

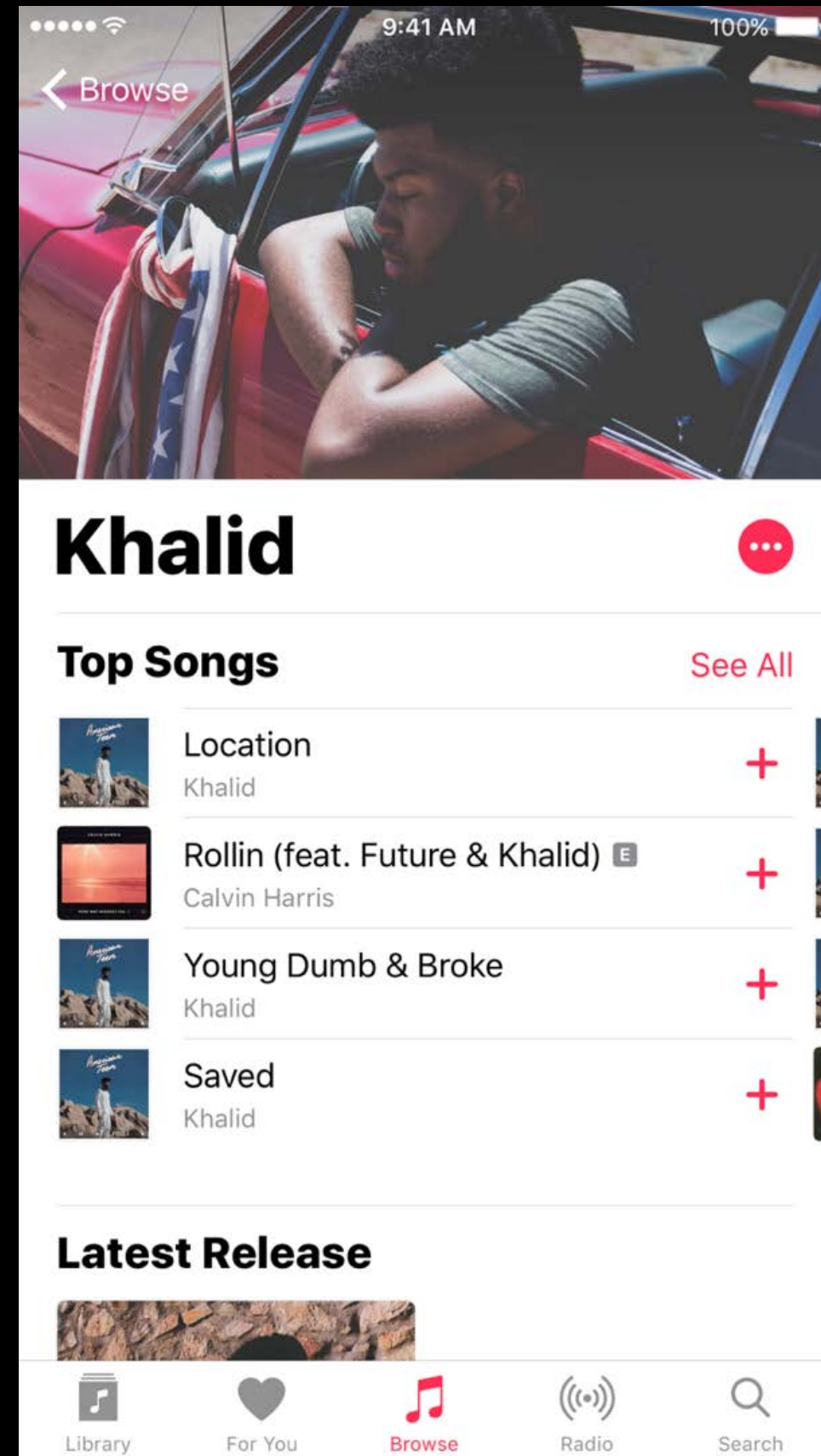
Introducing MusicKit

Session 502

Tim Parthemore, MusicKit Services
Joel Lopes Da Silva, iOS Music

Apple Music Content

Artists



Apple Music Content

Artists

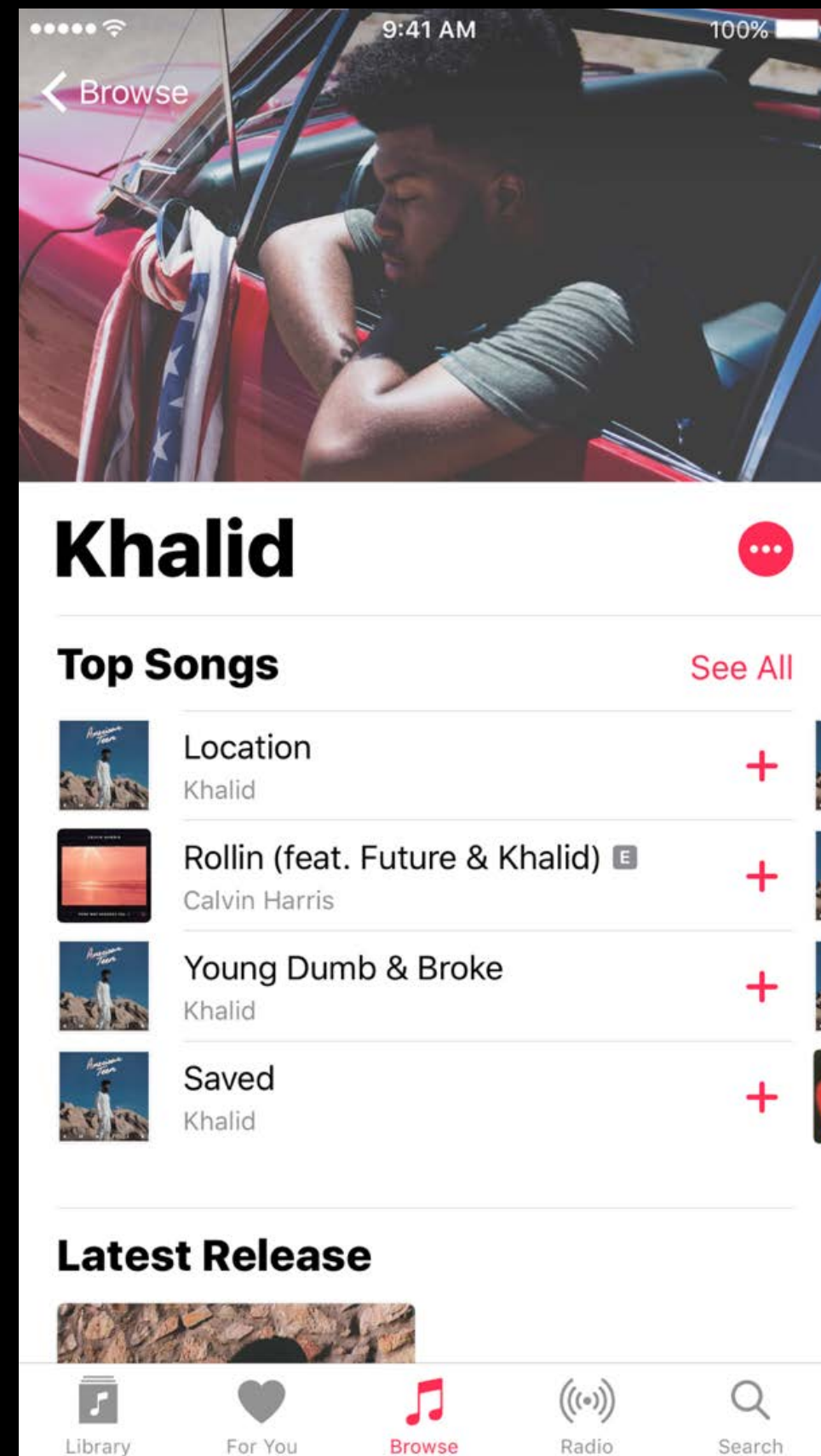
The screenshot shows the Apple Music interface for the artist Khalid. At the top, there's a 'Browse' button and a back arrow. Below that is a large image of Khalid. The artist's name 'Khalid' is prominently displayed with a red three-dot menu icon to its right. Underneath, the 'Top Songs' section is visible, featuring a list of four songs: 'Location', 'Rollin (feat. Future & Khalid)', 'Young Dumb & Broke', and 'Saved'. Each song has a small album cover icon, the song title, the artist's name, and a red plus sign to its right. A 'See All' link is positioned to the right of the 'Top Songs' header. Below this is the 'Latest Release' section, which shows a partial view of the 'American Teen' album cover. At the bottom, the navigation bar includes icons for Library, For You, Browse (highlighted in red), Radio, and Search.

Albums

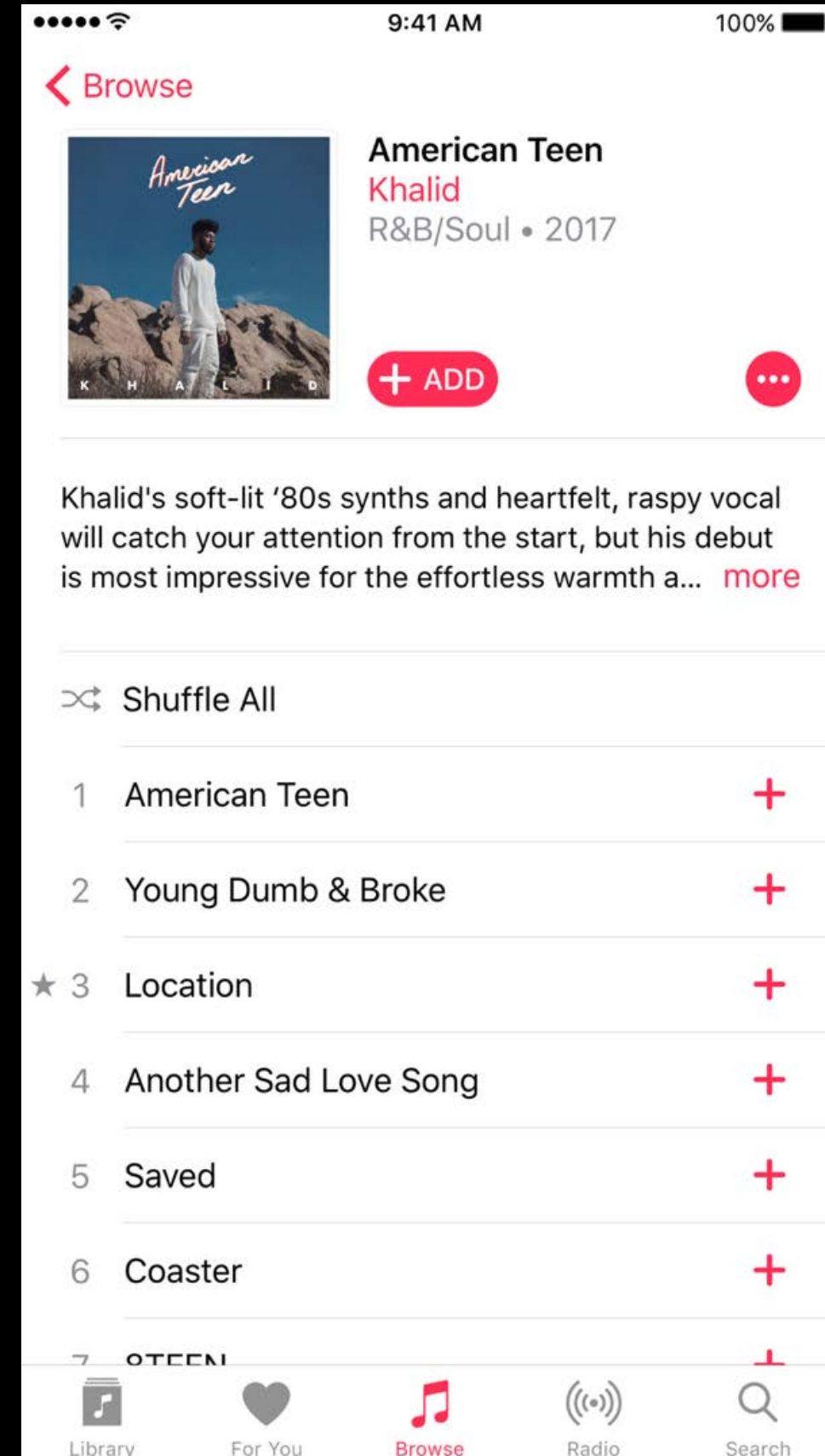
The screenshot displays the Apple Music interface for the album 'American Teen' by Khalid. It starts with a 'Browse' button and a back arrow. The album cover is shown on the left, with the title 'American Teen' and the artist 'Khalid' to its right. Below the title, it says 'R&B/Soul • 2017'. There are two buttons: a red '+ ADD' button and a red three-dot menu icon. A short description follows: 'Khalid's soft-lit '80s synths and heartfelt, raspy vocal will catch your attention from the start, but his debut is most impressive for the effortless warmth a...' followed by a 'more' link. Below the description is a 'Shuffle All' option with a shuffle icon. The track list is numbered 1 through 7, with the first six tracks visible: 'American Teen', 'Young Dumb & Broke', 'Location', 'Another Sad Love Song', 'Saved', and 'Coaster'. Each track has a red plus sign to its right. At the bottom, the navigation bar is the same as in the Artists view, with 'Browse' highlighted in red.

