<i>Int2</i> (integer)	Can be used for rule processing
<i>Date1</i> (string:datetime)	Can be used for rule processing Must be a UTC date/time in the "Round trip" date/time format (which is an ISO 8601 format). For example: "2020-11-11T08:51:02.465Z".
<i>Text1</i> (string:255)	Can be used for rule processing
<i>Text2</i> (string:255)	Can be used for rule processing

Example Request:

```
POST {{Events}}/api/ActivityEvents
Content-Type: application/json
ApiKey: API_KEY
Authorization: Bearer API_TOKEN

{
  "ActivityId": "0B-f22932c6-f3aa-4dd2-a81b-04897bcb764b ",
  "ActivityTime": "2022-08-02T00:50:09.095Z",
  "UserId": "Jamie@example.com",
  "Email": "jamie@example.com",
  "FirstName": "Jamie",
  "LastName": "Citizen",
  "Evidence": [{
    "Url": "https://example.com/results",
    "Name": "Results",
    "Description": "Documentation of record of results",
    "Public": true
  }]
}
```

Example Responses:

Success (200):

Empty Response Body

Invalid Model (401)

```
{
  "message": "Model validation failed",
  "errors": [
    {
      "property": "dto.ActivityId",
      "message": "The ActivityId field is required."
    }, {
      "property": "dto.UserId",
      "message": "The UserId field is required."
    }
  ]
}
```

Get The ID of the issued Badge

After issuing a badge, the following endpoint can be called to retrieve the last badge ID for the User / Badge Template.

<i>Property</i>	<i>Description</i>
<i>URL</i>	{{BadgesDomain}}/api/issuedbadgesexternal/Badgeld
<i>Methods</i>	GET
<i>Authentication</i>	Required

Required Fields:

<i>Field</i>	<i>Description</i>
<i>Email</i> (string)	The email of the recipient
<i>ActivityId</i> (string)	The Badge Template ID (AKA ActivityId). This ID should start with "OB-"

Example Request:

```
GET {{Badges}}/api/issuedbadgesexternal/BadgeId?email=
jamie@example.com&ActivityId= OB-f22932c6-f3aa-4dd2-a81b-04897bcb764b
Content-Type: application/json
ApiKey: API_KEY
Authorization: Bearer API_TOKEN
```

Example Responses:

Success (200):

```
{  "badgeId": "f8ab4e0c-79e2-4eb6-97c5-34079b306399" }
```

Unauthorised (401)

```
{  "message": "Authorization has been denied for this request." }
```

Badge Not Found (404)

```
{
  "success": false,
  "statusCode": 404,
  "message": "Badge Not found",
  "data": null
}
```

Unknown Error (500)

```
{
  "success": false,
  "statusCode": 500,
  "message": "Guid should contain 32 digits with 4 dashes.",
  "data": null
}
```

Getting a user's backpack

A user's backpack is a listing of all badges issued to a user based on their email address that have been issued by your account which is identified by your ApiKey.

Property	Description
URL	{{Badges}} /api/issuedbadgesexternal/email
Methods	GET
Authentication	Required

Required Fields:

Field	Description
<i>email</i> (string)	The email address of the backpack

Optional Querystring Fields:

Field	Description
<i>searchQuery</i> (string)	The search query will filter the results with ones that have a name that's contains the value or a tag that contains the value. <i>Note: The value should be URL encoded</i>
<i>skip</i> (integer)	The skip parameter will skip a given number or records from the results <i>Note: The Take parameter is required when using skip.</i>
<i>Take</i> (integer)	The take parameter limits the number of results by the given value. Skip / Take can be used for paging results.

