Cover Art Archive / API

musicbrainz.org/doc/Cover_Art_Archive/API

Any requests to fetch cover art must go through the coverartarchive.org service. There are 3 classes of end points, text surrounded in braces indicates a variable.

/release/{mbid}/

Summary

Fetches a JSON listing of available cover art for a MusicBrainz release.

Accepted Methods

- GET

- HEAD

Responses

- 307 redirect to an index.json file, if there is a release with this MBID.

- 400 if {mbid} cannot be parsed as a valid UUID.

- 404 if there is no release with this MBID.

- 405 if the request method is not one of GET or HEAD.

- 406 if the server is unable to generate a response suitable to the Accept header.

- 503 if the user has exceeded their rate limit.

Example

```
> GET /release/76df3287-6cda-33eb-8e9a-044b5e15ffdd HTTP/1.1
> Host: coverartarchive.org
> Accept: application/json
```

```
< HTTP/1.0 200 OK
< Status: 200
{
  "images":[
     {
        "types":[
           "Front"
        ],
        "front":true,
        "back":false,
        "edit":17462565,
        "image":"http://coverartarchive.org/release/76df3287-6cda-33eb-8e9a-044b5e15ffdd/829521842.jpg",
        "comment":"",
        "approved":true,
        "id":"829521842",
        "thumbnails":{
          "250":"http://coverartarchive.org/release/76df3287-6cda-33eb-8e9a-044b5e15ffdd/829521842-250.jpg",
          "500":"http://coverartarchive.org/release/76df3287-6cda-33eb-8e9a-044b5e15ffdd/829521842-500.jpg",
          "1200":"http://coverartarchive.org/release/76df3287-6cda-33eb-8e9a-044b5e15ffdd/829521842-1200.jpg",
          "small":"http://coverartarchive.org/release/76df3287-6cda-33eb-8e9a-044b5e15ffdd/829521842-250.jpg",
          "large":"http://coverartarchive.org/release/76df3287-6cda-33eb-8e9a-044b5e15ffdd/829521842-500.jpg"
        }
     }
  ],
  "release":"http://musicbrainz.org/release/76df3287-6cda-33eb-8e9a-044b5e15ffdd"
}
```

/release/{mbid}/front

Summary

Fetches the image that is most suitable to be called the "front" of a release. This is intentionally vague, and users will help curate this data into something that is meaningful, but here you'll find the artwork that users would most likely expect to see in:

- Digital shops when searching for the release
- Their portable media player
- The folder icon in their file browser, if supported
- What they would expect to find if they were looking for this release in a shop.

Accepted Methods

- GET

- HEAD

Responses

- 307 if the community have decided upon a "front" image for this release.

- 400 if {mbid} cannot be parsed as a valid UUID.

- 404 if there is either no release with this MBID, or the community have not chosen an image to represent the front of a release.

- 405 if the request method is not GET or HEAD.

- 503 if the user has exceeded their rate limit.

Example

```
   > GET /release/76df3287-6cda-33eb-8e9a-044b5e15ffdd/front HTTP/1.1
   > Host: coverartarchive.org

   < HTTP/1.1 307 Temporary Redirect
   < Status: 307
   < Location: http://s3.us.archive.org/mbid-76df3287-6cda-33eb-8e9a-
044b5e15ffdd/mbid-76df3287-6cda-33eb-8e9a-044b5e15ffdd-829521842.jpg
```

/release/{mbid}/back

Summary

Fetches the image that is most suitable to be called the "back" of a release. This is intentionally vague, and users will help curate this data into something that is meaningful, but here you'll usually find artwork that includes:

- A tracklisting, barcode, and label
- What users would expect to find on the back cover if they were looking for this release in a shop.

Accepted Methods

- GET

- HEAD

Responses

- 307 if the community have decided upon a "back" image for this release.

- 400 if {mbid} cannot be parsed as a valid UUID.

- 404 if there is either no release with this MBID, or the community have not chosen an image to represent the back of a release.

- 405 if the request method is not GET or HEAD.

- 503 if the user has exceeded their rate limit.