Apple Music Content

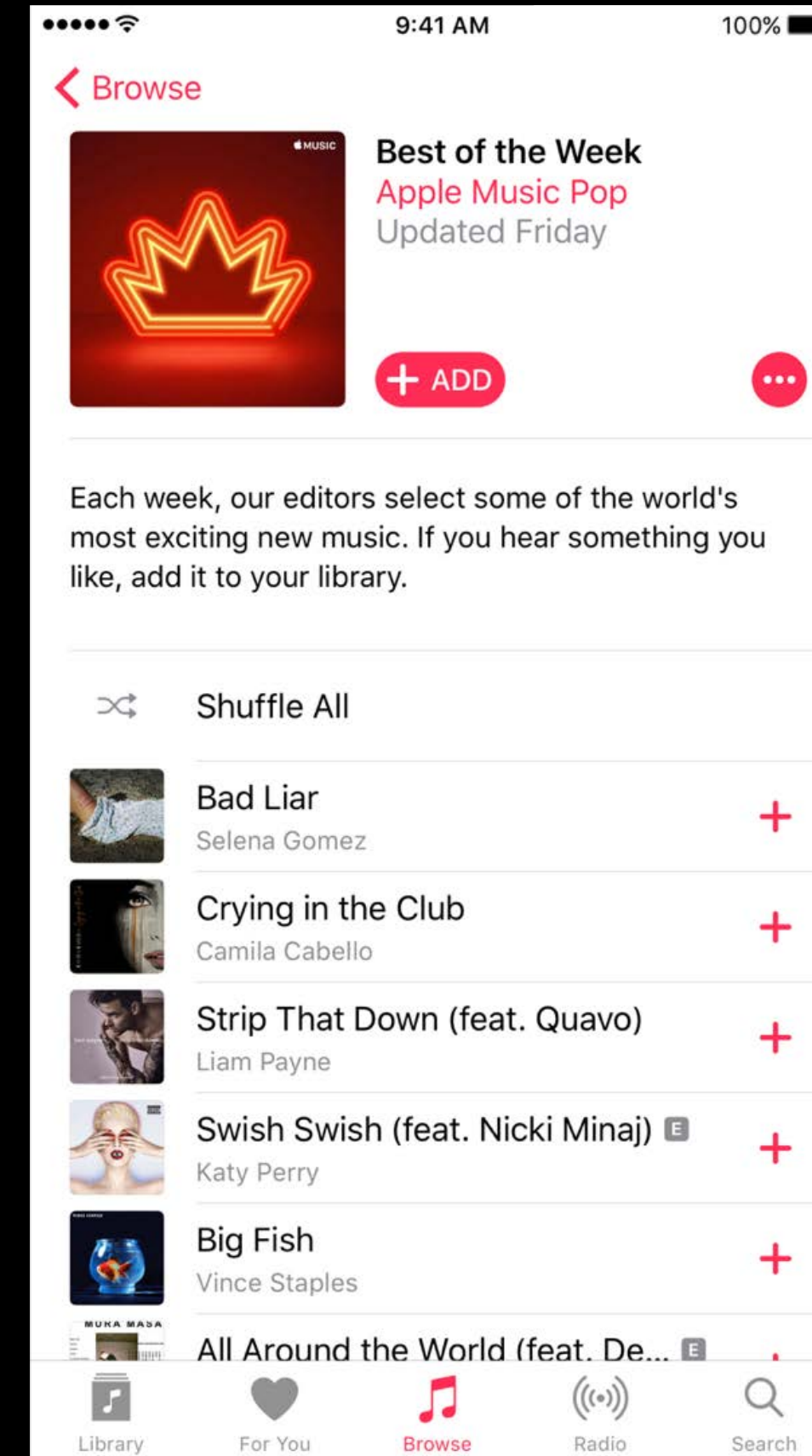
Artists



Albums

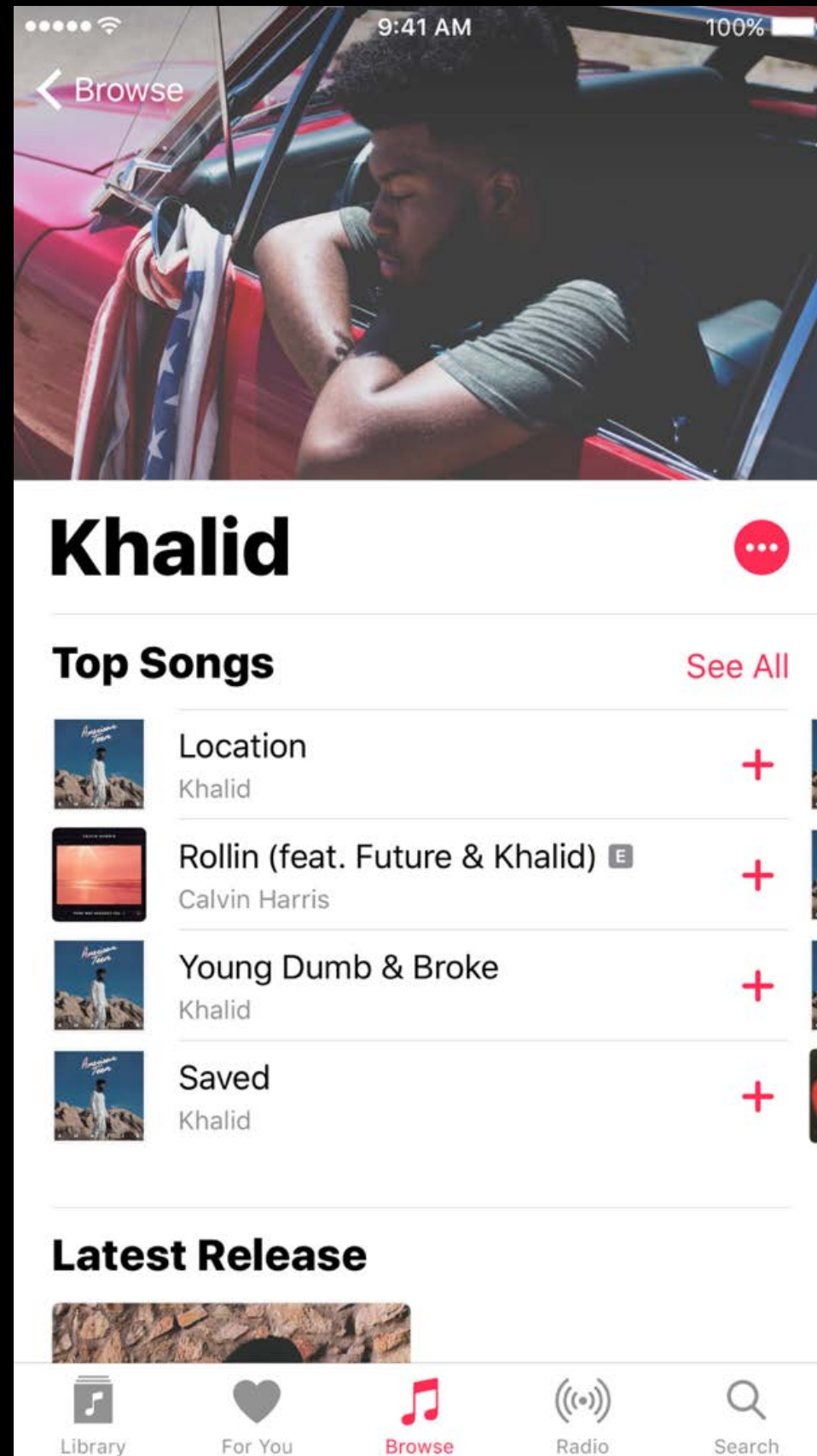


Playlists

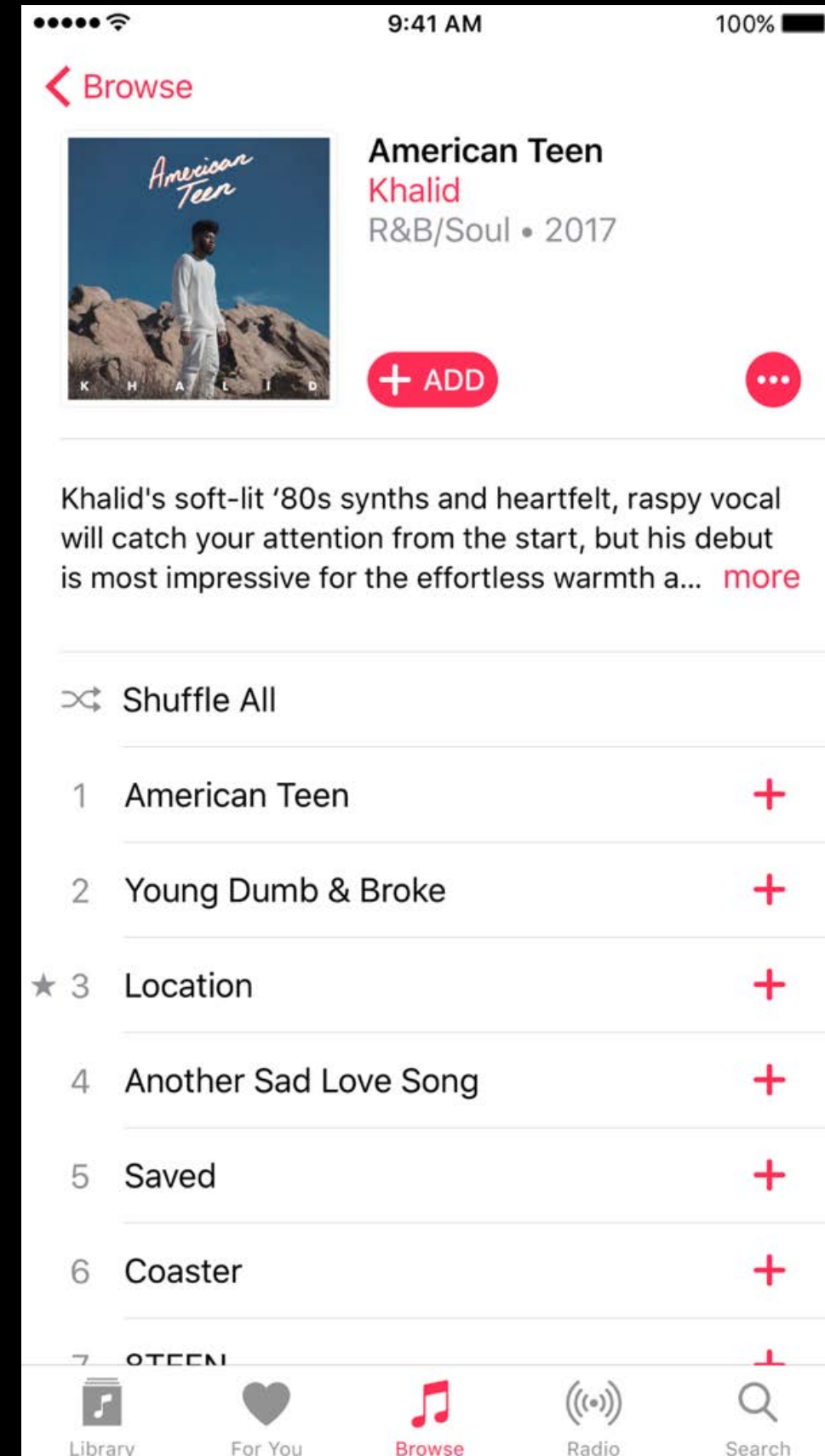


Apple Music Content

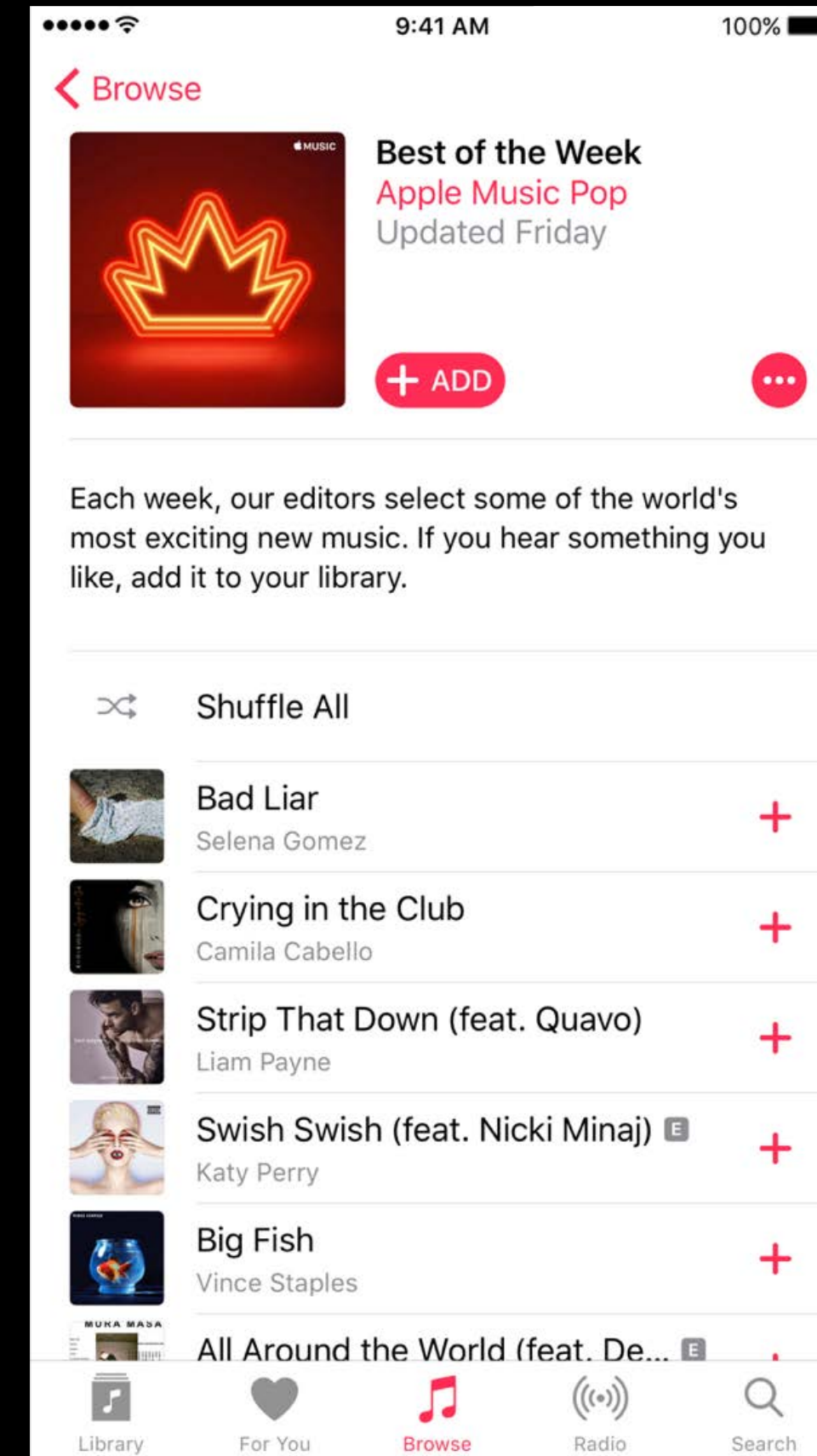
Artists



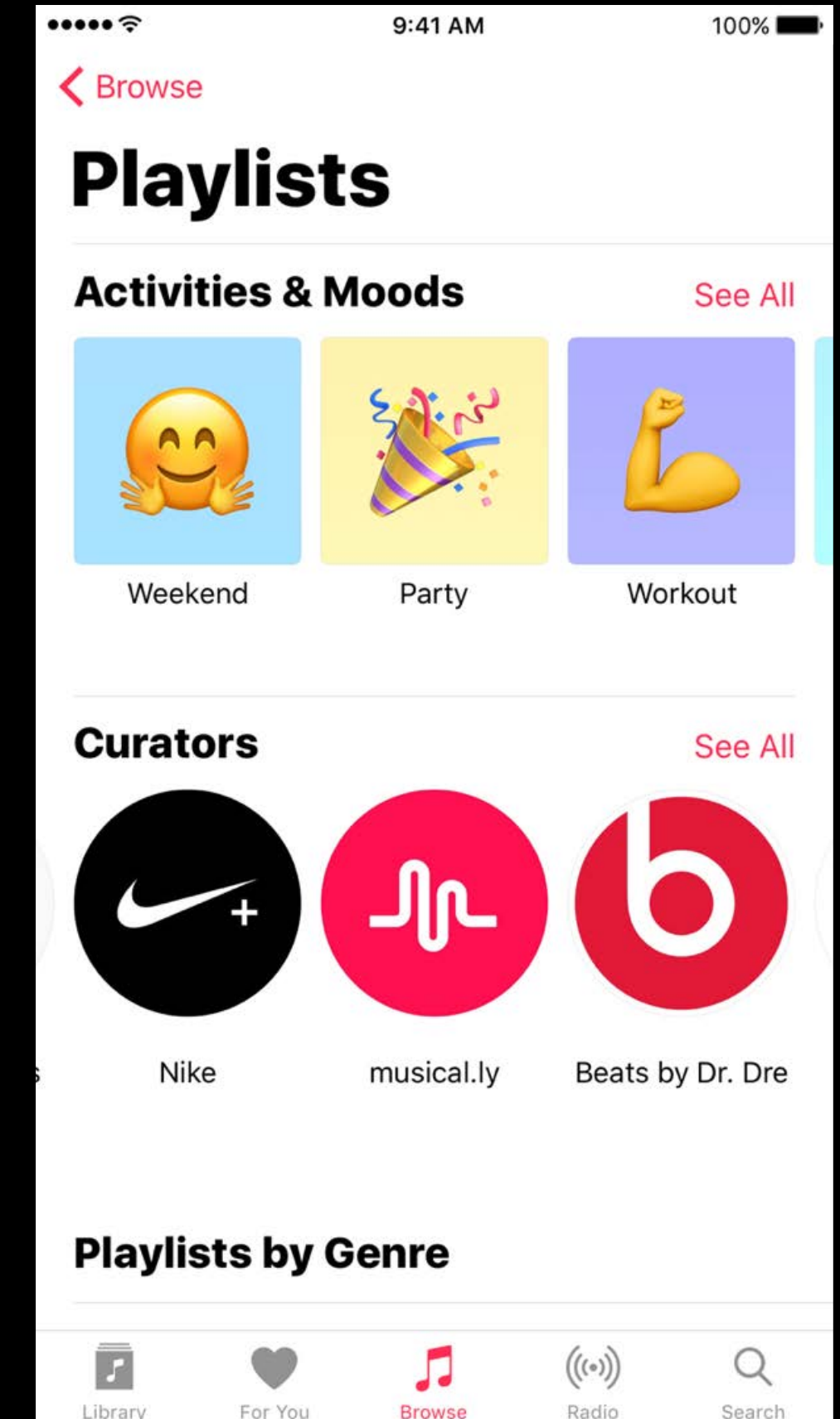
Albums



Playlists



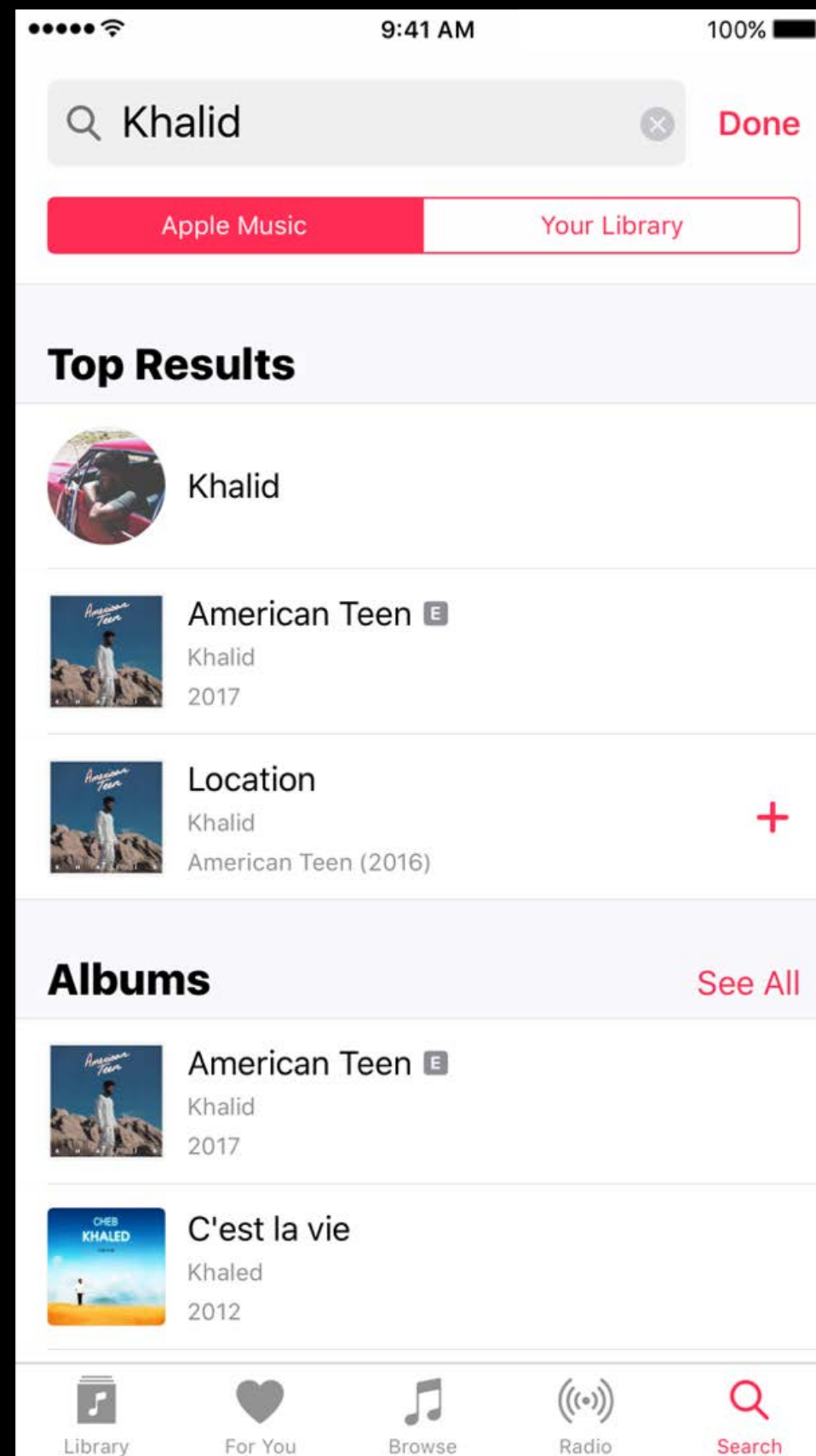
Curators



Apple Music

Discovery

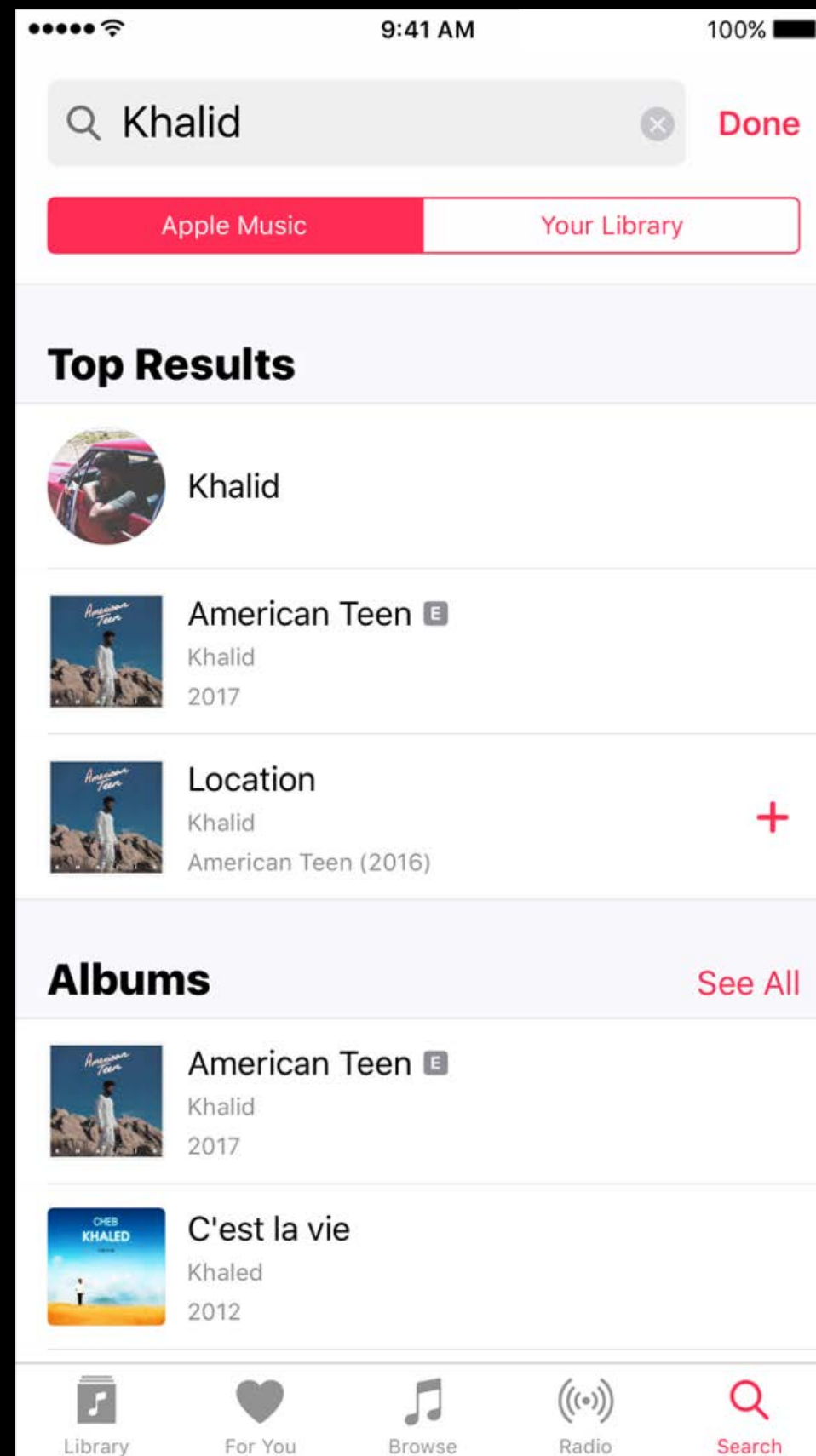
Search



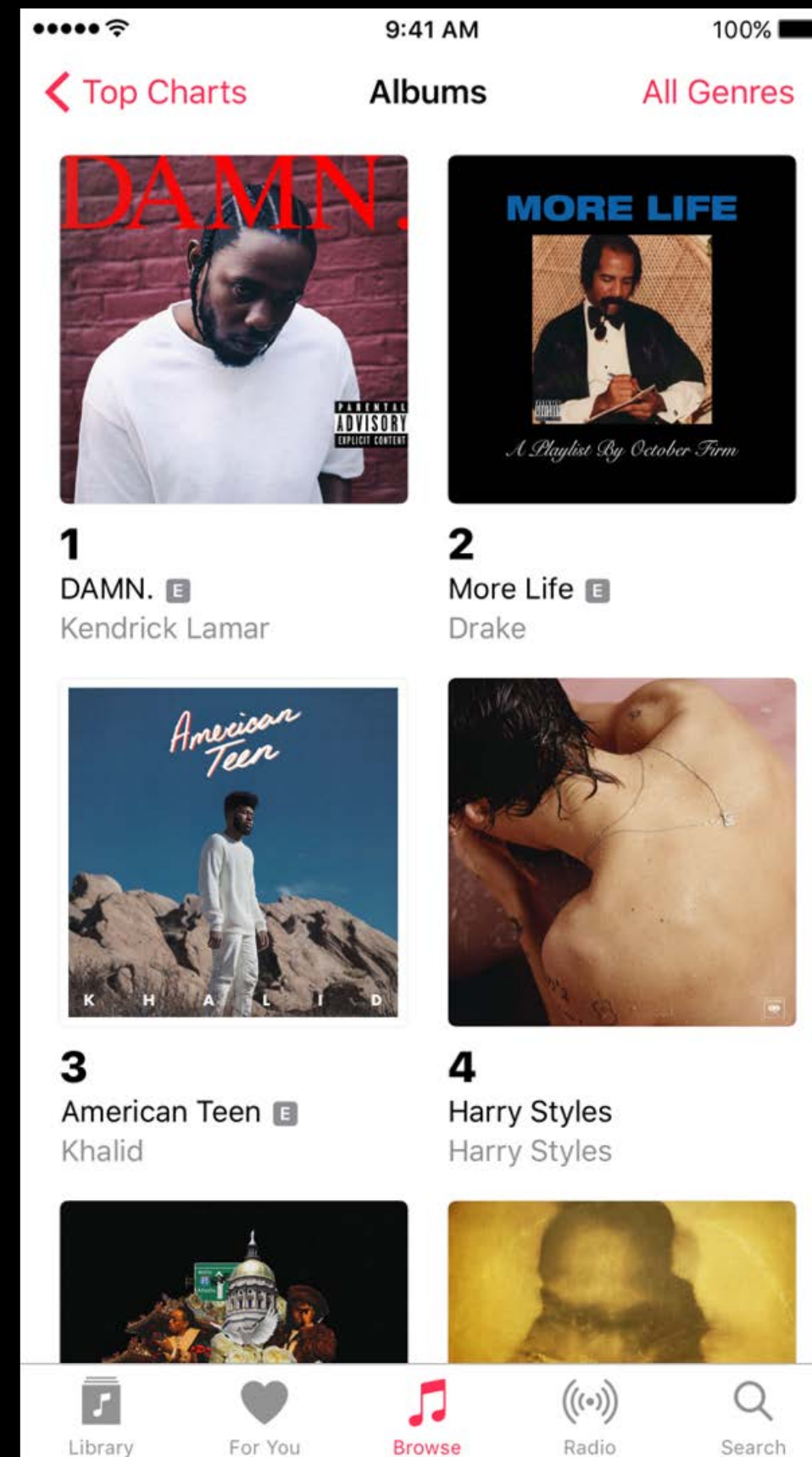
Apple Music

Discovery

Search



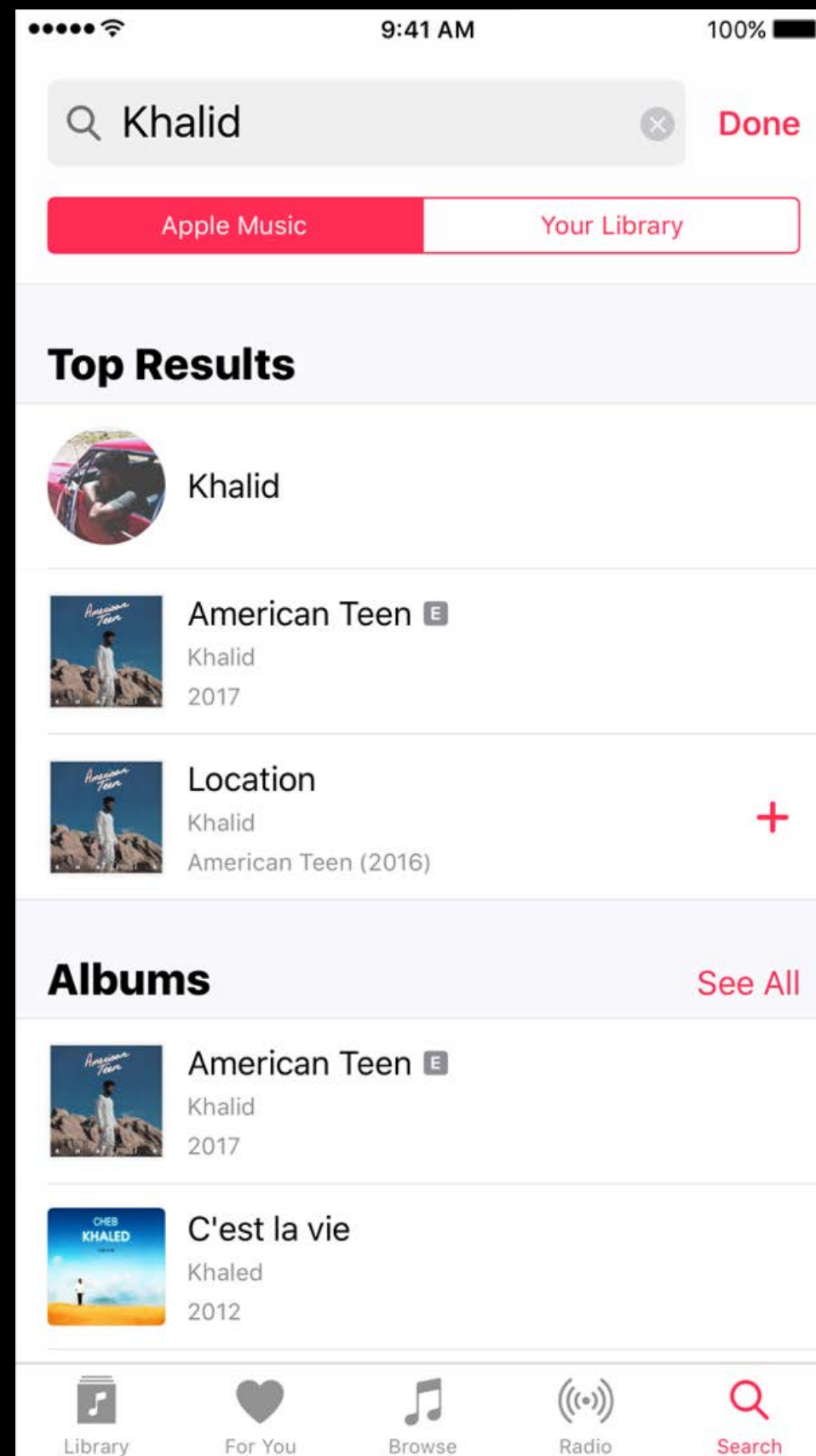
Charts



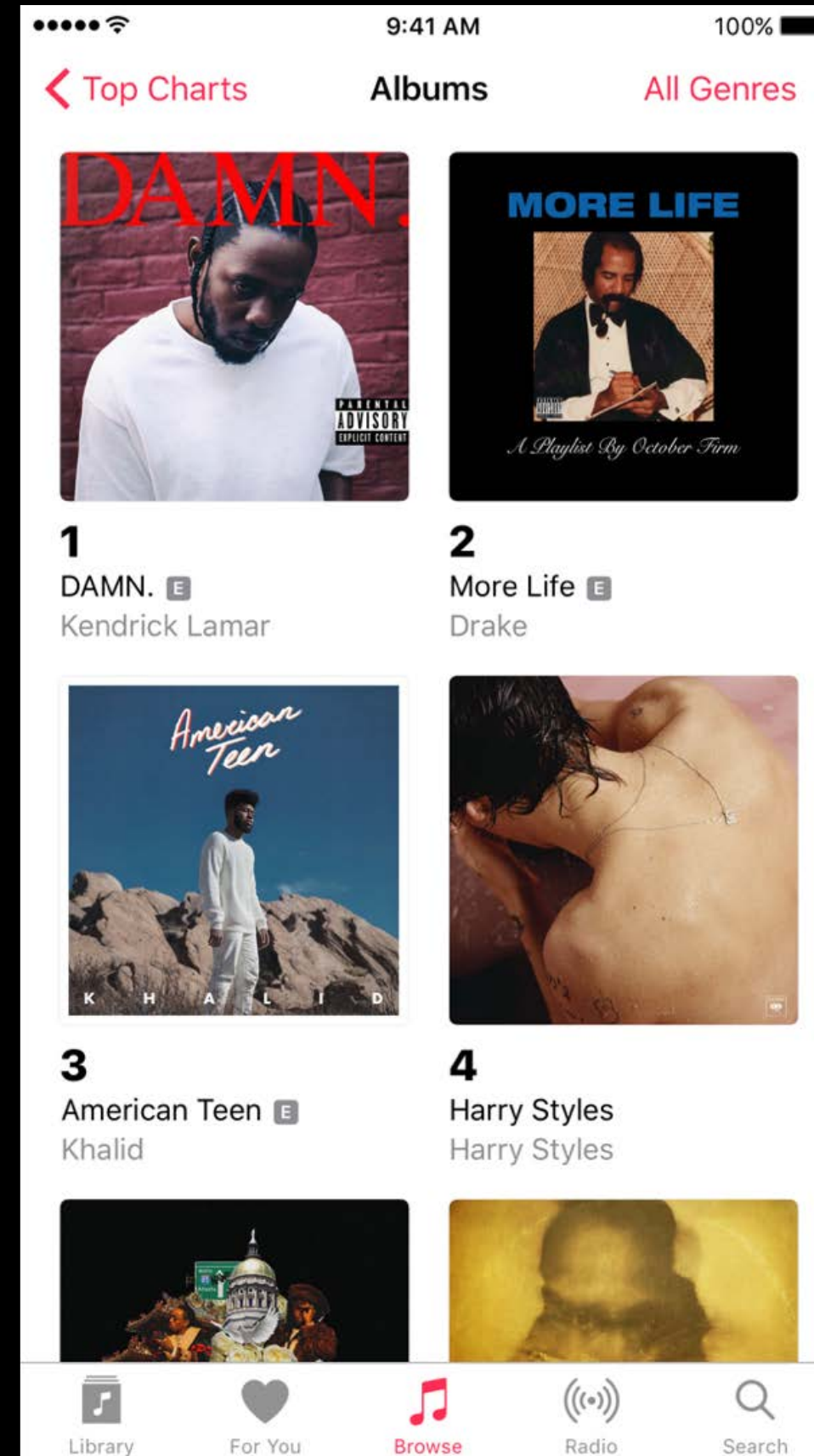
Apple Music

Discovery

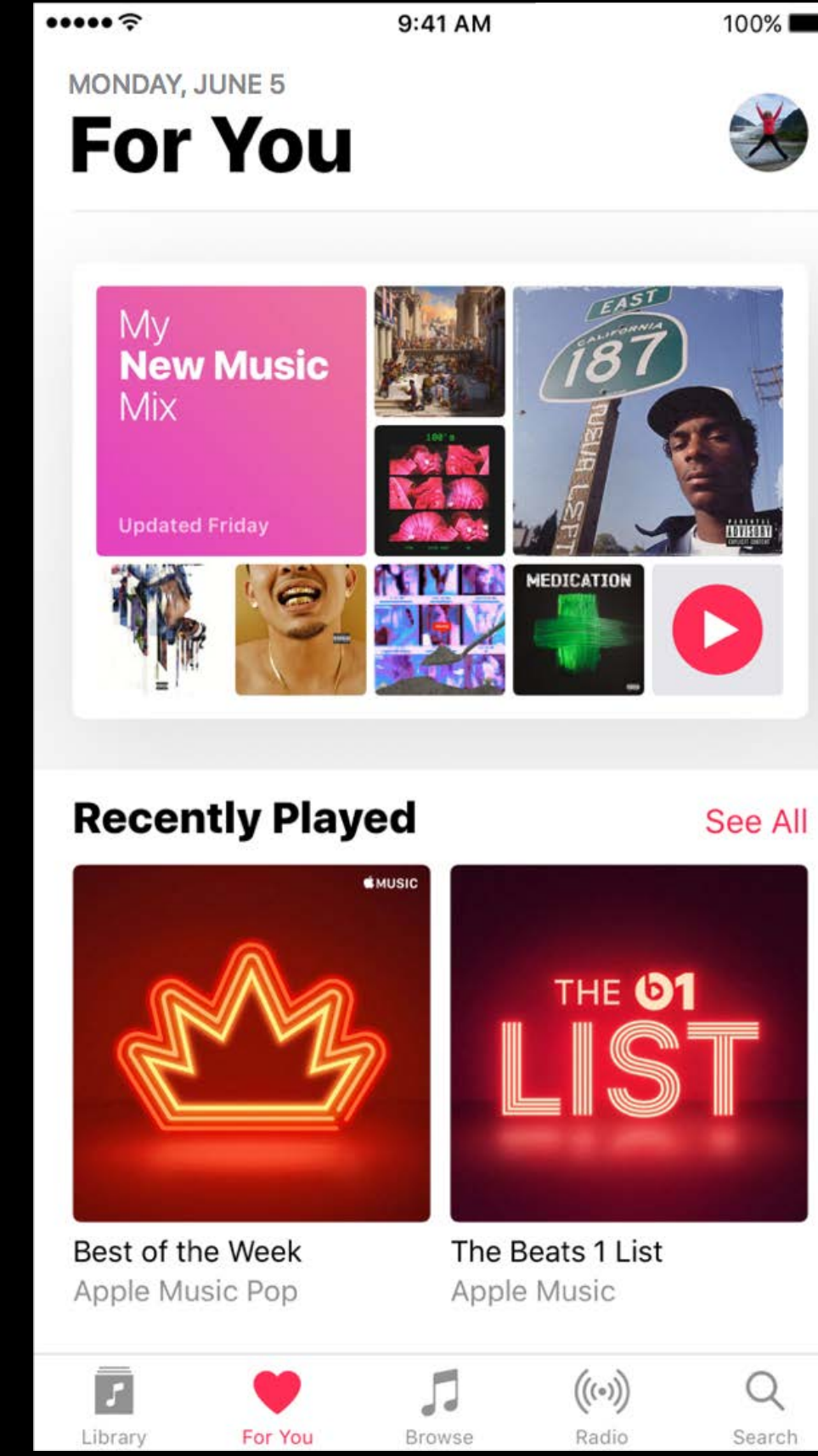
Search



Charts



Recommendations



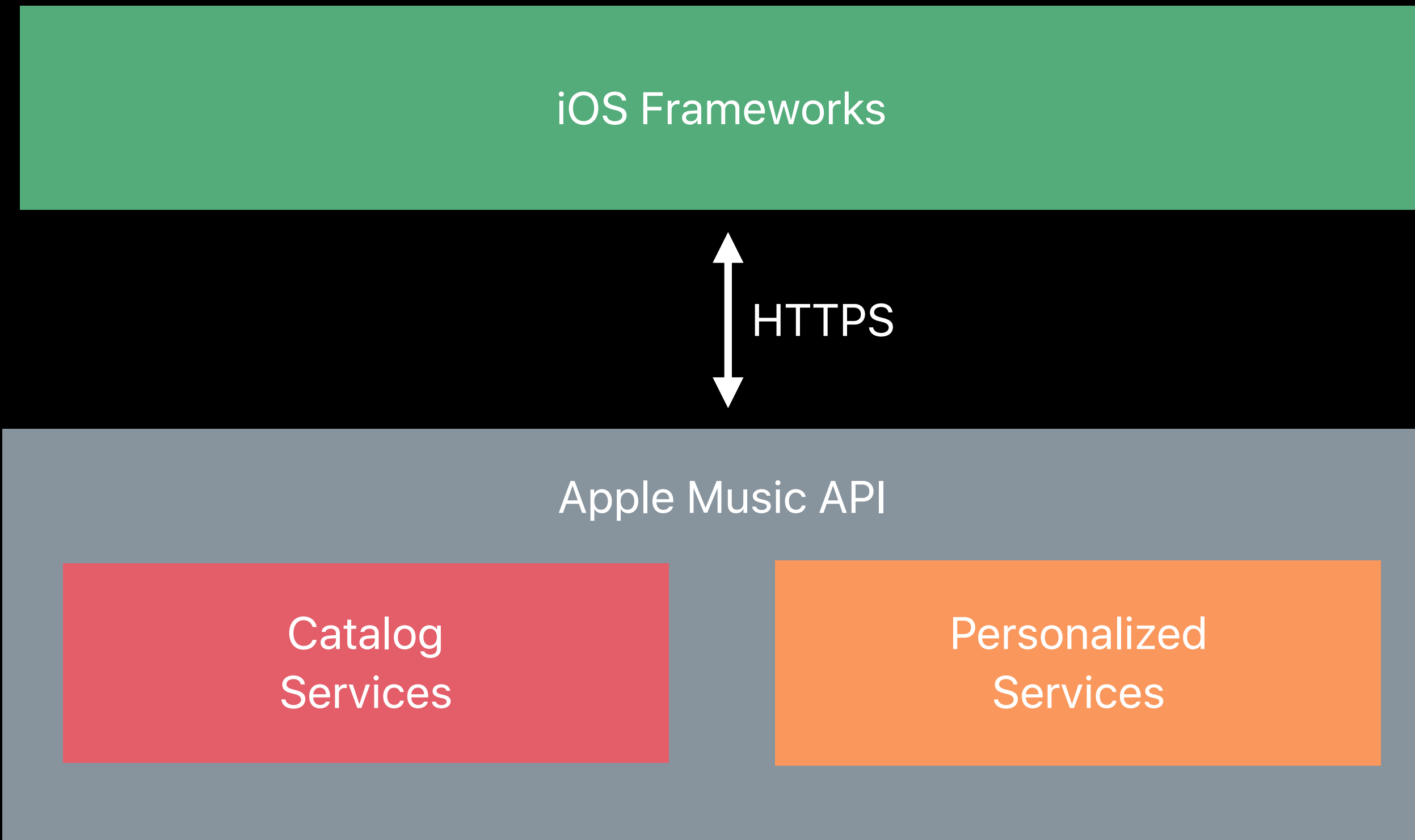
MusicKit



MusicKit

iOS Frameworks

MusicKit



Apple Music API

Tim Parthemore, MusicKit Services

Request format

Resource representations and relationships

Limits and pagination

Demo

Personalized services

API Access

Apple Music API

Catalog

<https://api.music.apple.com>

Apple Music API

Catalog

<https://api.music.apple.com/{version}>

- Versioned

Apple Music API

Catalog

<https://api.music.apple.com/{version}/catalog>

- Versioned

Apple Music API

Catalog

<https://api.music.apple.com/{version}/catalog/{storefront}>

- Versioned
- Storefront scopes the catalog

Apple Music API

Catalog

<https://api.music.apple.com/{version}/catalog/{storefront}/{api}>

- Versioned
- Storefront scopes the catalog

Apple Music API

Catalog

[https://api.music.apple.com/{version}/catalog/{storefront}/{api}?\[params\]](https://api.music.apple.com/{version}/catalog/{storefront}/{api}?[params])

- Versioned
- Storefront scopes the catalog

Apple Music API

Catalog

[https://api.music.apple.com/{version}/catalog/{storefront}/{api}?\[params\]](https://api.music.apple.com/{version}/catalog/{storefront}/{api}?[params])