Example Request:

```
GET {{Badges}} /api/issuedbadgesexternal/email?email= Jamie@example.com
Content-Type: application/json
ApiKey: API_KEY
Authorization: Bearer API_TOKEN
```

Example Responses:

Success (200):

```
[ {
  "id": "ce76091d-c136-4df9-9dcf-2e92f3ba03e4",
  "status": 2,
  "userId": "00000000-0000-0000-0000-000000000000",
  "email": "david+20221006_2@learningvault.com.au",
  "badgeId": "f8ab4e0c-79e2-4eb6-97c5-34079b306399",
  "error": null,
  "dateBaked": "2022-10-06T12:35:20.72",
  "dateIssued": "2022-10-06T12:35:27.373",
  "dateCreated": "2022-10-06T12:35:16.047",
  "dateUpdated": "2022-10-06T12:35:27.373",
  "dateDelete": null,
  "createdBy": "00000000-0000-0000-0000-000000000000",
  "badgeTemplateId": "f22932c6-f3aa-4dd2-a81b-04897bcb764a",
  "badgeName": "Laptop 2",
  "badgeDescription": "Laptop 2",
  "bakedBadgeUrl": "https://openbadges.blob.core.windows.net/bakedbadges-au-
uat/f22932c6-f3aa-4dd2-a81b-04897bcb764a-ce76091d-c136-4df9-9dcf-
2e92f3ba03e4.png",
  "name": "Jamie Citizen",
  "issuedByName": "Learning Vault",
  "expired": null,
  "revoked": null,
  "expiredDate": null,
  "revokedDate": null,
  "testimonial": null,
  "createdByOrganisationId": "00000000-0000-0000-0000-000000000000",
  "summaryUrl": "https://openbadges.blob.core.windows.net/badges-au-uat/assertion-
f8ab4e0c-79e2-4eb6-97c5-34079b306399.json",
  "expiryDate": null,
  "openBadgeId": "f8ab4e0c-79e2-4eb6-97c5-34079b306399"
}
```

Unauthorised (401)

```
{ "message": "Authorization has been denied for this request." }
```

Revoking a Badge

Revokes a badge that has been issued to a user

<i>Property</i>	<i>Description</i>
<i>URL</i>	{{Badges}}/api/issuedbadgesexternal/instruct
<i>Methods</i>	POST
<i>Authentication</i>	Required

Required Fields:

<i>Field</i>	<i>Description</i>
<i>email</i> (string)	The email address of the recipient
<i>BadgeId</i>	The Issued Badge ID
<i>Instruction</i>	Always "R"

Example Request:

```
GET {{Badges}}/api/issuedbadgesexternal/instruct
Content-Type: application/json
ApiKey: API_KEY
Authorization: Bearer API_TOKEN

{
  "UniqueId": "jamie@example.com",
  "BadgeId": "a409d2ac-940e-4813-af52-50e57c670556",
  "Instruction": "R"
}
```

Example Responses:

Success (200):

```
"Badge R Completed"
```

Badge Not Found (400)

```
{ "message": "Badge with id = a409d2ac-940e-4813-af52-50e57c670556 not found" }
```

Unauthorised (401)

```
{ "message": "Authorization has been denied for this request." }
```

Expire a Badge

Expires a badge that has been issued to a user. Same endpoint as Revoking a badge, except with a different Instruction.

Property	Description
URL	{{Badges}}/api/issuedbadgesexternal/instruct
Methods	POST
Authentication	Required

Required Fields:

Field	Description
<i>email</i> (string)	The email address of the recipient
<i>BadgeId</i>	The Issued Badge ID
<i>Instruction</i>	Always "Ex"
<i>DateTime</i>	Must be a UTC date/time in the "Round trip" date/time format (which is an ISO 8601 format). For example: "2020-11-11T08:51:02.465Z".

Example Request:

```
GET {{Badges}}/api/issuedbadgesexternal/instruct
Content-Type: application/json
ApiKey: API_KEY
Authorization: Bearer API_TOKEN

{
  "UniqueId": "jamie@example.com",
  "BadgeId": "a409d2ac-940e-4813-af52-50e57c670556",
  "Instruction": "R",
  "DateTime": "2022-10-06T09:20:28.098Z"
}
```

Example Responses:

Success (200):

```
"Badge Ex Completed"
```

Badge Not Found (400)

```
{ "message": "Badge with id = a409d2ac-940e-4813-af52-50e57c670556 not found" }
```

Unauthorised (401)

```
{ "message": "Authorization has been denied for this request." }
```

Getting a Badge Item / Backpack Item / Issued Badge

Revokes a badge that has been issued to a user

Property	Description
URL	{{Badges}}/api/badgeitem/{badgeId}
Methods	GET
Authentication	Required