Example

```
   > GET /release/99b09d02-9cc9-3fed-8431-f162165a9371/back HTTP/1.1
   > Host: coverartarchive.org

   < HTTP/1.1 307 Temporary Redirect
   < Status: 307
   < Location: http://archive.org/download/mbid-99b09d02-9cc9-3fed-8431-
f162165a9371/mbid-99b09d02-9cc9-3fed-8431-f162165a9371-135822686.jpg
```

/release/{mbid}/{id}

Summary

Fetches a specific piece of artwork. The possible {id} values can be found by parsing the response of a /release/{mbid} request.

Accepted methods

- GET

- HEAD

Responses

- 307 redirect to a binary image.

- 404 if a release with this MBID cannot be found.

- 405 if the request method is not GET or HEAD.

- 503 if the user has exceeded their rate limit.

Example

```
   > GET /release/foo/135741621.jpg HTTP/1.1
   > Host: coverartarchive.org

   < HTTP/1.1 307 Temporary Redirect
   < Status: 307
   < Location: http://archive.org/download/mbid-99b09d02-9cc9-3fed-8431-
f162165a9371/mbid-99b09d02-9cc9-3fed-8431-f162165a9371-135741621.jpg
```

/release/{mbid}/({id}|front|back)-(250|500|1200)

Summary

Fetches a thumbnail for a specific piece of artwork. The possible {id} values can be found by parsing the response of a /release/{mbid} request. The current supported thumbnail sizes are 250px, 500px, and 1200px.

Accepted methods

- GET

- HEAD

Responses

- 307 redirect to a binary image. This redirected request may resolve to a 404 if the thumbnail does not exist.

- 404 if a release with this MBID cannot be found.

- 405 if the request method is not GET or HEAD.

- 503 if the user has exceeded their rate limit.

Example

```
  > GET /release/99b09d02-9cc9-3fed-8431-f162165a9371/135741621-250.jpg HTTP/1.1
  > Host: coverartarchive.org

  < HTTP/1.1 307 Temporary Redirect
  < Status: 307
  < Location: http://archive.org/download/mbid-99b09d02-9cc9-3fed-8431-f162165a9371/mbid-99b09d02-9cc9-3fed-8431-f162165a9371-135741621-250.jpg
```

/release-group/{mbid}/

Summary

Fetches a JSON listing of available cover art for a MusicBrainz release group, as well as the url for the specific release from which the art was sourced.

Accepted Methods

- GET

- HEAD

Responses

- 200 if there is a release group with this MBID and it has cover art.

- 400 if {mbid} cannot be parsed as a valid UUID.

- 404 if either there is no release group with this MBID, or the community have not chosen an image to represent the release group.

- 501 if the request method is not supported.

- 503 if the user has exceeded their rate limit.

Example

```
   > GET /release-group/c31a5e2b-0bf8-32e0-8aeb-ef4ba9973932 HTTP/1.1
   > Host: coverartarchive.org
   > Accept: application/json

   < HTTP/1.1 200 OK
   < Status: 200
   {
       "release":"https://musicbrainz.org/release/f268b8bc-2768-426b-901b-
c7966e76de29",
       "images":[
           {
               "edit":37284546,
               "id":"12750224075",
               "image":"http://coverartarchive.org/release/f268b8bc-2768-426b-
901b-c7966e76de29/12750224075.png",
               "thumbnails":{
                   "250":"http://coverartarchive.org/release/f268b8bc-2768-426b-
901b-c7966e76de29/12750224075-250.jpg",
                   "500":"http://coverartarchive.org/release/f268b8bc-2768-426b-
901b-c7966e76de29/12750224075-500.jpg",
                   "1200":"http://coverartarchive.org/release/f268b8bc-2768-426b-
901b-c7966e76de29/12750224075-1200.jpg",
                   "small":"http://coverartarchive.org/release/f268b8bc-2768-426b-
901b-c7966e76de29/12750224075-250.jpg",
                   "large":"http://coverartarchive.org/release/f268b8bc-2768-426b-
901b-c7966e76de29/12750224075-500.jpg"
               },
               "comment":"",
               "approved":true,
               "front":false,
               "types":[
                   "Back"
               ],
               "back":true
           }
       ]
   }
```

/release-group/{mbid}/front[-(250|500|1200)]

Summary

Fetches the image that is most suitable to be called the "front" of a release group (or one of its thumbnails). This is intentionally vague, and users will help curate this data into something that is meaningful, but here you'll find the artwork that users would most likely expect to see in:

- Digital shops when searching for the release
- Their portable media player
- The folder icon in their file browser, if supported
- What they would expect to find if they were looking for this release in a shop.

