Nintendo Music - API Workflow Documentation (with Google OAuth2)

This document provides a detailed breakdown of the Nintendo Music mobile application's API workflows, focusing on:

- accounts.nintendo.com
- api.m.nintendo.com
- Google OAuth2 token exchange (used for authentication)

Other third-party services such as Firebase are excluded.

Nintendo API Workflow Documentation

1. Accounts API (accounts.nintendo.com)

Main Function:

Handles user authentication and account-related actions.

Key Endpoint:

- /withdraw/confirm (POST)

  Purpose: Confirm user account withdrawal.

  Payload: Unknown (app didn't provide details on exact request body).

  Response: Typically a success confirmation (200 OK) or 403 Forbidden.

2. Main Nintendo API (api.m.nintendo.com)

Main Function:

Handles music track metadata, playlists, subscriptions, DRM license requests, and virtual currency management.

Endpoints and Payloads:

- /tracks/ (GET): Fetch available music tracks. No payload.
- /userPlaylists/ (GET): Retrieve user playlists. No payload.
- /user_playlist/{playlist_id} (GET): Fetch details for a specific playlist. No payload.
- /vcm/v1/ (GET): Virtual Currency Management. No payload.
- /widevine/license (POST): Request a Widevine DRM license for audio playback. Binary DRM challenge payload (structure unknown).

Headers for all endpoints:

- Authorization: Bearer <access_token>
- User-Agent: Nintendo Music/Version
- X-Device-Token: Optional.

3. Google OAuth2 (oauth2.googleapis.com)

Main Function:

Used for exchanging the Nintendo authorization code for an access_token via Google's OAuth2 service.

Key Endpoint:

- /token (POST)

Purpose: Exchange the Nintendo authorization code for an access_token.

Payload Example:

```
{
  grant_type: authorization_code,
  code: <auth_code>,
  client_id: <nintendo_client_id>,
  client_secret: <nintendo_client_secret>,
  redirect_uri: <app_redirect_uri>
}
```

Response Example:

```
{
  access_token: <token>,
  refresh_token: <refresh_token>,
  expires_in: 3600,
  token_type: Bearer
}
```

Note: This step is mandatory to gain access to Nintendo APIs.

4. Other Detected Endpoints (Unknown Logic)

Detected endpoints/fragments in the APK without full documentation:

- /apiErrorLoggingInterceptor
- /apiWithCache
- /apiWithRightsTokenCache
- /appCheckRepository
- /appCheckLocalDataSource

These likely handle internal error logging, caching, or app check validations, but the app does not provide full context or payload structures for them.