Required Fields:

Field	Description
BadgeId	The Issued Badge ID

Example Request:

```
GET {{Badges}}/api/badgeitem/ a409d2ac-940e-4813-af52-50e57c670556
Content-Type: application/json
ApiKey: API_KEY
Authorization: Bearer API_TOKEN
```

Example Responses:

Success (200):

```
{
  "id": "66df0acc-7466-4ba8-88da-44f1ca31bbd1",
  "status": 2,
  "userId": "00000000-0000-0000-0000-000000000000",
  "email": david+20221006\_2@learningvault.com.au,
  "openBadgeId": "d7aca1c7-16ea-4019-9f08-78c104fc09d2",
  "error": null,
  "dateBaked": "2022-10-06T09:34:30.91",
  "dateIssued": "2022-10-06T09:34:33.827",
  "dateCreated": "2022-10-06T09:34:25.793",
  "dateUpdated": "2022-10-06T09:36:08.59",
  "dateDelete": null,
  "createdBy": "00000000-0000-0000-0000-000000000000",
  "createdByName": "Automated Badge Issued",
  "badgeTemplateId": "f22932c6-f3aa-4dd2-a81b-04897bcb764a",
  "badgeTemplate": {
```

```
"claimCodes": [],
"criteria": {
  "id": "594326c5-ed1c-4196-9e1a-4d60f0461668",
  "type": 0,
  "value": "<p>asdasd</p>",
  "url": http://mybadges-openbadges-test.mkmapps.com/app/#!/criteria?badgeid=f22932c6-f3aa-4dd2-a81b-04897bcb764a,
  "dateCreated": "2022-08-31T12:31:13.9",
  "dateUpdated": "2022-08-31T12:31:13.9"
},
"customData": [],
"family": null,
"graphic": {
  "organisations": [
    {
      "id": "dcb13d81-dc0d-43ba-a3f6-f7a7464c95e5",
      "badgeGraphicId": "6a05977d-57b1-45e0-a808-458c25150d7b",
      "badgeGraphic": null,
      "organisationId": "a190a4c4-8183-49f2-a3cd-af509d2646f8",
      "organisation": null,
      "dateCreated": "2022-08-13T06:47:44.89",
      "dateUpdated": "2022-08-13T06:47:44.89",
      "dateDeleted": null,
      "createdBy": "ed89dbfa-ef81-43d1-9e6e-8c9fcb3129d2"
    }
  ],
  "id": "6a05977d-57b1-45e0-a808-458c25150d7b",
  "url": https://res.cloudinary.com/myprogress/image/upload/w\_300,h\_300,c\_pad/v1660373265/uploaded\_images/abszcnqsz8wakag8o3k.png,
  "name": "lapi",
  "description": "lapi",
  "originatorId": "a190a4c4-8183-49f2-a3cd-af509d2646f8",
  "ownerType": 0,
  "source": 1,
  "status": 1,
  "createdBy": "ed89dbfa-ef81-43d1-9e6e-8c9fcb3129d2",
  "lastUpdatedBy": "00000000-0000-0000-0000-000000000000",
  "dateCreated": "2022-08-13T06:47:44.817",
  "dateUpdated": "2022-08-13T06:47:44.817",
  "dateDeleted": null
},
"issuer": {
  "id": "ab0c6b90-ba29-4ae2-912c-af7cf9049ca7",
  "name": "Learning Vault",
  "description": "Learning Vault is a solutions-focused education company, specialising in building resources for the workforce of the future.",
  "url": https://www.learningvault.com.au/,
  "logoUrl": https://res.cloudinary.com/myprogress/image/upload/w\_300/v1593791692/uploaded\_images/regfradbjbhol2hgucwy.png,
  "jsonUrl": https://openbadges.blob.core.windows.net/badges-au/issuer-
```

```
ab0c6b90-ba29-4ae2-912c-af7cf9049ca7.json,
  "createdBy": "00000000-0000-0000-0000-000000000000",
  "withdrawn": null,
  "dateCreated": "2020-06-15T21:10:54.793",
  "dateUpdated": "2020-06-15T21:10:54.793",
  "dateDeleted": null,
  "organisationId": "a190a4c4-8183-49f2-a3cd-af509d2646f8",
  "organisation": null
},
"skills": [],
"tags": [],
"id": "f22932c6-f3aa-4dd2-a81b-04897bcb764a",
"name": "Laptop 2",
"description": "test badge",
"status": 1,
"category": 0,
"expiryUnit": 0,
"expiryValue": 0,
"availabilityStartDate": null,
"availabilityEndDate": null,
"emailText": "Congratulations.\n\nYou have been issued with a Badge by
[BadgeIssuer].\n\nIf you have any questions about this email then please contact the
badge issuer.",
"expires": null,
"language": "en-AU",
"dateCreated": "2022-08-31T12:31:13.897",
"dateUpdated": "2022-08-31T12:31:13.897",
"dateDeleted": null,
"dateErroredOn": "2022-10-07T14:12:30.877",
"datePublished": "2022-08-31T12:31:17.26",
"createdBy": "162c143b-e412-473f-ae0a-f98e56221897",
"issuerId": "ab0c6b90-ba29-4ae2-912c-af7cf9049ca7",
"graphicId": "6a05977d-57b1-45e0-a808-458c25150d7b",
"criteriaId": "594326c5-ed1c-4196-9e1a-4d60f0461668",
"familyId": null,
"organisations": null,
"discoverable": null,
"jsonUrl": https://openbadges.blob.core.windows.net/badges-au-uat/badge-
f22932c6-f3aa-4dd2-a81b-04897bcb764a.json,
"errorCount": 1,
"errorUserId": "00000000-0000-0000-0000-000000000000"
},
"bakedBadgeUrl": https://openbadges.blob.core.windows.net/bakedbadges-au-
uat/f22932c6-f3aa-4dd2-a81b-04897bcb764a-66df0acc-7466-4ba8-88da-
44f1ca31bbd1.png,
"name": null,
"issuedByName": null,
"recipientName": "Jamie Citizen",
"expired": true,
"revoked": null,
"expiredDate": "2022-10-06T09:36:08.753",
"revokedDate": null,
```

```
"lastVerified": null,  
"verified": null,  
"dateExpires": "2022-10-06T09:20:28.097",  
"customData": [],  
"evidence": [],  
"testimonial": null,  
"createdByOrganisationId": "a190a4c4-8183-49f2-a3cd-af509d2646f8",  
"customSkills": null  
}
```