Accepted Methods

- GET

- HEAD

Responses

- 307 if the community have decided upon a "front" image for this release group.

- 400 if {mbid} cannot be parsed as a valid UUID.

- 404 if there is either no release with this MBID, or the community have not chosen an image to represent the front of a release.

- 405 if the request method is not GET or HEAD.

- 503 if the user has exceeded their rate limit.

Example

```
   > GET /release-group/48140466-cff6-3222-bd55-63c27e43190d/front HTTP/1.1
   > Host: coverartarchive.org

   < HTTP/1.1 307 Temporary Redirect
   < Status: 307
   < Location: http://archive.org/download/mbid-76df3287-6cda-33eb-8e9a-
044b5e15ffdd/mbid-76df3287-6cda-33eb-8e9a-044b5e15ffdd-829521842.jpg
```

OPTIONS support

In theory

All end points at the coverartarchive.org support the HTTP OPTIONS method, to determine which request methods are valid for each resource. On an OPTIONS request, the server will do no processing, other than returning an empty response with the 'Allow:' header field set to the support methods for that resource. For supported methods, consult the specification of each individual resource.

In practice

see CAA-56, "The documentation claims the HTTP OPTIONS method is supported but it isn't".

Rate limiting rules

There are currently no rate limiting rules in place at http://coverartarchive.org.

Cover Art Archive Metadata

The Cover Art Archive provides a collection of metadata with each release, which allows users to determine what cover art can be used for. The metadata will be stored and served as application/json. The metadata consists of a ordered list of entries, where each entry contains:

image
the full coverartarchive.org url to the original image

thumbnails
object containing the keys "250", "500", and "1200" linking to thumbnails of those sizes (in pixels); "small" and "large" are deprecated keys and are equivalent to "250" and "500", respectively

types
list of zero or more types for the image (see the list).

front
boolean, indicates if this is the "main front" (returned by /front)

back
boolean, indicates if this is the "main back" (returned by /back)

comment
a free text comment

approved
whether the image was approved by the musicbrainz edit system

edit
the edit id on musicbrainz (e.g. 123)

id
archive.org's internal file id

The metadata also contains a link to the MusicBrainz release, under the **release** field.

JSON Output Example

```
 {
    "images" : [
        {
            "types" : [ "Front" ],
            "front" : true,
            "back" : false,
            "comment" : "",
            "image" : "http://coverartarchive.org/...jpg",
            "thumbnails" : {
                "small" : "http://coverartarchive.org/...-250.jpg",
                "large" : "http://coverartarchive.org/...-500.jpg"
            },
            "approved" : true,
            "edit" : 123,
            "id" : "456"
        },
        {
            "types" : [ "Other" ],
            "front" : false,
            "back" : false,
            "comment" : "autographed by ModBot",
            "image" : "http://coverartarchive.org/...jpg",
            "thumbnails" : {
                "250" : "http://coverartarchive.org/...-250.jpg",
                "500" : "http://coverartarchive.org/...-500.jpg",
                "1200" : "http://coverartarchive.org/...-1200.jpg",
                "small" : "http://coverartarchive.org/...-250.jpg",
                "large" : "http://coverartarchive.org/...-500.jpg"
            },
            "approved" : true,
            "edit" : 124,
            "id" : "457"
        }
    ],
    "release" : "http://musicbrainz.org/release/2ba4396d-c0be-4a56-b4ea-0438306eb3be"
  }
```

File naming

Files are named using integers derived from the current high resolution system time, for example via the `Time::HiRes::time` function in Perl. The exact formula to create a new file name is:

```
int((time() - 1327528905) * 100)
```

This filename is allocated by MusicBrainz at the time of upload, and will never change.

Libraries to use the Cover Art Archive

The Cover Art Archive can be accessed with libcoverart and CoverArtArchive perl bindings.

3rd party libraries

- coverartarchive-api, Java data bindings (CoverArtArchiveClient)
- gocaa, Go client for Cover Art Archive
- MetaBrainz.MusicBrainz.CoverArt, .NET Client API (NuGet Package)
- php-cover-art-archive-api, PHP 7.4+ API wrapper