- Versioned
- Storefront scopes the catalog
- Localized

Apple Music API

Catalog

[https://api.music.apple.com/{version}/catalog/{storefront}/{api}?\[params\]](https://api.music.apple.com/{version}/catalog/{storefront}/{api}?[params])

- Versioned
- Storefront scopes the catalog
- Localized
- Limits and pagination

Apple Music API

Catalog

[https://api.music.apple.com/{version}/catalog/{storefront}/{api}?\[params\]](https://api.music.apple.com/{version}/catalog/{storefront}/{api}?[params])

- Versioned
- Storefront scopes the catalog
- Localized
- Limits and pagination
- Requires a developer token

Apple Music API

Resource representations

id

type

href

```
{"data": [{  
  "id": "1200868601",  
  "type": "albums",  
  "href": "/v1/catalog/us/albums/1200868601"}]}
```

Apple Music API

Resource representations

id
type
href
attributes

```
{
  "data": [
    {
      "id": "1200868601",
      "type": "albums",
      "href": "/v1/catalog/us/albums/1200868601",
      "attributes": {
        "artwork": {...},
        "artistName": "Khalid",
        "discNumber": 1,
        ...
        "playParams": {...},
        "releaseDate": "2017-03-03",
        "trackNumber": 15
      }
    }
  ]
}
```


Apple Music API

Resource representations

id

type

href

attributes

relationships

```
{
  "data": [
    {
      "id": "1200868601",
      "type": "albums",
      "href": "/v1/catalog/us/albums/1200868601",
      "attributes": {
        ...
      },
      "relationships": {
        "tracks": {
          "data": [{...}, ..., {...}],
          "href": "/v1/.../albums/1200868601/tracks"
        },
        "artists": {
          "data": [{...}],
          "href": "/v1/.../albums/1200868601/artists"
        }
      }
    }
  ]
}
```

