

Article

Handling Resource Representation and Relationships

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

Overview

Resources are fetched with a default representation of attributes and relationships. Extend the default representation for a resource type to fetch additional attributes and include relationships and relationship views.

Note

If any relationship or relationship view contains too many related resources, use pagination to fetch the next set of objects. See [Fetching Resources by Page](#).

Extend Resource Attributes

The server will fetch a subset of the available attributes for a resource type by default. The attributes not fetched by default are known as *extended attributes*. You may specify a set of attribute names using the `extend` query parameter to request these additional attributes for a resource type.

For example, when fetching an [Albums](#) resource, you can request that the server add the extended `artistUrl` attribute to the album's other attributes:

```
GET https://api.music.apple.com/v1/catalog/us/albums/310730204?extend=artistUrl
```

Include Resource Relationships

Responses contain some relationships and their related resources by default. Related resources in the response will be either resource objects with their attributes or resource identifiers without attributes. See how to include relationships in a single response below.

For example, these Album relationships have different default behaviors:

- The `tracks` relationship includes the resource objects for the album's songs and music videos, which are typically essential for working with albums.
- The `artists` relationship only includes resource identifiers for the artist or artists associated with the album and excludes the `attributes` member in each resource object. This relationship allows you to link to an artist from an album, although some artist attributes are also attributes of the album.
- The response doesn't include all relationships with a resource by default. For example, the response omits the `genres` relationship, though you can expressly add this relationship to the request. This seldom-used relationship is often superseded by `genreNames`, which appear as an attribute of an album.

Use the `include` parameter to specify a set of relationships to include with the resource in the response. The value of the `include` parameter is a comma-separated list of the relationship names to include. For relationships that include resource objects by default, there's no need to specify them in the list; they're included along with the specified relationships. For example, when fetching an Artists object, you can request the `playlists` objects using `include`.

```
GET https://api.music.apple.com/v1/catalog/us/artists/462006?include=playlists
```

You may request the related resources without their attributes for one or more relationships using the `relate` parameter. This is a way to quickly retrieve identifiers to related resources to fetch later. Modifying the example above, you can request related `playlists` resource identifiers for inclusion when fetching an Artists object.

```
GET https://api.music.apple.com/v1/catalog/us/artists/462006?relate=playlists
```

Include Resource Relationship Views

The response never includes relationship views with a resource by default. Use the `views` parameter to specify a set of relationship views to include with the resource in the response. The value of the `views` parameter is a comma-separated list of the relationship view names to include. For example, when fetching an Albums object, you can request the `other-versions` relationship view using `views`.

```
GET https://api.music.apple.com/v1/catalog/us/albums/310730204?views=other-versions
```

Note

You can request any combination of resource relationship and relationship views using the `include`, `relate`, and `views` parameters.

Scoping Parameters

When not classified, the `extend`, `relate`, and `include` parameters apply to the primary resource collection of the response. For example, when fetching one or more artists, applying `&include=music-videos` adds the `music-video` relationship for the artists in the response's primary data resource collection.

The `extend`, `include`, and `relate` parameters can be scoped to specifically target a resource type in the response. This applies to all resources in the response instead of just the primary resources. For example, if you wish to fetch a playlist, and would like to extend the `artistUrl` attribute on all songs resources in the response, you can extend the songs resource type specifically.

```
https://api.music.apple.com/v1/catalog/us/playlists/pl.2b0e6e332fdf4b7a91164da316212
```

If you wish to fetch an artist, and would like to include the `featured-albums` relationship view along with the tracks for those albums, you can scope the `include` parameter to the albums resource type specifically.

```
GET https://api.music.apple.com/v1/catalog/us/artists/462006?views=featured-albums&i
```

Fetch Resource Relationships Directly

To fetch a relationship exclusively, specify the name of the relationship after the `id` path parameter. For example, fetch just the tracks belonging to an album.

```
GET https://api.music.apple.com/v1/catalog/us/albums/310730204/tracks
```

By default, the API fetches a limited number of related resources for the relationship. See the corresponding object model reference for the default fetch limits. To specify the number of objects fetched, use the `limit` parameter. For example, you can fetch an artist's first five albums.

```
GET https://api.music.apple.com/v1/catalog/us/artists/462006/albums?limit=5
```

Fetch Resource Relationship Views Directly

To fetch a relationship view exclusively, you must append the `"/view/"` followed by name of the view to the `id` path parameter. For example, fetch similar-artists for an identified artist.

```
GET https://api.music.apple.com/v1/catalog/us/artists/462006/view/similar-artists
```

When fetching a relationship view directly, you may request the associated attributes using the `with` parameter.

```
GET https://api.music.apple.com/v1/catalog/us/artists/462006/view/similar-artists?with=attributes
```

By default, the API fetches a limited number of related resources for the relationship view. See the corresponding object model reference for the default fetch limits. To specify the number of objects fetched, use the `limit` parameter. For example, you can fetch the first three featured playlists for an artist.

```
GET https://api.music.apple.com/v1/catalog/us/artists/462006/view/featured-playlists?limit=3
```

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 Requests and Responses

Write a request and handle responses from the API.



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.