

□ Documentation

[Apple Music API](#) / Handling Requests and Responses

Article

Handling Requests and Responses

Write a request and handle responses from the API.



Overview

After adding the appropriate tokens to the header, compose your request to get data from the API and handle responses.

Compose a Request

Apple Music API requests have common components. To compose a request, first specify the root path, `https://api.music.apple.com/v1`.

Follow this part of the path with either `/catalog` or `/me`:

- The `/catalog` path accesses the public Apple Music catalog.
- The `/me` path accesses personalized content for the user such as their music library, ratings and listening history.

Follow this path with any parameters that are specific to the endpoint. For example, to request information about a specific genre, construct a URL that includes the ID of the genre in the path.

```
GET https://api.music.apple.com/v1/catalog/{storefront}/genres/{id}
```

When requesting resources from the Apple Music catalog, you must include the storefront in the path. When requesting resources from the user's personal library, replace the storefront information with the `me` string, as shown in the following example.

```
GET https://api.music.apple.com/v1/me/ratings/albums/{id}
```

Every resource has a unique identifier in the Apple Music catalog. Resources in the user's library have unique identifiers that are different than the Apple Music catalog identifiers. If a user removes an item from their music library but then adds it back later, it will have a new and different identifier.

Most requests return only the requested resource. For information about how to request related resources at the same time, see [Handling Resource Representation and Relationships](#).

Handle a Response

There are three kinds of responses: Resource Collection, Results, and Errors.

The resource collections can come in multiple common forms which include:

- Resource Collection (for example, `https://api.music.apple.com/v1/catalog/us/albums?ids=123,234`)
- Paginated Resource Collection (for example, `https://api.music.apple.com/v1/me/library-albums`)
- Relationship Response (for example, `https://api.music.apple.com/v1/catalog/us/albums/123/tracks`)
- Relationship View Response (for example, `https://api.music.apple.com/v1/catalog/us/albums/123/view/related-albums`)

For results responses, there is always a top-level `results` member object that contains the information for the response. Results responses are always unique to the endpoint, such as charts or a search response.

```
GET https://api.music.apple.com/v1/catalog/{storefront}/charts
```

```
GET https://api.music.apple.com/v1/catalog/{storefront}/search
```

Error responses contain an array of one or more error objects that indicate any issues while handling the request. The status code of the response reflects the primary error. See [Error](#) and [HTTP Status Codes](#).

Default responses include:

- If the request is for an existing single resource object, the status code is 200 (OK) and the data array contains the requested resource object.
- If the request is for a single resource object that doesn't exist, the status code is 404 (Not Found) and the response doesn't contain a data array.

- If the request is for multiple resource objects by ID, the status code is 200 (OK) and the data array includes the existing resource objects.
 - If the request is for multiple resource objects by ID and none of the resources exist, the status code is 200 (OK) and the data array is empty.
 - A successful response for a resource collection with additional results contains a next link to the next page of fetched resources in the collection. See also [Fetching Resources by Page](#).
 - A successful response for a relationship request is 200 OK and the data array contains a paginated collection of resources contained in the relationship. A next link signifies that additional resources may be fetched from the relationship's resource collection.
 - A successful response for a relationship view request is 200 OK and the data array contains a paginated collection of resources contained in the view. The relationship view's attributes are included when requested using the with parameter. A next link signifies that additional resources may be fetched.
 - If the request contains successfully modified or deleted resources, the status code is 204 (No Content) and the body is empty.
 - If you make a successful request to an endpoint that returns results, the status code is 200 (OK) and the response includes those results.
 - If the request isn't accepted because its authorization is missing or invalid due to an issue with the developer token, the status code is 401 (Unauthorized). In this case, the response doesn't contain a data array.
 - If the request isn't accepted due to an issue with the media user token or because the request is using incorrect authentication, the status code is 403 (Forbidden). In this case, the response doesn't contain a data array.
 - If the request isn't supported as specified, the status code is 400 (Bad Request) and the errors array contains an error object for any identified problem.
 - The response status code is in the 500 range when errors are encountered on the server while processing the request. In this case, the errors array contains error objects for the errors for any identified problem.
-

Topics

Related Objects

object EmptyBodyResponse

A response object that contains no content.

object ErrorsResponse

A response object indicating that an error occurred while processing the request.

object UnauthorizedResponse

A response object indicating that the request's authorization is missing or invalid.

object ForbiddenResponse

A response object indicating that the request wasn't accepted due to an issue with the authentication.

object ResourceCollectionResponse

A response object composed of resource objects for the request.

See Also

Essentials

Generating Developer Tokens

Generate a developer token needed to make requests to Apple Music API.

User Authentication for MusicKit

Authenticate requests for user data using the Music User Token.

Handling Resource Representation and Relationships

Fetch resources with extended attributes and included relationships and relationship views.

Storefronts and Localization

Pick a region-specific geographic location from which to retrieve catalog information, or retrieve information from the user's personal library.

Common Objects

Understand the common JSON objects that framework responses contain.

Managing Content Ratings, Alternate Versions, and Equivalencies

Handle multiple and alternate versions of content.

Fetching Resources by Page

Use pagination to fetch the next set of objects.