Apple Music API

Resource relationships

Apple Music API

Resource relationships

Album Resource

id

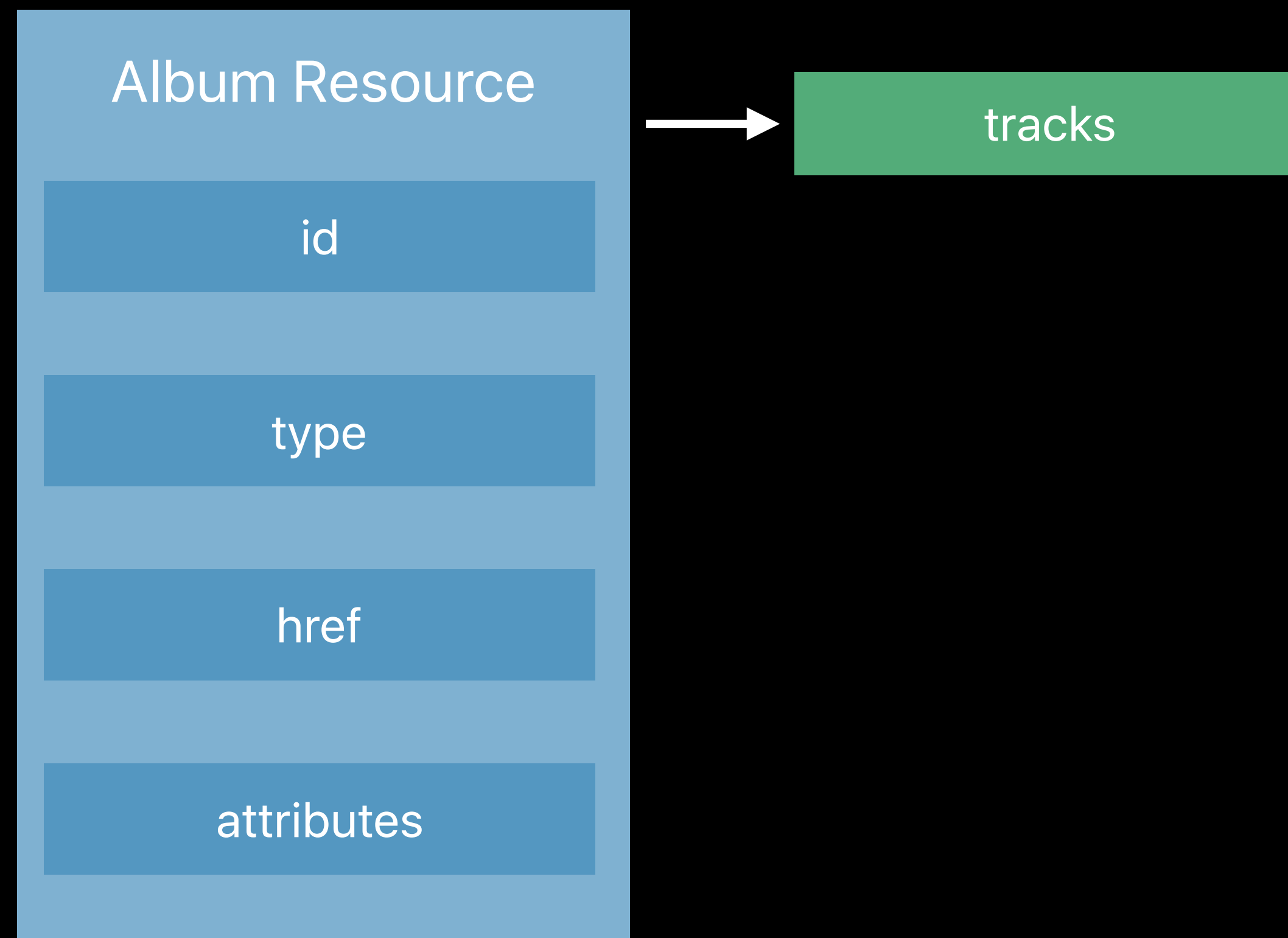
type

href

attributes

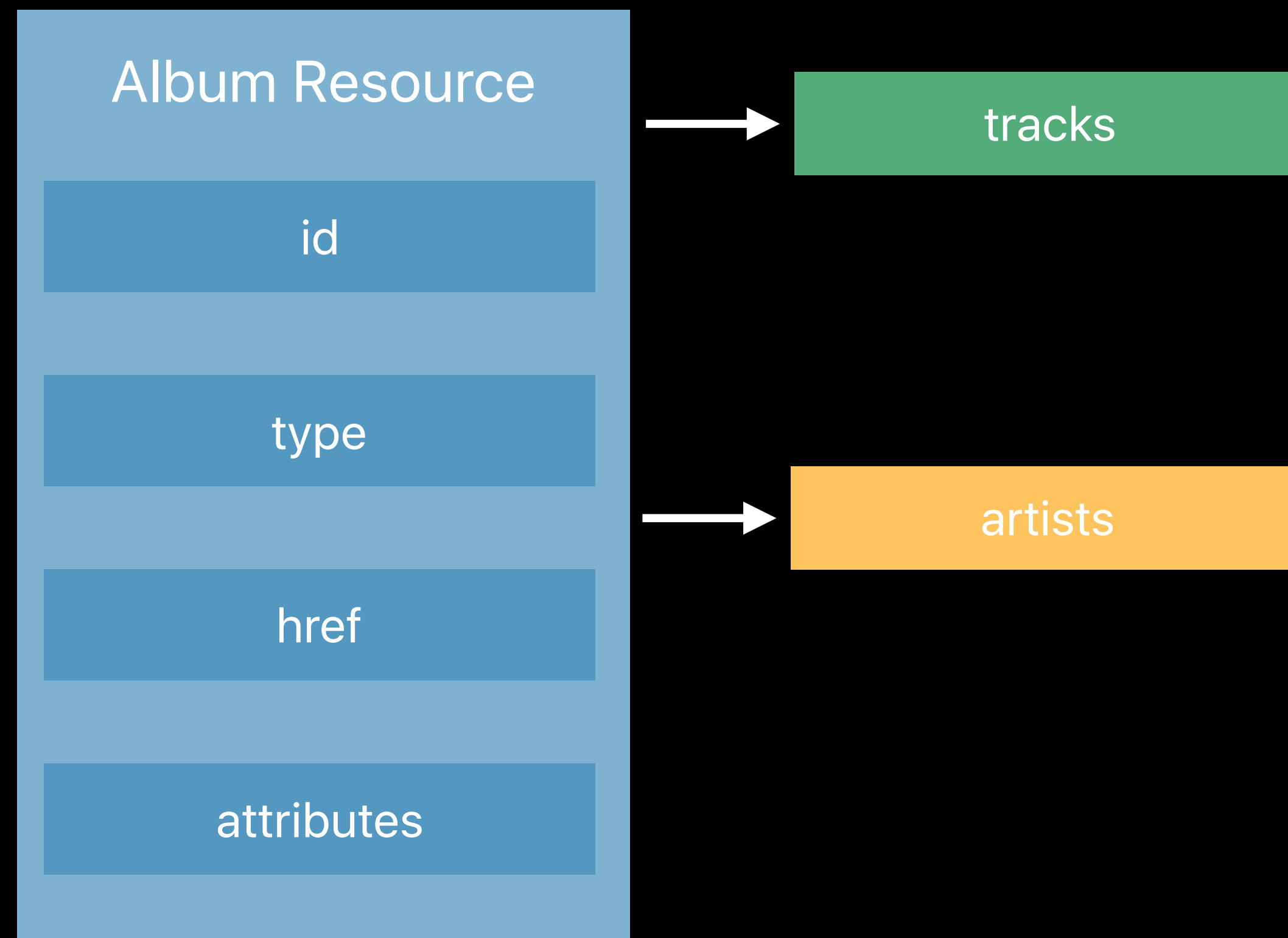
Apple Music API

Resource relationships



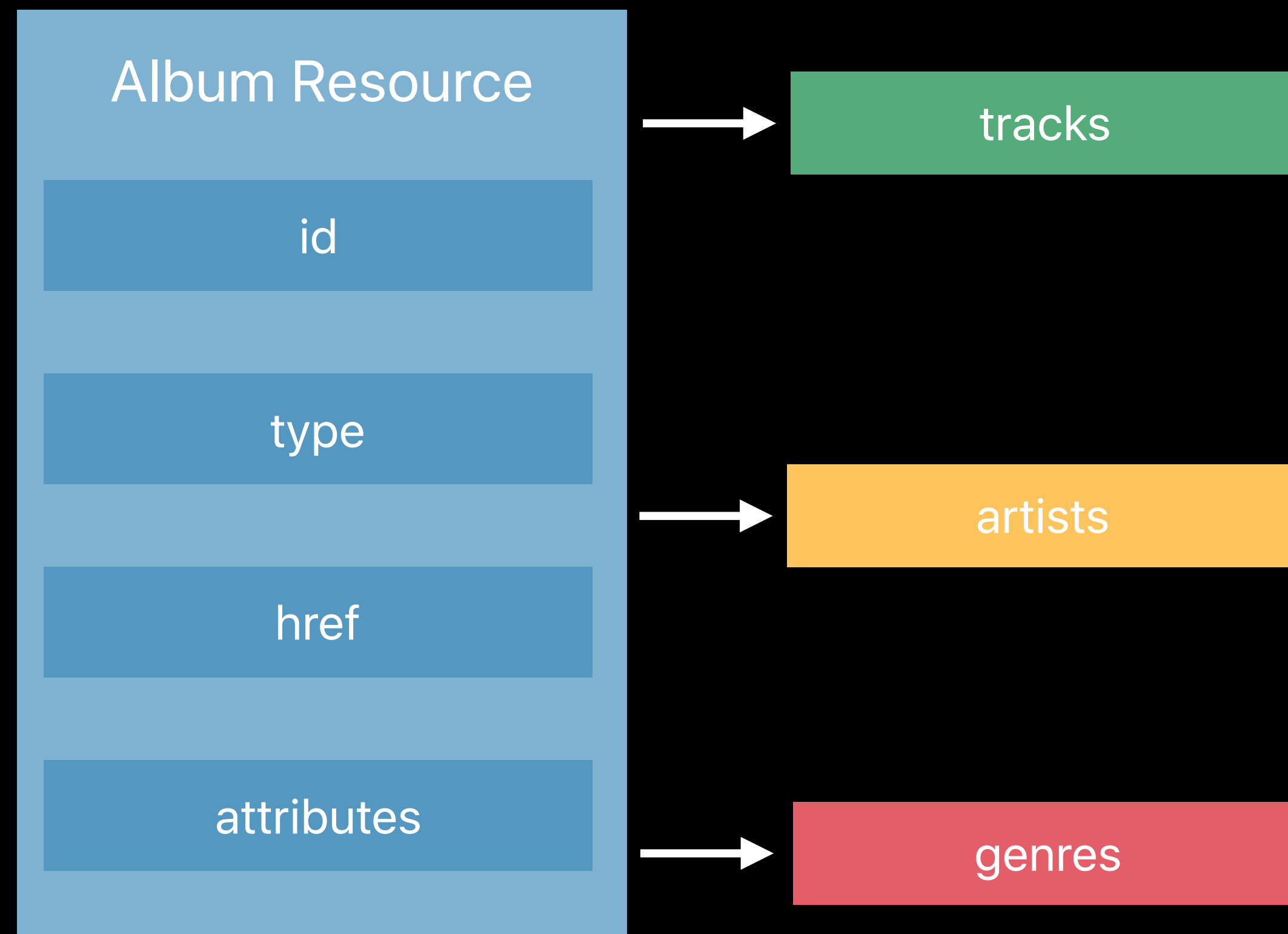
Apple Music API

Resource relationships



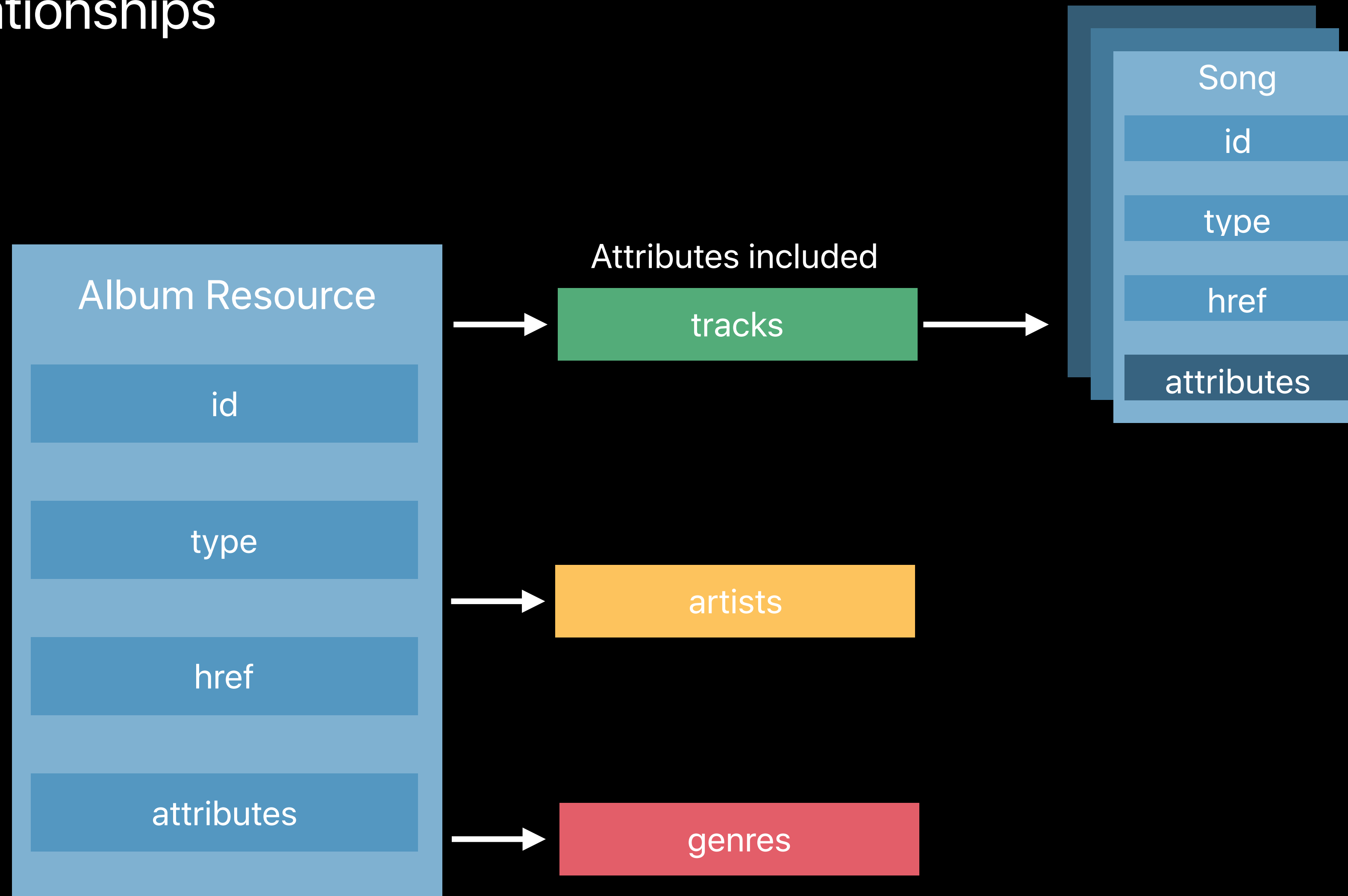
Apple Music API

Resource relationships



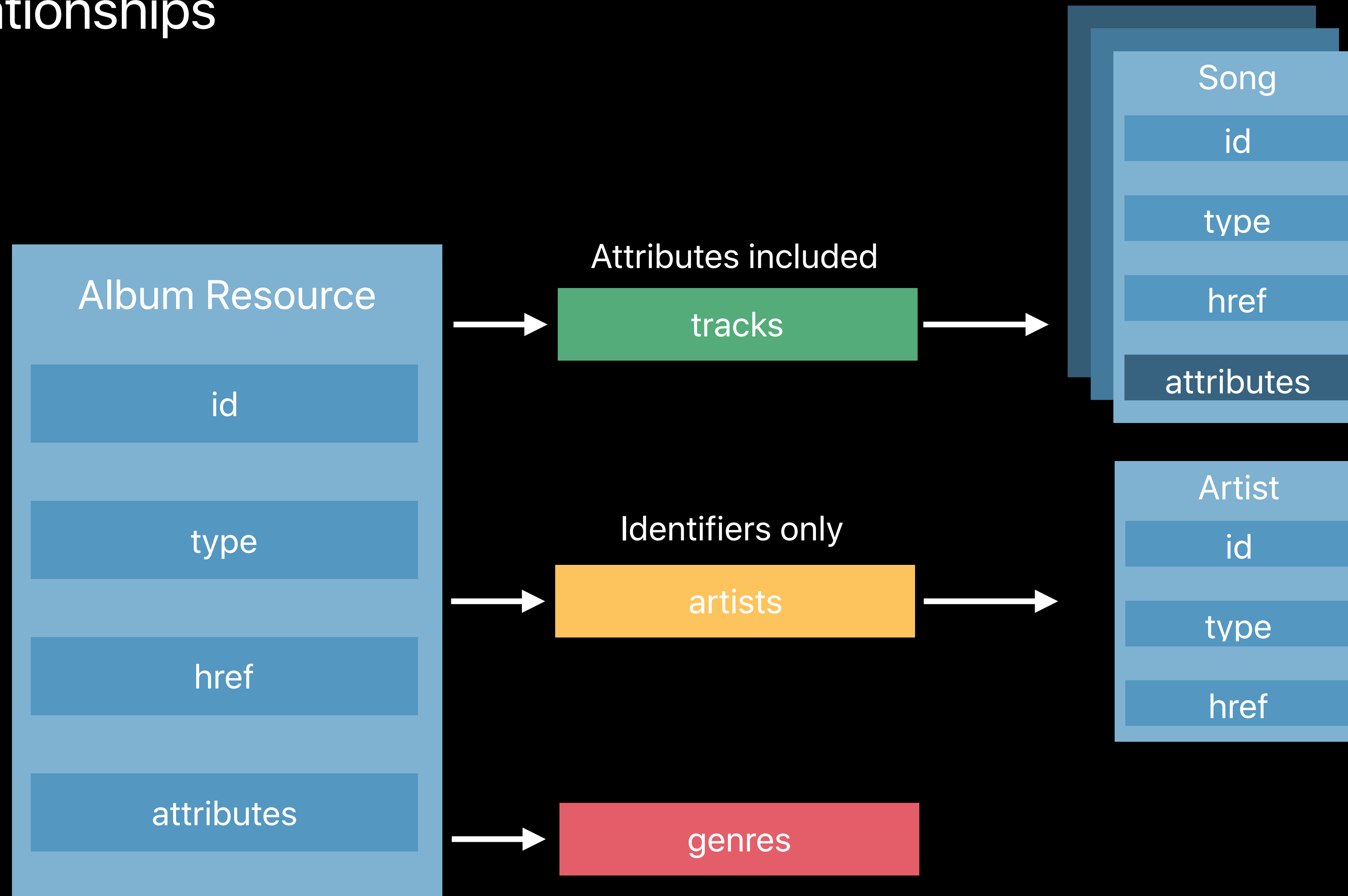
Apple Music API

Resource relationships



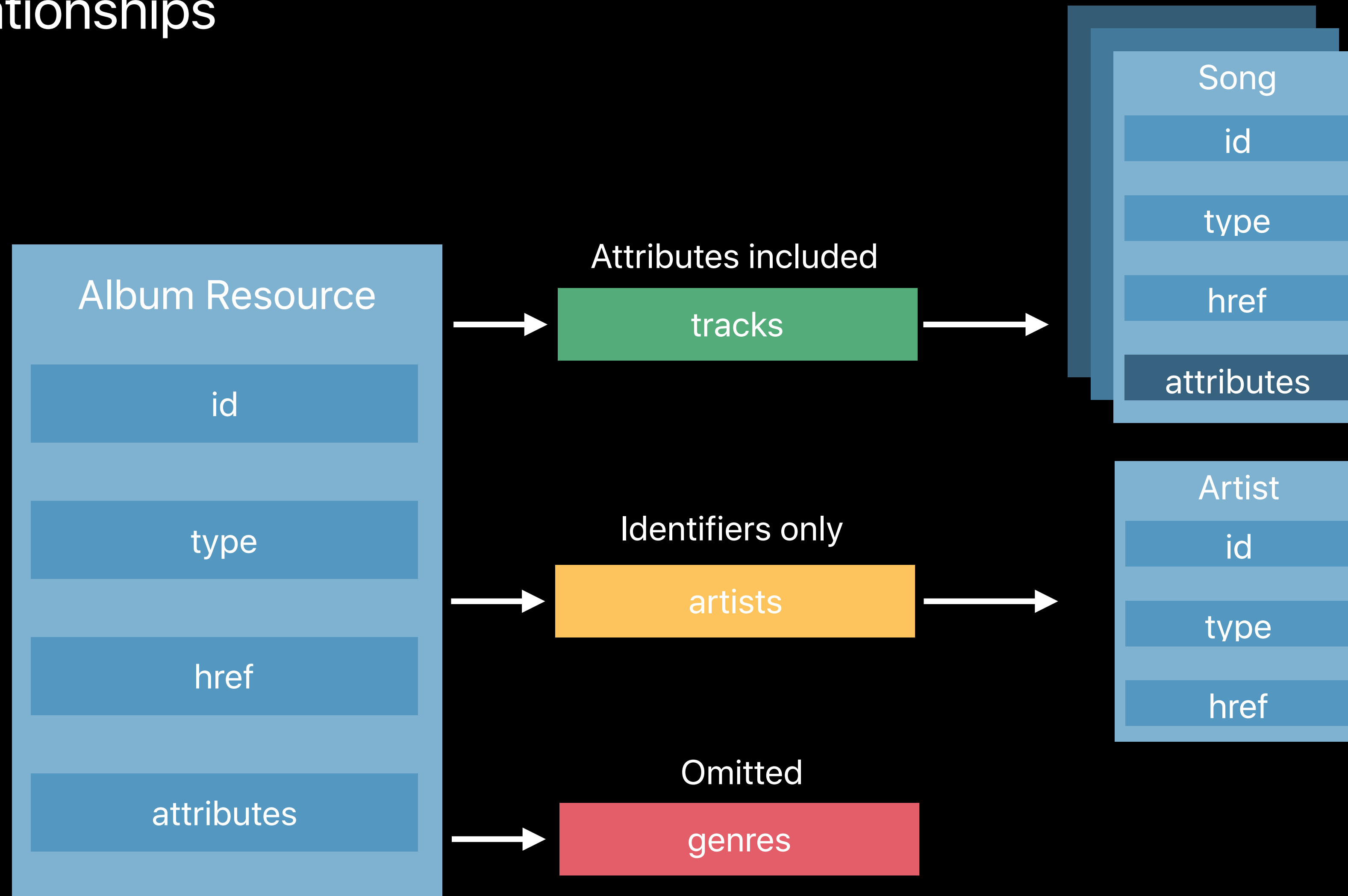
Apple Music API

Resource relationships



Apple Music API

Resource relationships



<https://api.music.apple.com/v1/catalog/us/albums/1200868601>

```
"relationships": {
  "tracks": {"data": [{"id": "548392190",
    "type": "songs",
    "href": "/v1/catalog/us/songs/548392190",
    "attributes": {...}},
    {"id": "548392191",
    "type": "songs",
    "href": "/v1/catalog/us/songs/548392191",
    "attributes": {...}}],
    "href": "/v1/catalog/us/albums/1200868601/tracks"},
  "artists": {"data": [{"id": "82842423",
    "type": "artists",
    "href": "/v1/catalog/us/artists/82842423"}],
    "href": "/v1/catalog/us/albums/1200868601/artists"}}
```

<https://api.music.apple.com/v1/catalog/us/albums/1200868601>

```
"relationships": {
  "tracks": {"data": [{"id": "548392190",
    "type": "songs",
    "href": "/v1/catalog/us/songs/548392190",
    "attributes": {...}},
    {"id": "548392191",
    "type": "songs",
    "href": "/v1/catalog/us/songs/548392191",
    "attributes": {...}}],
    "href": "/v1/catalog/us/albums/1200868601/tracks"},
  "artists": {"data": [{"id": "82842423",
    "type": "artists",
    "href": "/v1/catalog/us/artists/82842423"}],
    "href": "/v1/catalog/us/albums/1200868601/artists"}}
```

<https://api.music.apple.com/v1/catalog/us/albums/1200868601>

```
"relationships": {
  "tracks": {"data": [{"id": "548392190",
    "type": "songs",
    "href": "/v1/catalog/us/songs/548392190",
    "attributes": {...}},
    {"id": "548392191",
    "type": "songs",
    "href": "/v1/catalog/us/songs/548392191",
    "attributes": {...}}],
    "href": "/v1/catalog/us/albums/1200868601/tracks"},
  "artists": {"data": [{"id": "82842423",
    "type": "artists",
    "href": "/v1/catalog/us/artists/82842423"}],
    "href": "/v1/catalog/us/albums/1200868601/artists"}}
```

```
https://api.music.apple.com/v1/catalog/us/albums/1200868601?include=artists
```

```
"relationships": {
  "tracks": {...},
  "artists": {"data": [{
    "id": "1200868601",
    "type": "artists",
    "href": "/v1/catalog/us/artists/82842423",
    "attributes": {
      "genreNames": ["R&B/Soul", "Hip-Hop/Rap", "Rap"]
      "isGroup": false,
      "name": "Khalid",
      ...
    }
  ]},
  "href": "/v1/catalog/us/albums/1200868601/artists"}}
```

```
https://api.music.apple.com/v1/catalog/us/albums/1200868601?include=artists
```

```
"relationships": {  
  "tracks": {...},  
  "artists": {"data": [{  