Badge Id not supplied (400)

```
{ "message": "id must not be empty" }
```

Badge Not Found (400)

```
{ "message": "Badge with id = a409d2ac-940e-4813-af52-50e57c670556 not found" }
```

Unauthorised (401)

```
{ "message": "Authorization has been denied for this request." }
```

Field Definitions

Below outline the definitions for some of the fields that need more explanation.

<i>Field</i>	<i>Models</i>	<i>Description</i>
<i>activityId</i>	BadgeTemplate, Activities	The Badge Template ID (AKA ActivityId) Values: starts with "OB-"
<i>bakedBadgeUrl</i>	IssuedBadge	The URL of the image for a badge that has been baked (completed creation) Values: URL to graphic
<i>expiryUnit</i>	BadgeTemplate	The unit of measurement that the expiryValue refers to in a BadgeTemplate Values: • 0 = Day • 1 = Month • 2 = Year
<i>instruction</i>	BadgeInstruction	The value used to determine the type of badge instruction to perform Values: • 'Ex' (expire) • 'R' (revoke) • 'Ext' (extend)
<i>Status</i>	BadgeTemplate	The status of a badge template can be: • 0 = Draft (Not ready to be issued) • 1 = Published (Can be issued) • 2 = With Drawn (No longer issuable)
<i>Status</i>	IssuedBadge	The status of an issued badge can be: • NotBaked • Baked • Issued • Expired • Revoked • InProgress
<i>Total Issued</i>	BadgeTemplate	Total number of badges issued for the template over its lifetime
<i>type</i>	BadgeCriteria	determines the type of criteria used in the badge criteria field Values: • 0 = TextBlock • 1 = Url