    "id": "1200868601",  
    "type": "artists",  
    "href": "/v1/catalog/us/artists/82842423",  
    "attributes": {  
      "genreNames": ["R&B/Soul", "Hip-Hop/Rap", "Rap"]  
      "isGroup": false,  
      "name": "Khalid",  
      ...  
    }  
  }],  
  "href": "/v1/catalog/us/albums/1200868601/artists"}}
```

```
https://api.music.apple.com/v1/catalog/us/albums/1200868601?include=artists,genres
```

```
"relationships": {  
  "tracks": {...},  
  "artists": {...}  
  "genres": {"data": [{"id": "14",  
                    "type": "genres",  
                    "href": "/v1/catalog/us/genres/14",  
                    "attributes": {  
                      "name": "Pop"}}},  
  {"id": "34",  
   "type": "genres",  
   "href": "/v1/catalog/us/genres/34",  
   "attributes": {  
     "name": "Music"}],  
  "href": "/v1/catalog/us/albums/1200868601/genres"  
}}
```

```
https://api.music.apple.com/v1/catalog/us/albums/1200868601?include=artists,genres
```

```
"relationships": {  
  "tracks": {...},  
  "artists": {...}  
  "genres": {"data": [{"id": "14",  
    "type": "genres",  
    "href": "/v1/catalog/us/genres/14",  
    "attributes": {  
      "name": "Pop"}}],  
    {"id": "34",  
      "type": "genres",  
      "href": "/v1/catalog/us/genres/34",  
      "attributes": {  
        "name": "Music"}]},  
    "href": "/v1/catalog/us/albums/1200868601/genres"  
  }}  
}}
```


Apple Music API

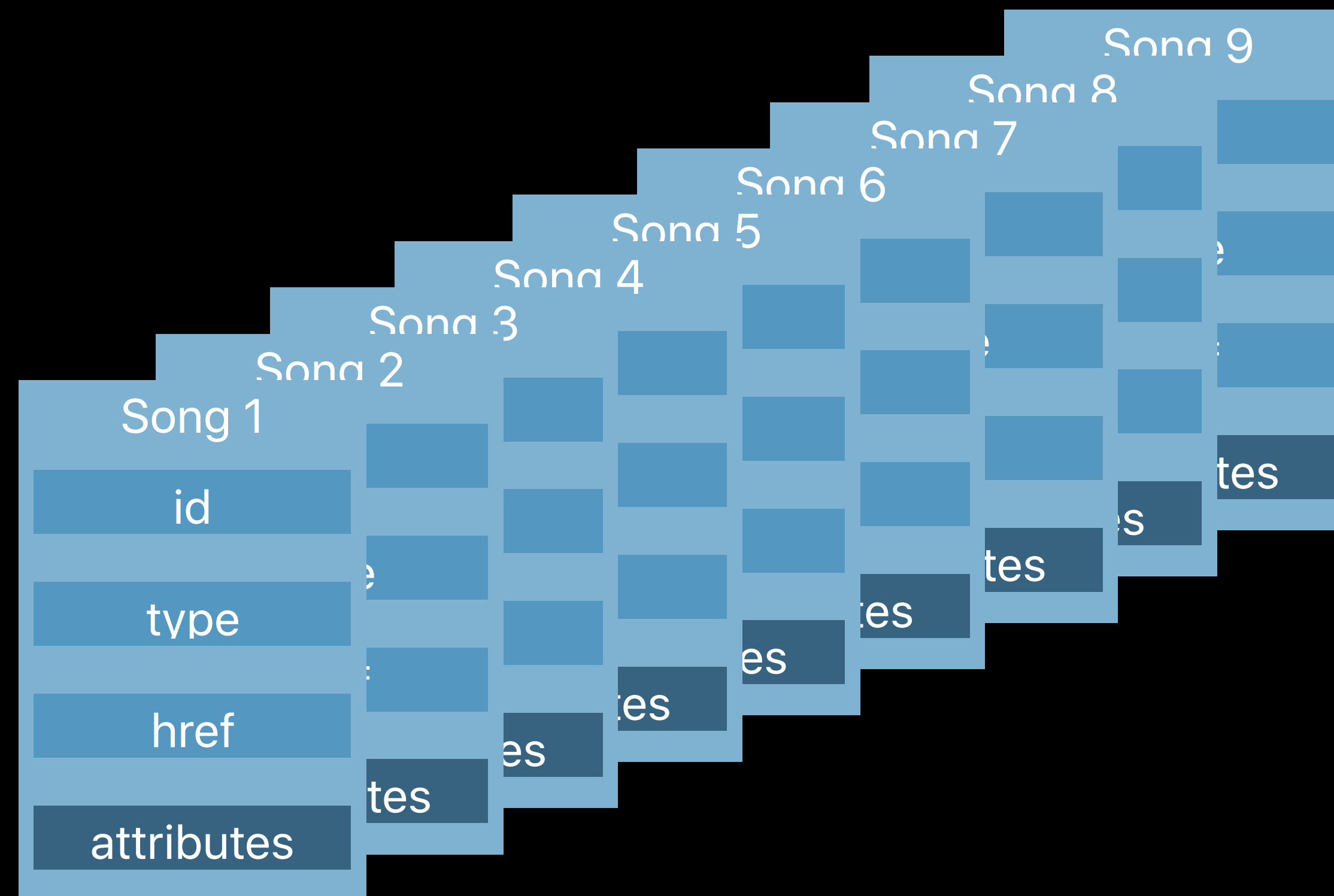
Limits and pagination

<https://api.music.apple.com/v1/catalog/us/albums/1200868601/tracks>

Apple Music API

Limits and pagination

<https://api.music.apple.com/v1/catalog/us/albums/1200868601/tracks>



Apple Music API

Limits and pagination

<https://api.music.apple.com/v1/catalog/us/albums/1200868601/tracks>

Apple Music API

Limits and pagination

<https://api.music.apple.com/v1/catalog/us/albums/1200868601/tracks?limit=2>

Apple Music API

Limits and pagination

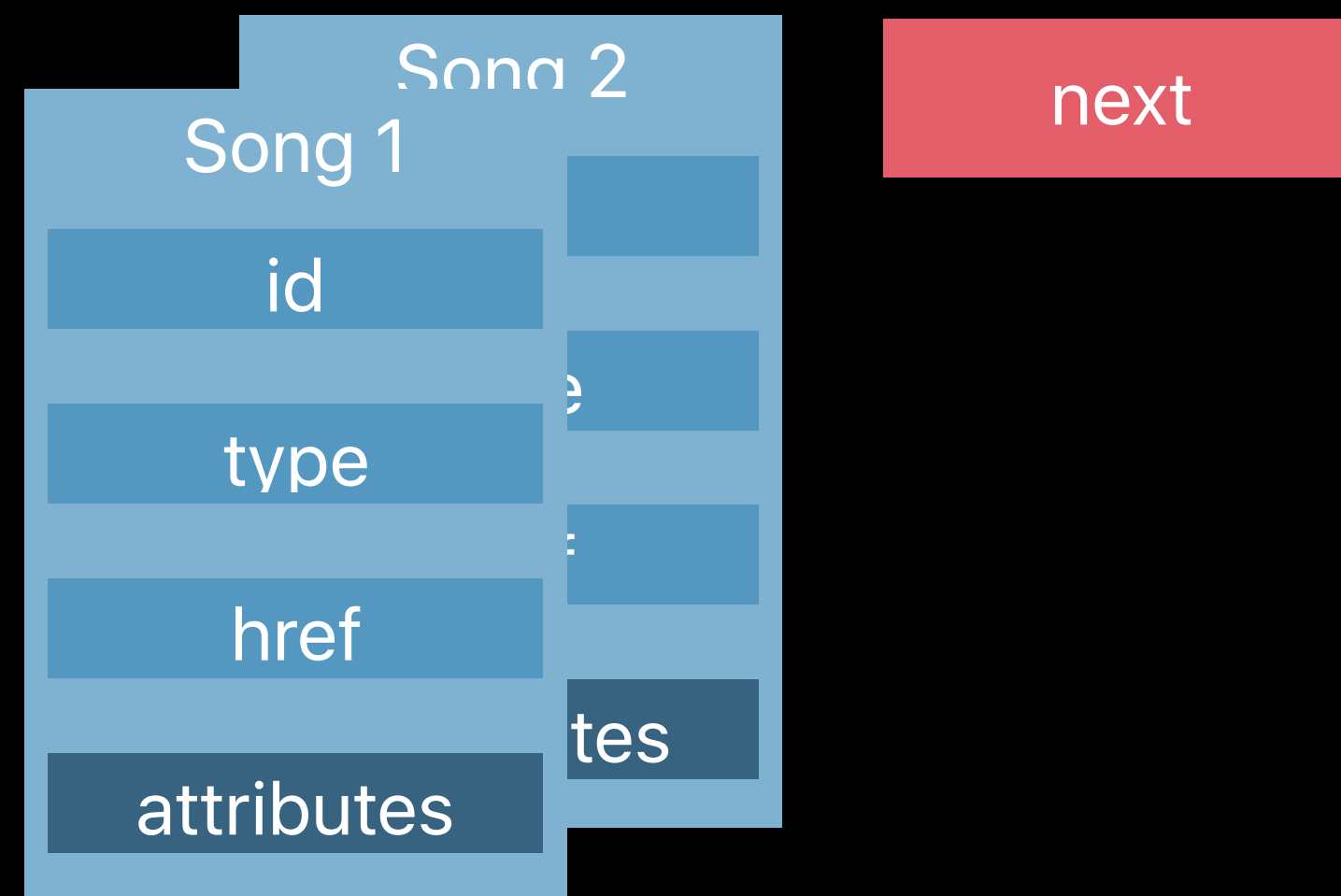
<https://api.music.apple.com/v1/catalog/us/albums/1200868601/tracks?limit=2>



Apple Music API

Limits and pagination

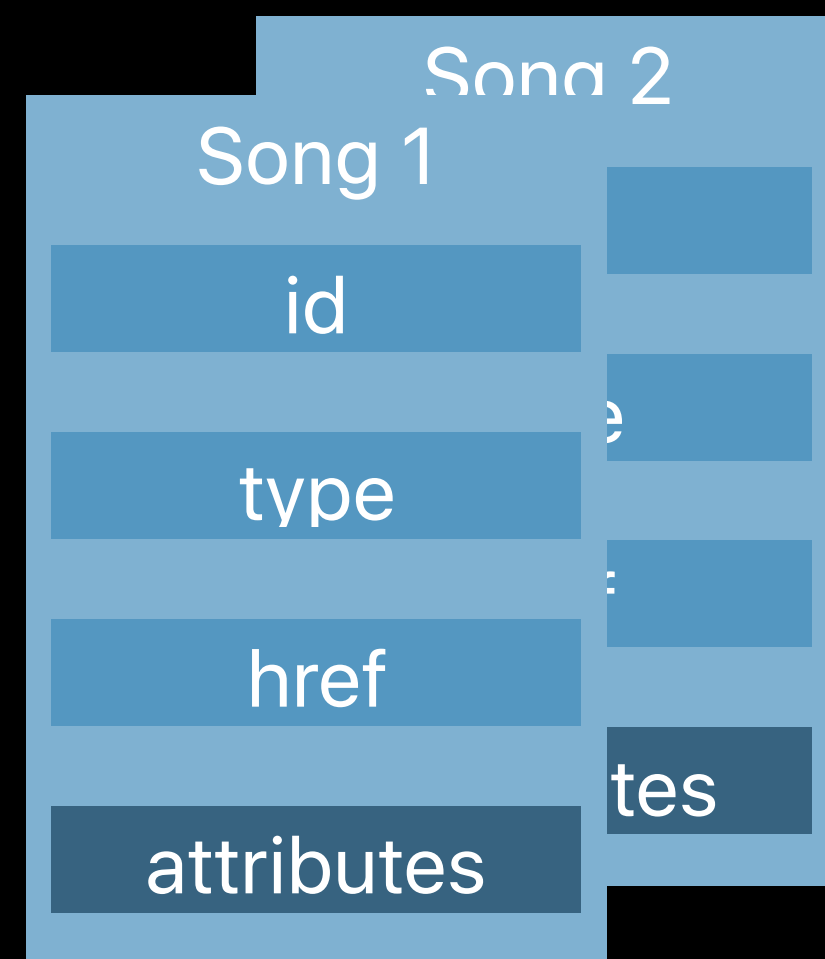
<https://api.music.apple.com/v1/catalog/us/albums/1200868601/tracks?limit=2>



Apple Music API

Limits and pagination

<https://api.music.apple.com/v1/catalog/us/albums/1200868601/tracks?limit=2>



next

</v1/catalog/us/albums/1200868601/tracks?offset=2>

Apple Music API

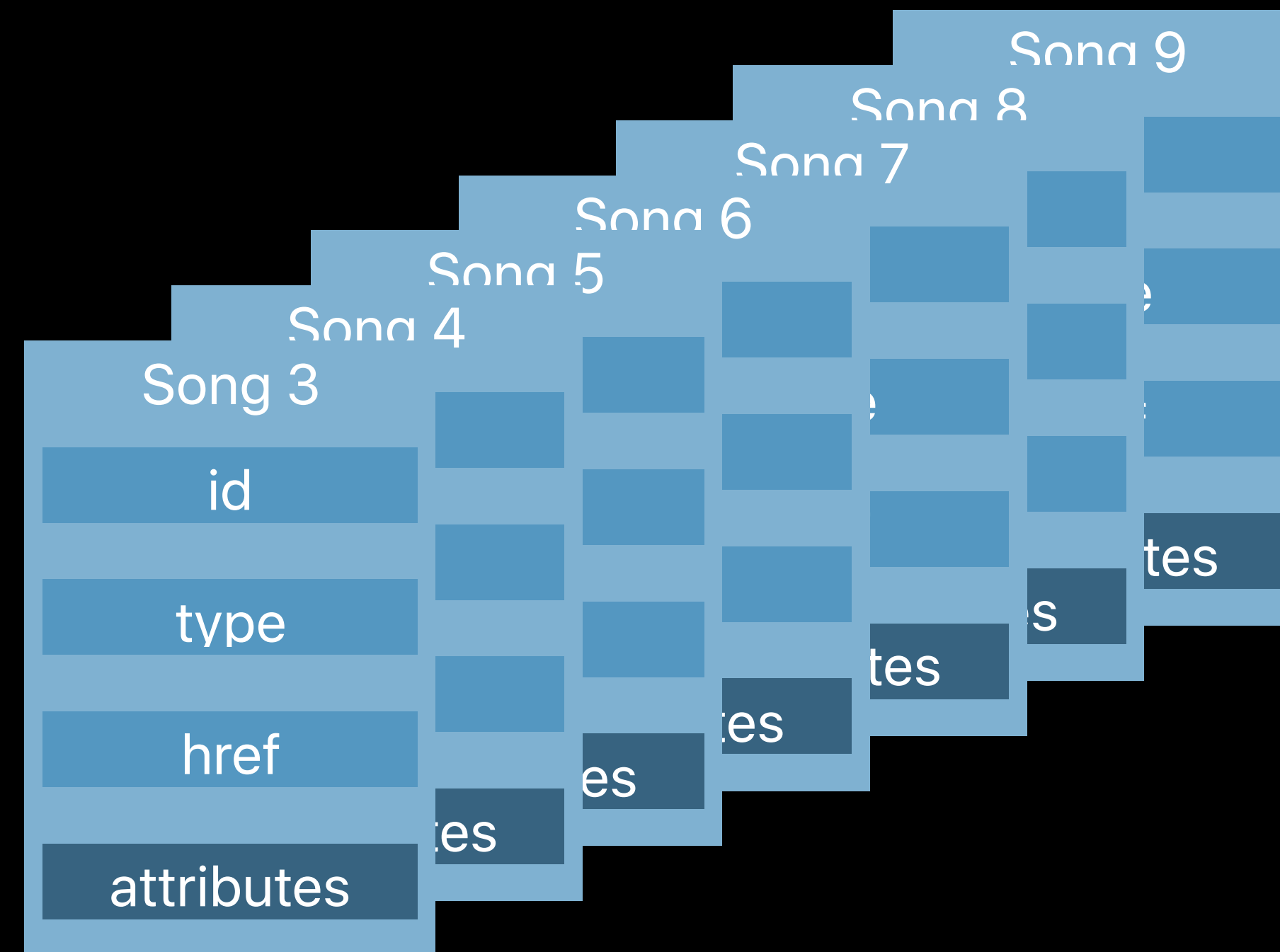
Limits and pagination

<https://api.music.apple.com/v1/catalog/us/albums/1200868601/tracks?offset=2>

Apple Music API

Limits and pagination

<https://api.music.apple.com/v1/catalog/us/albums/1200868601/tracks?offset=2>



Apple Music API

Limits and pagination

<https://api.music.apple.com/v1/catalog/us/albums/1200868601/tracks?offset=2>

Apple Music API

Limits and pagination

<https://api.music.apple.com/v1/catalog/us/albums/1200868601/tracks?offset=2&limit=2>

Apple Music API

Limits and pagination

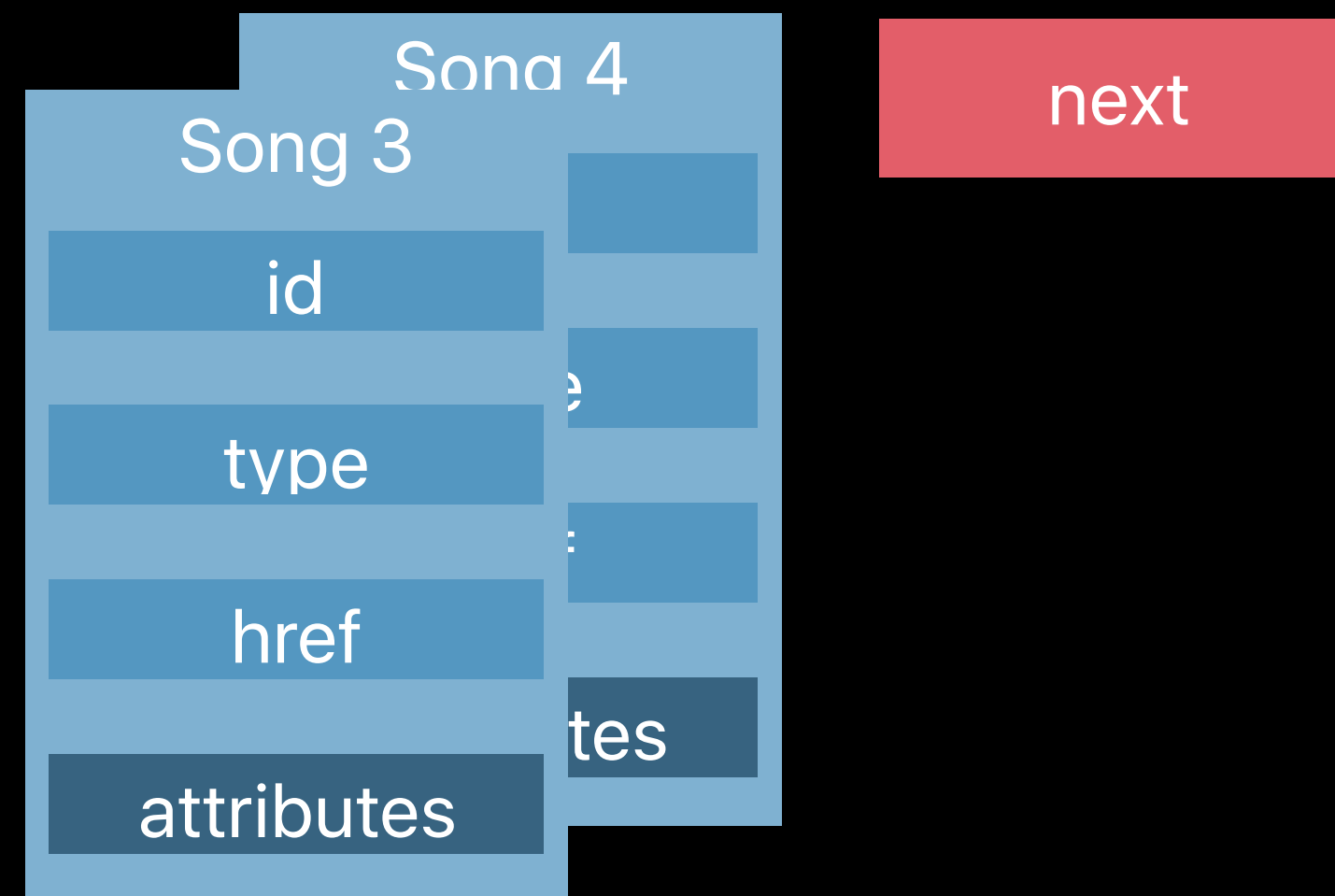
<https://api.music.apple.com/v1/catalog/us/albums/1200868601/tracks?offset=2&limit=2>



Apple Music API

Limits and pagination

<https://api.music.apple.com/v1/catalog/us/albums/1200868601/tracks?offset=2&limit=2>



Apple Music API

Limits and pagination

<https://api.music.apple.com/v1/catalog/us/albums/1200868601/tracks?offset=2&limit=2>



next

</v1/catalog/us/albums/1200868601/tracks?offset=4>

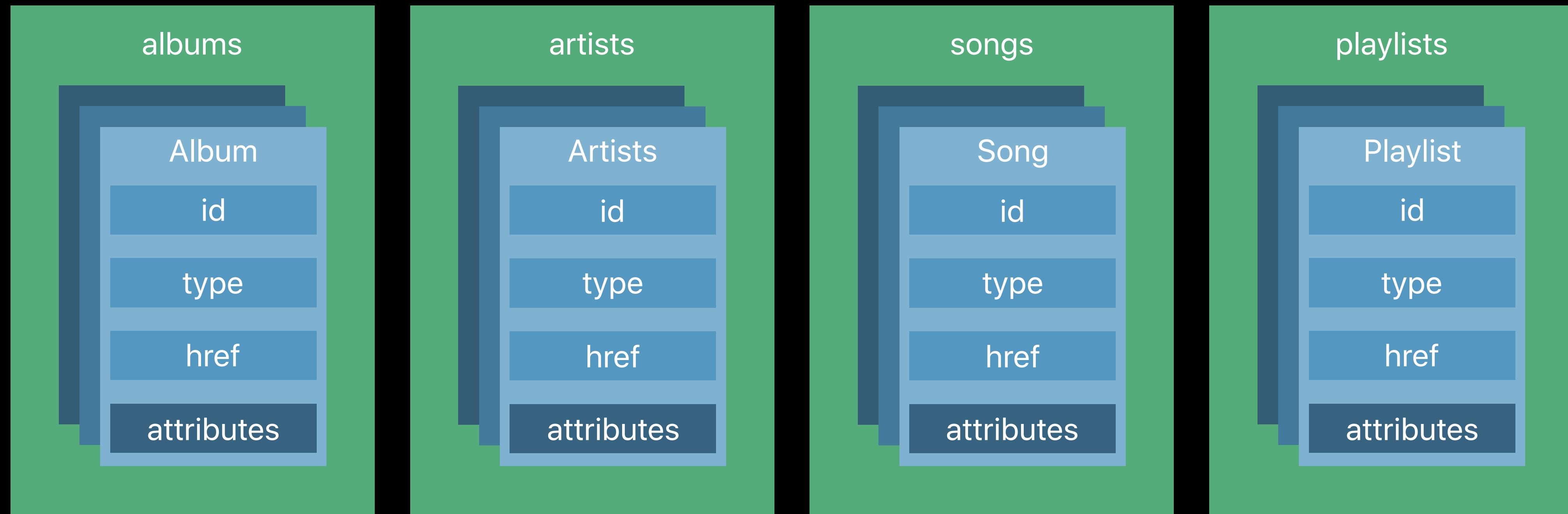
<https://api.music.apple.com/v1/catalog/us/albums/1200868601/tracks?limit=2>

```
{
  "data": [
    {
      "attributes": {...},
      "href": "/v1/catalog/us/songs/1200868616",
      "id": "1200868616",
      "relationships": {...},
      "type": "songs"
    },
    {
      "attributes": {...},
      "href": "/v1/catalog/us/songs/1200868874",
      "id": "1200868874",
      "relationships": {...},
      "type": "songs"
    }
  ],
  "next": "/v1/catalog/us/albums/1200868601/tracks?offset=2"
}
```

Apple Music API

Search

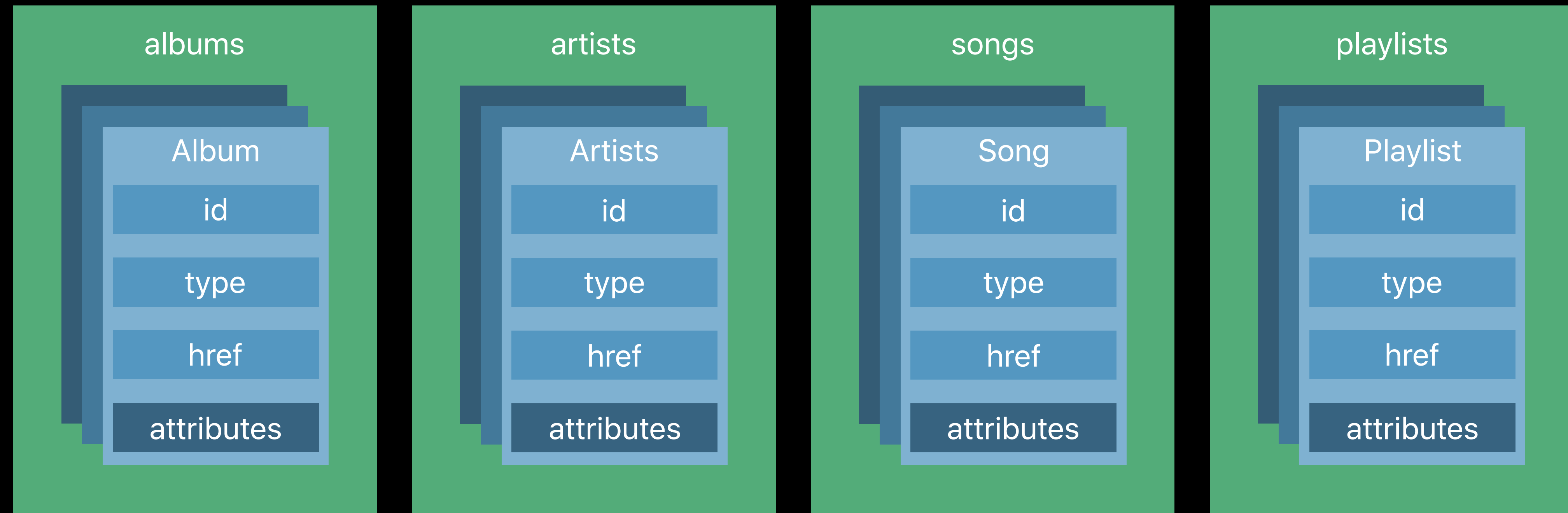
<https://api.music.apple.com/v1/catalog/us/search?term=khalid>



Apple Music API

Search

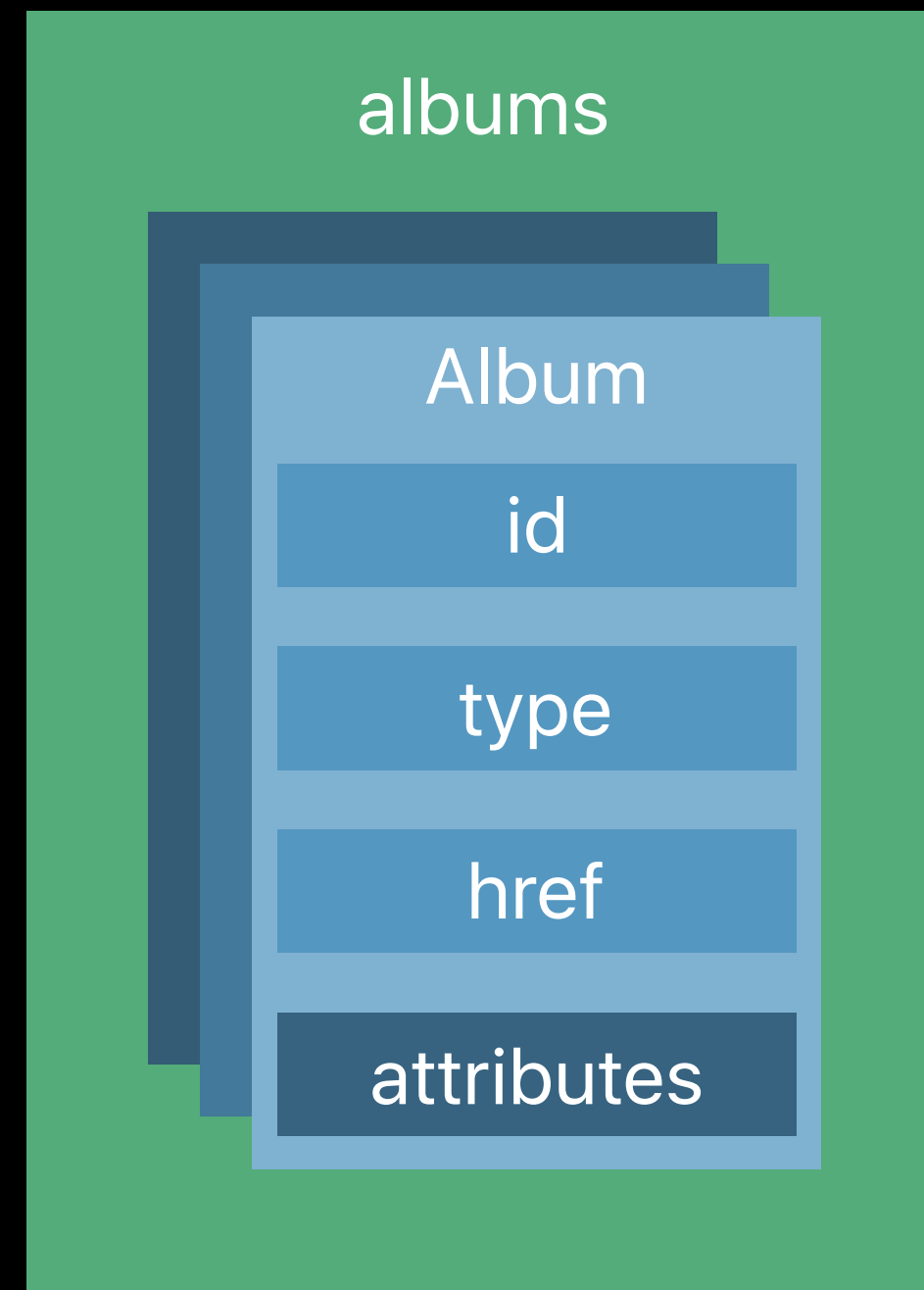
<https://api.music.apple.com/v1/catalog/us/search?term=khalid&types=albums>



Apple Music API

Search

<https://api.music.apple.com/v1/catalog/us/search?term=khalid&types=albums>



Demo

Apple Music Catalog Search

Joel Lopes Da Silva, iOS Music

Apple Music API

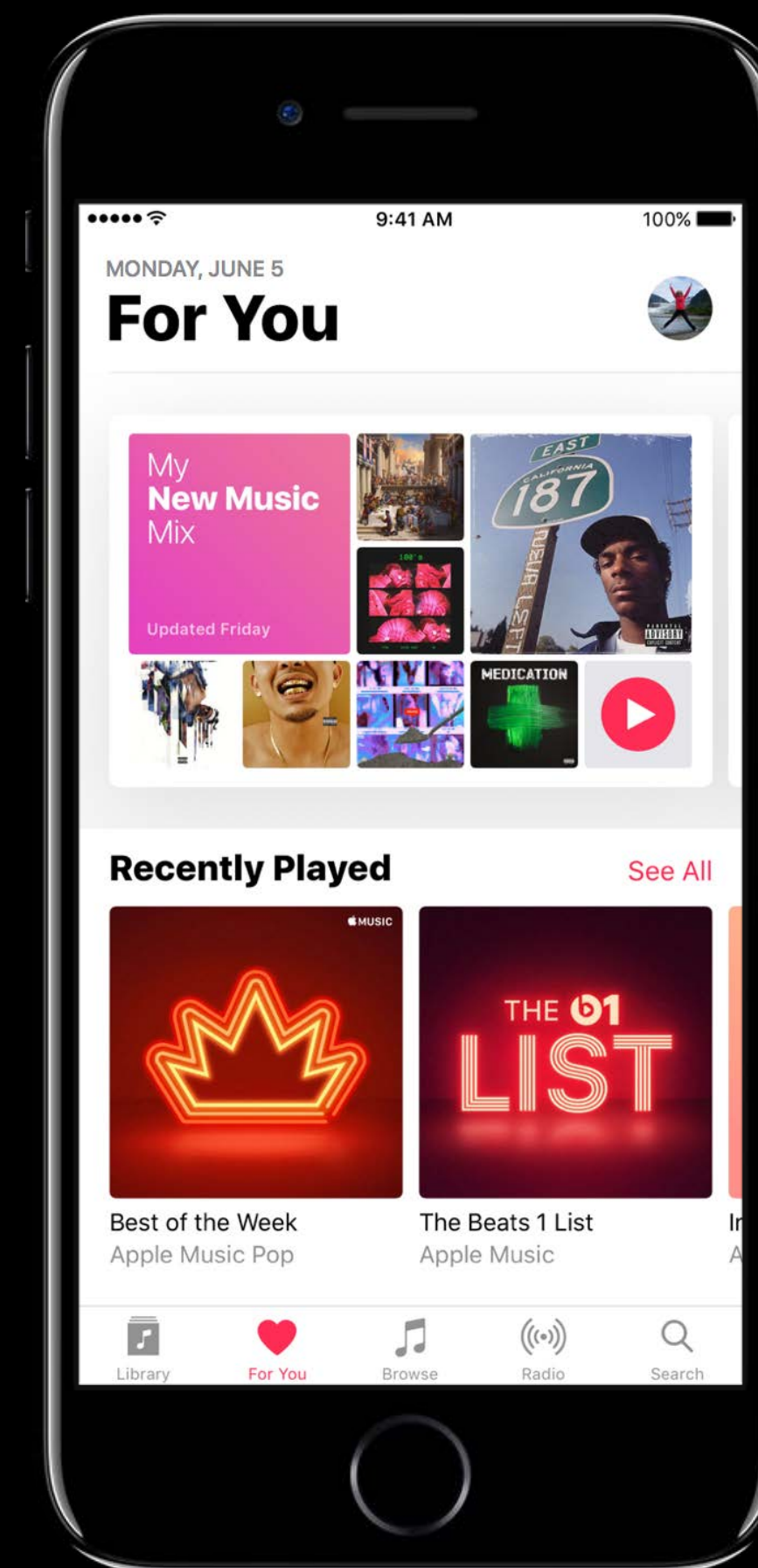
Personalized

<https://api.music.apple.com/v1/me/recent/played>

User must have an Apple Music Subscription

Required tokens

- Developer token
- Music user token



API Access

Requirements

API Access

Requirements

Apple Developer Program membership

API Access

Requirements

Apple Developer Program membership

MusicKit private key

API Access

Requirements

Apple Developer Program membership

MusicKit private key

Developer token

- JSON Web Token (RFC 7519)
- Required in the "Authorization" header for every Apple Music API request

API Access

Requirements

Apple Developer Program membership

MusicKit private key

Developer token

- JSON Web Token (RFC 7519)
- Required in the "Authorization" header for every Apple Music API request

Security

- Protect your MusicKit private key
- Recommend dynamically generating developer tokens

API Access

More about the developer token

API Access

More about the developer token

Required Headers - Encryption Algorithm (alg), Key Identifier (kid)

```
{  
  "alg": "ES256",  
  "kid": "ABC123DEFG"  
}
```

API Access

More about the developer token

Required Headers - Encryption Algorithm (alg), Key Identifier (kid)

Required Claims - Issuer/Team ID (iss), Issued at (iat), Expiration (exp)

```
{
  "alg": "ES256",
  "kid": "ABC123DEFG"
}
{
  "iss": "DEF123GHIJ",
  "iat": 1437179036,
  "exp": 1493298100
}
```

Using MusicKit on iOS

Joel Lopes Da Silva, iOS Music

Get access to Apple Music API

Allow users to subscribe

Play subscription and library content

Get access to Apple Music API

Allow users to subscribe

Play subscription and library content

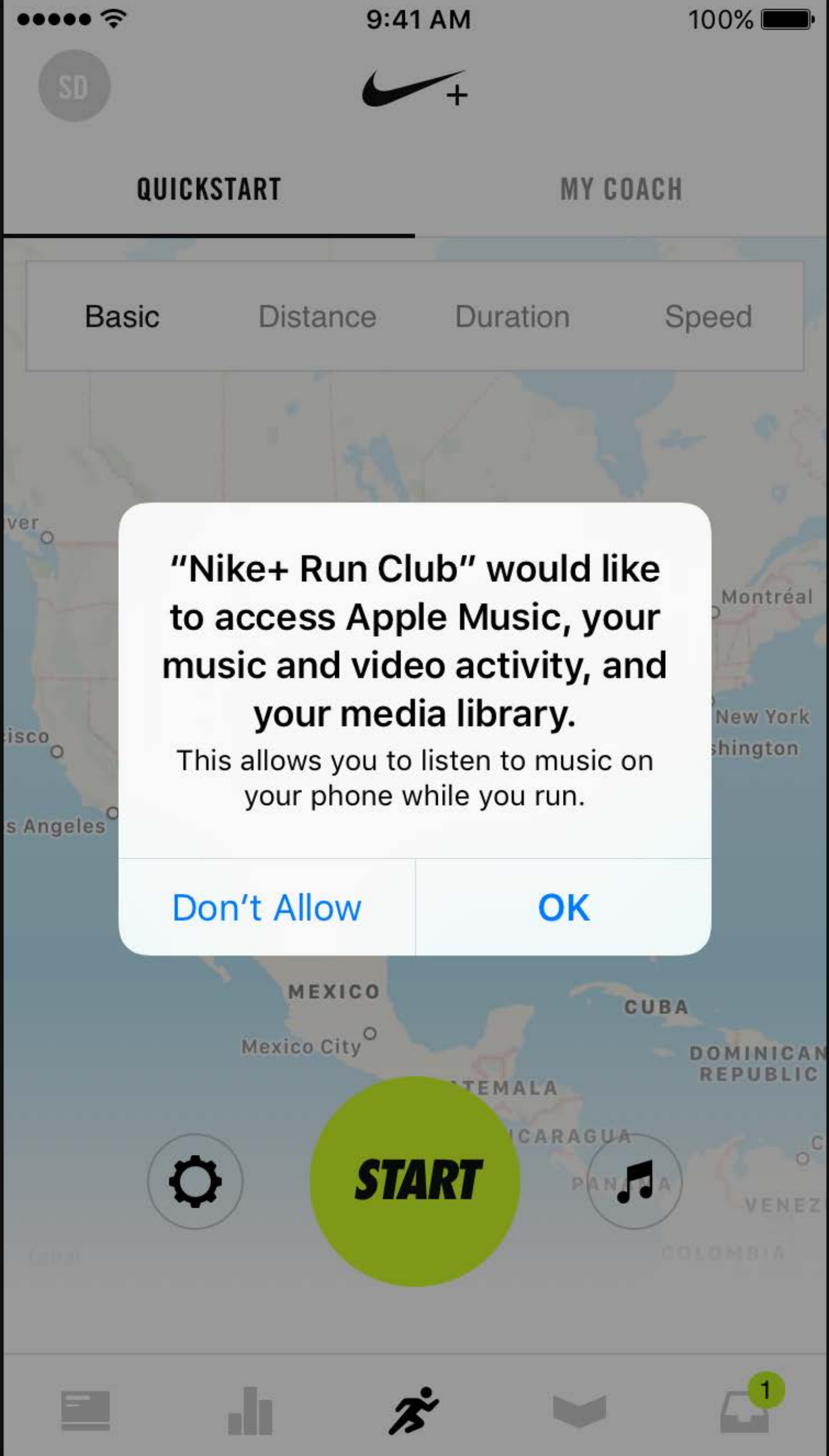
Privacy and User Consent

Most users are already signed in

No need to authenticate to use MusicKit

Requires user consent

User consent is per iOS device and per app



"Nike+ Run Club" would like to access Apple Music, your music and video activity, and your media library.

This allows you to listen to music on your phone while you run.

Don't Allow

OK

Requesting User Consent

Needed to access device library or get the music user token

Reading authorization status will not prompt

Requesting will prompt unless already authorized

Requesting User Consent

Needed to access device library or get the music user token

Reading authorization status will not prompt

Requesting will prompt unless already authorized

```
SKCloudServiceController.requestAuthorization { (status: SKCloudServiceAuthorizationStatus) in
    switch status {
        case .authorized: ...
        case .denied, .restricted: ...
        case .notDetermined: ...
    }
}
```

Fetching Subscription Status

Fetching Subscription Status

Actual status may be complex

Fetching Subscription Status

Actual status may be complex

We expose three capabilities:

- Music catalog playback
- Add to iCloud Music Library
- Can become subscriber

Fetching Subscription Status

Actual status may be complex

We expose three capabilities:

- Music catalog playback
- Add to iCloud Music Library
- Can become subscriber

Check each capability before corresponding app action

Fetching Subscription Status

Requires user consent or else you get an error

```
let controller = SKCloudServiceController()
controller.requestCapabilities { (capabilities: SKCloudServiceCapability, error: Error?) in
    let canPlay = capabilities.contains(.musicCatalogPlayback)
    let canAdd = capabilities.contains(.addToCloudMusicLibrary)
    let canSubscribe = capabilities.contains(.musicCatalogSubscriptionEligible)
    ...
}
```


Fetching Storefront Country Code

Content identifiers vary across storefronts

```
let controller = SKCloudServiceController()
controller.requestStorefrontCountryCode { (countryCode: String?, error: Error?) in
    if let storefrontCountryCode = countryCode {
        ...
    }
}
```

Fetching Storefront Country Code

Content identifiers vary across storefronts

```
let controller = SKCloudServiceController()
controller.requestStorefrontCountryCode { (countryCode: String?, error: Error?) in
    if let storefrontCountryCode = countryCode {
        var topSongsURLComponents = URLComponents()
        topSongsURLComponents.scheme = "https"
        topSongsURLComponents.host = "api.music.apple.com"
        topSongsURLComponents.path = "/v1/catalog/\(storefrontCountryCode)/charts"
        topSongsURLComponents.queryItems = [URLQueryItem(name: "types", value: "songs")]
        ...
    }
}
```

Fetching Storefront Country Code

Content identifiers vary across storefronts

```
let controller = SKCloudServiceController()
controller.requestStorefrontCountryCode { (countryCode: String?, error: Error?) in
    if let storefrontCountryCode = countryCode {
        var topSongsURLComponents = URLComponents()
        topSongsURLComponents.scheme = "https"
        topSongsURLComponents.host = "api.music.apple.com"
        topSongsURLComponents.path = "/v1/catalog/\(storefrontCountryCode)/charts"
        topSongsURLComponents.queryItems = [URLQueryItem(name: "types", value: "songs")]
        ...
    }
}
```

Fetching Music User Token

From your developer token

You should cache the token, but don't transfer it across devices

Refetch if you get a "Forbidden" status code (403) from Apple Music API

Fetching Music User Token

```
let developerToken = "ey..."

let controller = SKCloudServiceController()
controller.requestUserToken(forDeveloperToken: developerToken) {
    (userToken: String?, error: Error?) in
        if let musicUserToken = userToken {
            ...
        }
}
}
```

Fetching Music User Token

```
let developerToken = "eyJ..."

let controller = SKCloudServiceController()
controller.requestUserToken(forDeveloperToken: developerToken) {
    (userToken: String?, error: Error?) in
        if let musicUserToken = userToken {
            var request = URLRequest(url: recentlyPlayedURL)
            request.setValue("Bearer \(developerToken)", forHTTPHeaderField: "Authorization")
            request.setValue(musicUserToken, forHTTPHeaderField: "Music-User-Token")
            ...
        }
    }
}
```

Fetching Music User Token

```
let developerToken = "eyJ..."

let controller = SKCloudServiceController()
controller.requestUserToken(forDeveloperToken: developerToken) {
    (userToken: String?, error: Error?) in
        if let musicUserToken = userToken {
            var request = URLRequest(url: recentlyPlayedURL)
            request.setValue("Bearer \(developerToken)", forHTTPHeaderField: "Authorization")
            request.setValue(musicUserToken, forHTTPHeaderField: "Music-User-Token")
            ...
        }
}
```

Get access to Apple Music API

Allow users to subscribe

Play subscription and library content



9:41 AM

100%

MUSIC

Join Apple Music



Listen to millions of songs and your entire music library on all your devices.



Get handpicked playlists & albums based on the music you love.



Hear new music first, plus the best artists and DJs live or on demand.

Cancel anytime. Plan automatically renews for \$9.99/month. [Apple Music and Privacy](#) >

Get 3 Months Free

Not Now

When to Show the Subscription View?

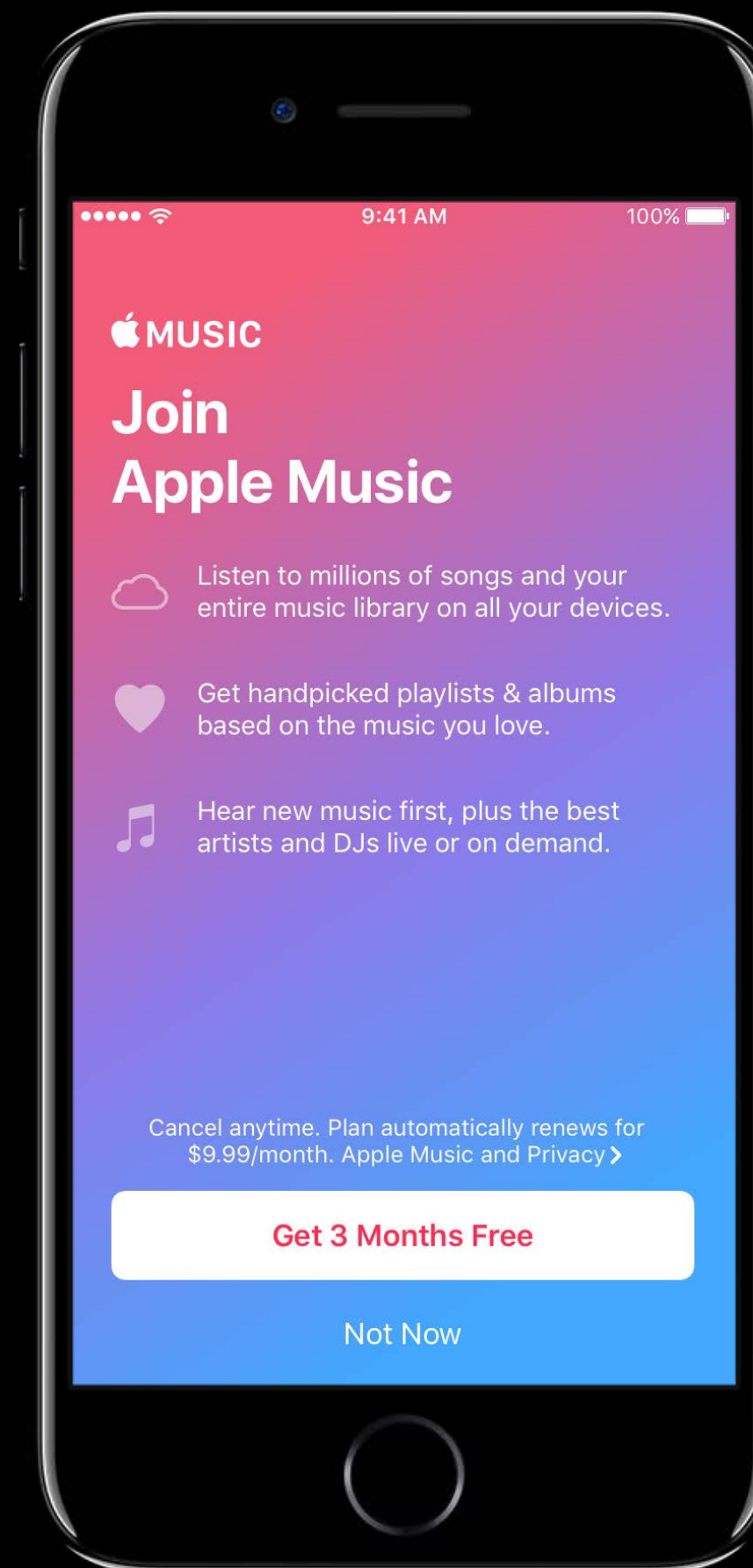
Only if `.musicCatalogPlayback` capability is *disabled* and...
`.musicCatalogSubscriptionEligible` is *enabled*

Configure the Subscription View

Choose the main message

Optional: `.messageIdentifier`

Default: `.join`



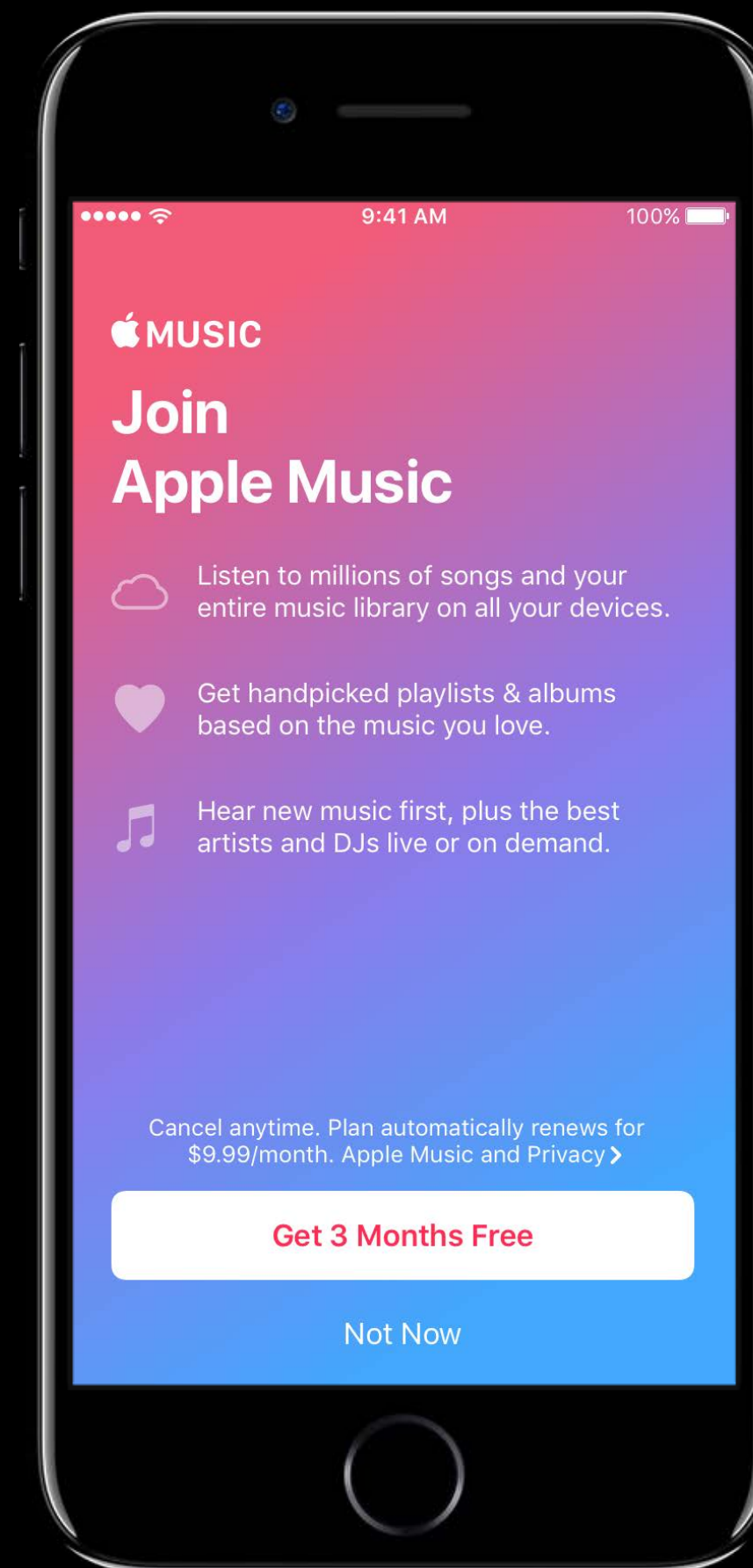
Configure the Subscription View

Choose the main message

Optional: `.messageIdentifier`

Default: `.join`

Other values:



Configure the Subscription View

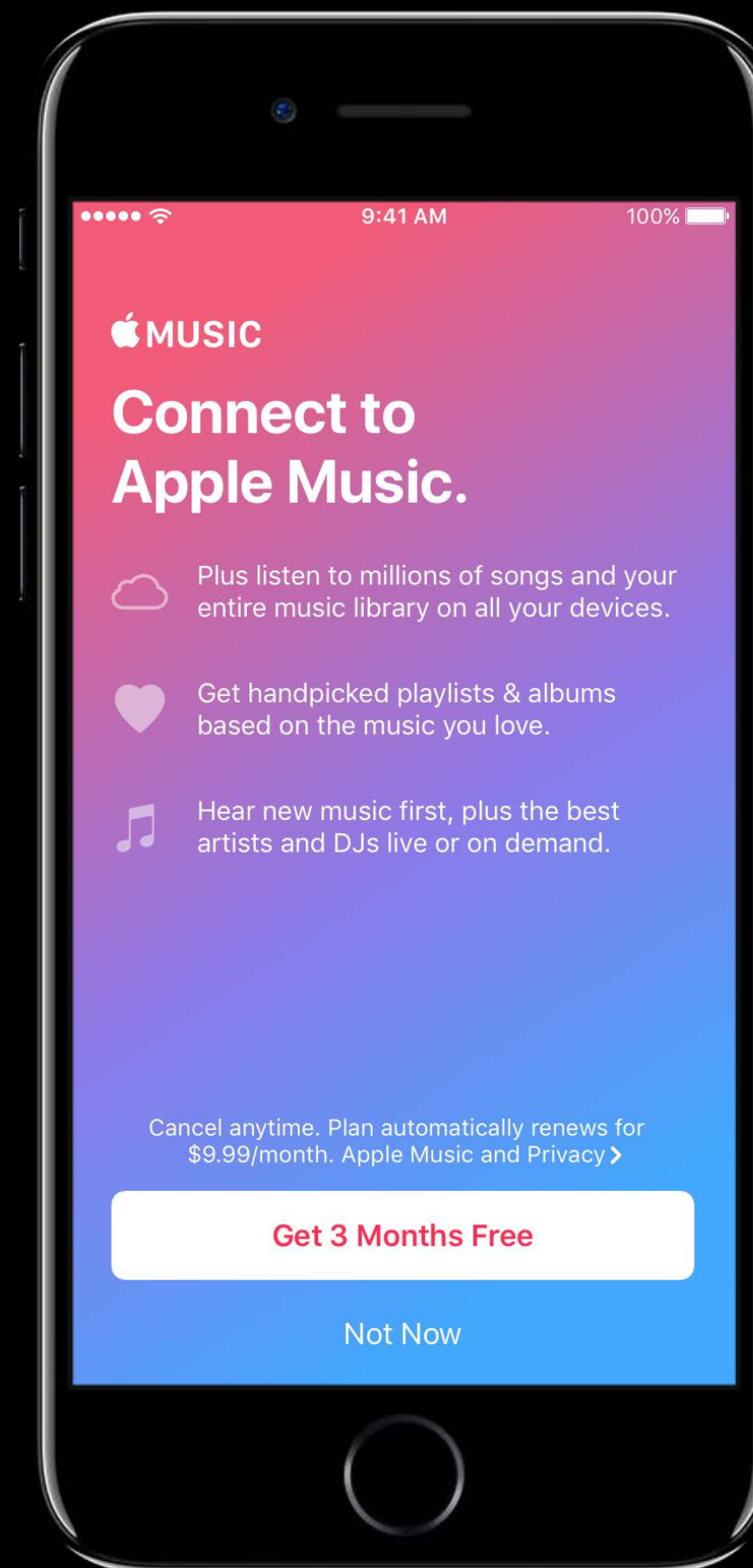
Choose the main message

Optional: `.messageIdentifier`

Default: `.join`

Other values:

`.connect`



Configure the Subscription View

Choose the main message

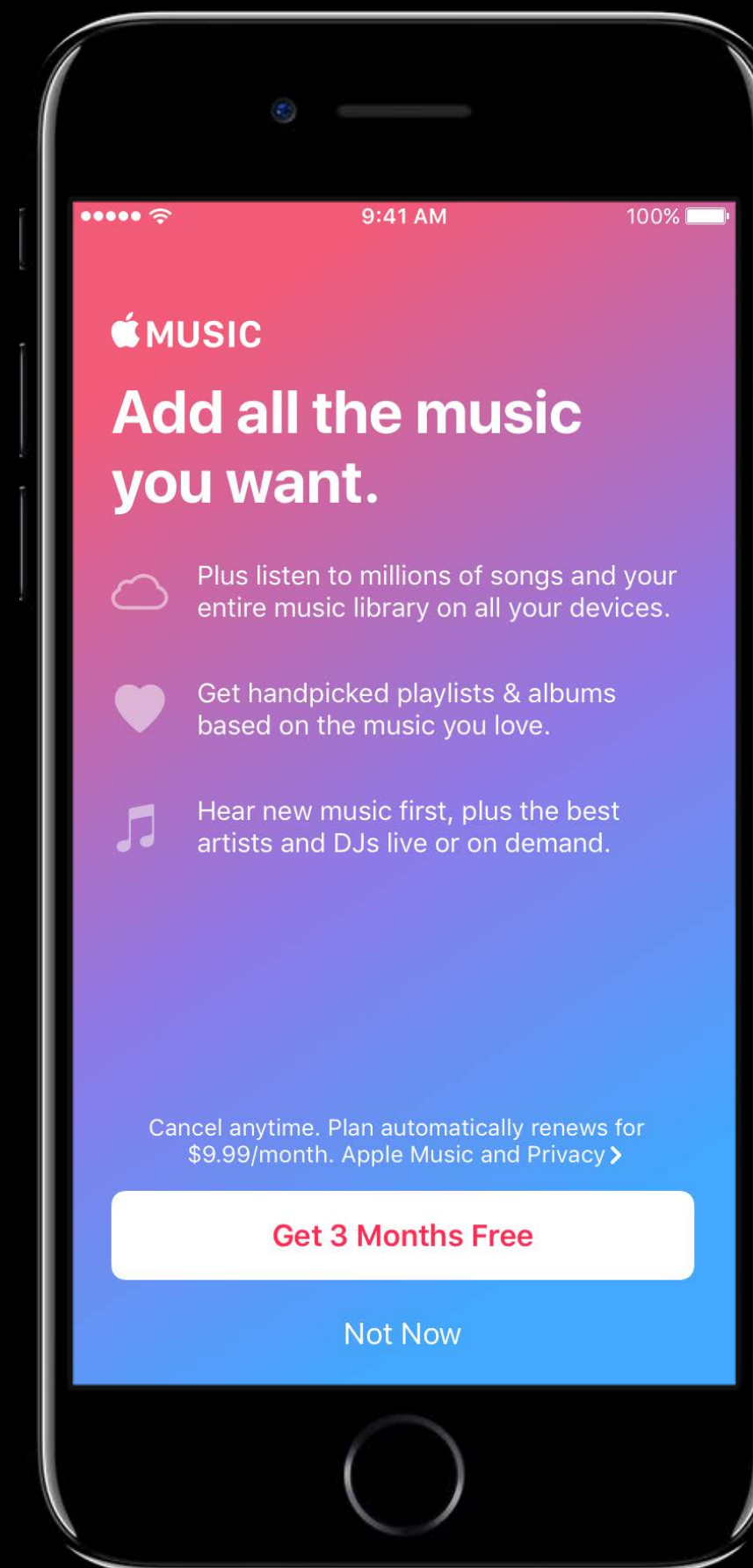
Optional: `.messageIdentifier`

Default: `.join`

Other values:

`.connect`

`.addMusic`



Configure the Subscription View

Choose the main message

Optional: `.messageIdentifier`

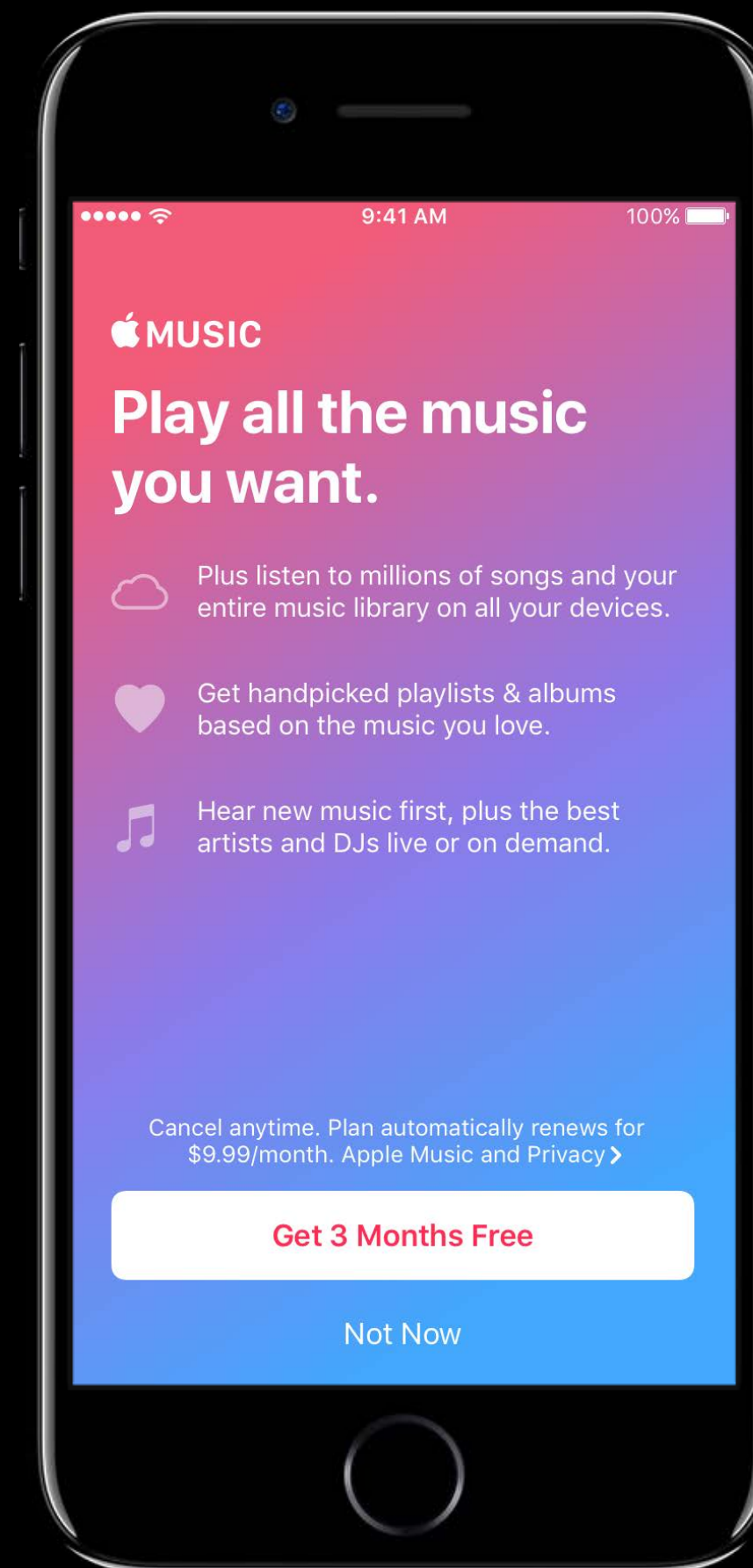
Default: `.join`

Other values:

`.connect`

`.addMusic`

`.playMusic`



Configure the Subscription View

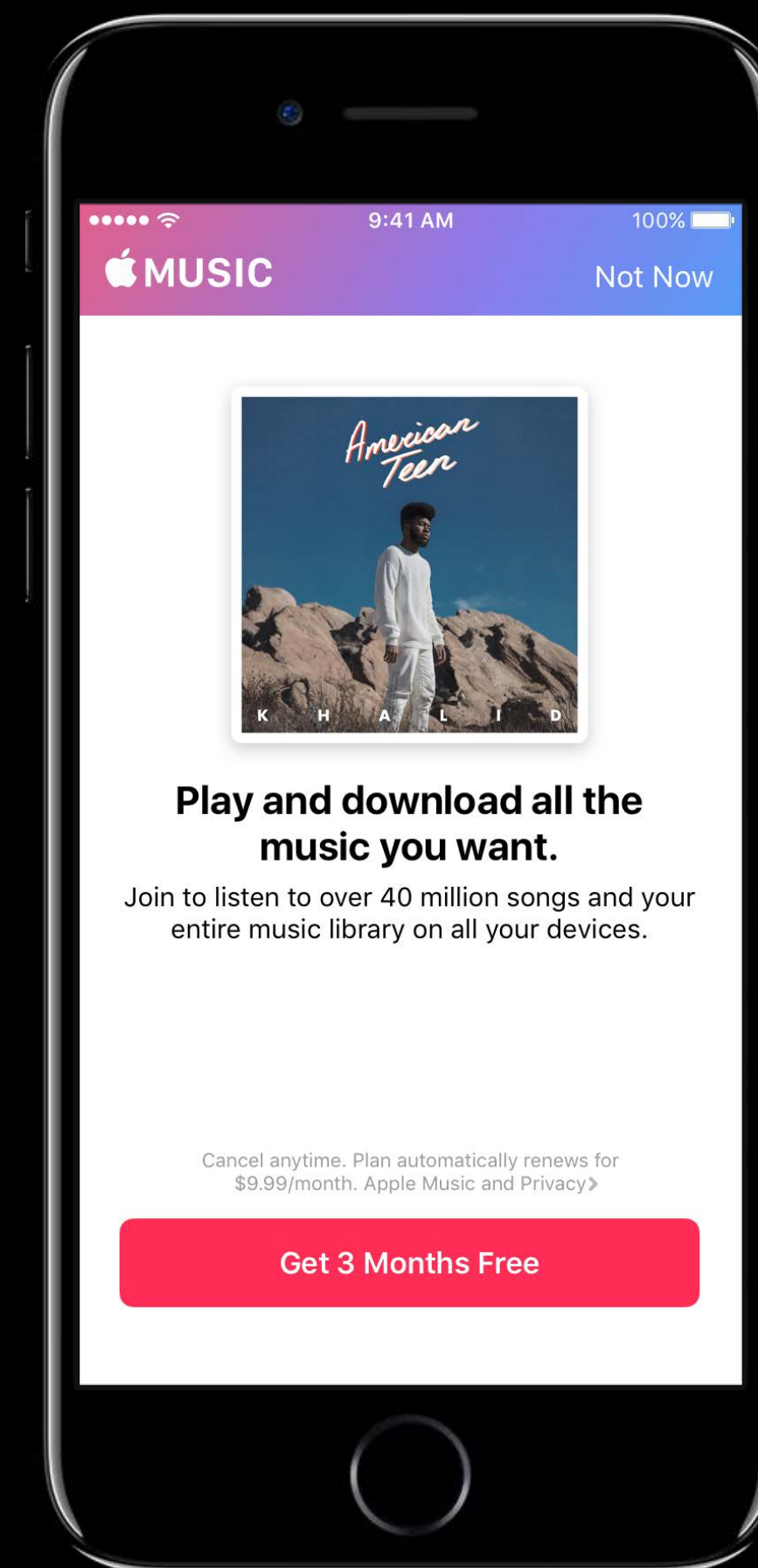
Contextual offer

Optional: `.iTunesItemIdentifier`

Type: `String`

Supported content types:

- songs
- albums
- playlists
- radio stations



Configure the Subscription View

Other options

Key	Value	
.action	.subscribe	Required
.affiliateToken	String	Optional
.campaignToken	String	Optional

```
// Showing Subscription View

let setupViewController = SKCloudServiceSetupViewController()
setupViewController.delegate = self

let setupOptions: [SKCloudServiceSetupOptionsKey: Any] = [
    .action: SKCloudServiceSetupAction.subscribe,
    .messageIdentifier: SKCloudServiceSetupMessageIdentifier.playMusic
]

setupViewController.load(options: setupOptions) { (didSucceedLoading: Bool, error: Error?) in
    if didSucceedLoading {
        self.present(setupViewController, animated: true, completion: nil)
    }
})
```

```
// Showing Subscription View
```

```
let setupViewController = SKCloudServiceSetupViewController()
setupViewController.delegate = self
```

```
let setupOptions: [SKCloudServiceSetupOptionsKey: Any] = [
    .action: SKCloudServiceSetupAction.subscribe,
    .messageIdentifier: SKCloudServiceSetupMessageIdentifier.playMusic
]
```

```
setupViewController.load(options: setupOptions) { (didSucceedLoading: Bool, error: Error?) in
    if didSucceedLoading {
        self.present(setupViewController, animated: true, completion: nil)
    }
})
```

```
// Showing Subscription View
```

```
let setupViewController = SKCloudServiceSetupViewController()  
setupViewController.delegate = self
```

```
let setupOptions: [SKCloudServiceSetupOptionsKey: Any] = [  
    .action: SKCloudServiceSetupAction.subscribe,  
    .messageIdentifier: SKCloudServiceSetupMessageIdentifier.playMusic  
]
```

```
setupViewController.load(options: setupOptions) { (didSucceedLoading: Bool, error: Error?) in  
    if didSucceedLoading {  
        self.present(setupViewController, animated: true, completion: nil)  
    }  
})
```

```
// Showing Subscription View
```

```
let setupViewController = SKCloudServiceSetupViewController()  
setupViewController.delegate = self
```

```
let setupOptions: [SKCloudServiceSetupOptionsKey: Any] = [  
    .action: SKCloudServiceSetupAction.subscribe,  
    .messageIdentifier: SKCloudServiceSetupMessageIdentifier.playMusic  
]
```

```
setupViewController.load(options: setupOptions) { (didSucceedLoading: Bool, error: Error?) in  
    if didSucceedLoading {  
        self.present(setupViewController, animated: true, completion: nil)  
    }  
})
```

```
// Showing Subscription View

let setupViewController = SKCloudServiceSetupViewController()
setupViewController.delegate = self

let setupOptions: [SKCloudServiceSetupOptionsKey: Any] = [
    .action: SKCloudServiceSetupAction.subscribe,
    .messageIdentifier: SKCloudServiceSetupMessageIdentifier.playMusic
]

setupViewController.load(options: setupOptions) { (didSucceedLoading: Bool, error: Error?) in
    if didSucceedLoading {
        self.present(setupViewController, animated: true, completion: nil)
    }
})
```

Get access to Apple Music API

Allow users to subscribe

Play subscription and library content

Two Players

MPMusicPlayerController

systemMusicPlayer

applicationQueuePlayer

Two Players

MPMusicPlayerController

systemMusicPlayer

applicationQueuePlayer

Example

Two Players

MPMusicPlayerController

systemMusicPlayer

applicationQueuePlayer

Example

Social networking app

Two Players

MPMusicPlayerController

systemMusicPlayer

applicationQueuePlayer

Example

Social networking app





Fitness app

Two Players

MPMusicPlayerController

systemMusicPlayer

applicationQueuePlayer



	systemMusicPlayer	applicationQueuePlayer
Example	Social networking app	Fitness app
Now Playing info reporting		
Remote commands handling		

Two Players

MPMusicPlayerController

systemMusicPlayer

applicationQueuePlayer



	systemMusicPlayer	applicationQueuePlayer
Example	Social networking app	Fitness app
Now Playing info reporting Remote commands handling		
Now Playing app	Music app	Your app

Two Players

MPMusicPlayerController

systemMusicPlayer

applicationQueuePlayer





	systemMusicPlayer	applicationQueuePlayer
Example	Social networking app	Fitness app
Now Playing info reporting Remote commands handling		
Now Playing app	Music app	Your app
Playback queue ownership	Remote-controlling the Music app	Your app owns the queue

Two Players

MPMusicPlayerController

systemMusicPlayer

applicationQueuePlayer






	systemMusicPlayer	applicationQueuePlayer
Example	Social networking app	Fitness app
Now Playing info reporting Remote commands handling		
Now Playing app	Music app	Your app
Playback queue ownership	Remote-controlling the Music app	Your app owns the queue
Set queue / Play next / Play later		

Two Players

MPMusicPlayerController

systemMusicPlayer

applicationQueuePlayer

	systemMusicPlayer	applicationQueuePlayer
Example	Social networking app	Fitness app
Now Playing info reporting Remote commands handling		
Now Playing app	Music app	Your app
Playback queue ownership	Remote-controlling the Music app	Your app owns the queue
Set queue / Play next / Play later		
Insert items in the middle / Remove items		

Queue Descriptors

Data

Use Case

Queue Descriptors

Data

Use Case

Media Item

MPMediaItem
MPMediaItemCollection

Device library queries (MPMediaQuery)
Media picker

Queue Descriptors

Data

Use Case

Media Item

MPMediaItem
MPMediaItemCollection

Device library queries (MPMediaQuery)
Media picker

Store Queue

Store IDs (String)

Catalog content
(e.g. catalog search)

Queue Descriptors

Data

Use Case

Media Item

MPMediaItem
MPMediaItemCollection

Device library queries (MPMediaQuery)
Media picker

Store Queue

Store IDs (String)

Catalog content
(e.g. catalog search)

Play Parameters

Dictionaries

Personalized content
(e.g. recently played)

```
let player = MPMusicPlayerController.systemMusicPlayer
player.repeatMode = .all

let album: MPMediaItemCollection = ...
let albumQueueDescriptor = MPMusicPlayerMediaItemQueueDescriptor(itemCollection: album)
albumQueueDescriptor.startItem = album.items[3]
player.setQueue(with: albumQueueDescriptor)

let catalogSongStoreID: String = ...
let catalogQueueDescriptor = MPMusicPlayerStoreQueueDescriptor(storeIDs: [catalogSongStoreID])
player.prepend(catalogQueueDescriptor)

let recentlyPlayedSongPlayParameters: [String: Any] = ...
let recentlyPlayedSongQueueDescriptor = MPMusicPlayerPlayParametersQueueDescriptor(
    playParametersQueue: [recentlyPlayedSongPlayParameters]
)
player.append(recentlyPlayedSongQueueDescriptor)
```

```
let player = MPMusicPlayerController.systemMusicPlayer
player.repeatMode = .all
```

```
let album: MPMediaItemCollection = ...
let albumQueueDescriptor = MPMusicPlayerMediaItemQueueDescriptor(itemCollection: album)
albumQueueDescriptor.startItem = album.items[3]
player.setQueue(with: albumQueueDescriptor)
```

```
let catalogSongStoreID: String = ...
let catalogQueueDescriptor = MPMusicPlayerStoreQueueDescriptor(storeIDs: [catalogSongStoreID])
player.prepend(catalogQueueDescriptor)
```

```
let recentlyPlayedSongPlayParameters: [String: Any] = ...
let recentlyPlayedSongQueueDescriptor = MPMusicPlayerPlayParametersQueueDescriptor(
    playParametersQueue: [recentlyPlayedSongPlayParameters]
)
player.append(recentlyPlayedSongQueueDescriptor)
```

```
let player = MPMusicPlayerController.systemMusicPlayer
player.repeatMode = .all
```

```
let album: MPMediaItemCollection = ...
let albumQueueDescriptor = MPMusicPlayerMediaItemQueueDescriptor(itemCollection: album)
albumQueueDescriptor.startItem = album.items[3]
player.setQueue(with: albumQueueDescriptor)
```

```
let catalogSongStoreID: String = ...
let catalogQueueDescriptor = MPMusicPlayerStoreQueueDescriptor(storeIDs: [catalogSongStoreID])
player.prepend(catalogQueueDescriptor)
```

```
let recentlyPlayedSongPlayParameters: [String: Any] = ...
let recentlyPlayedSongQueueDescriptor = MPMusicPlayerPlayParametersQueueDescriptor(
    playParametersQueue: [recentlyPlayedSongPlayParameters]
)
player.append(recentlyPlayedSongQueueDescriptor)
```

```
let player = MPMusicPlayerController.systemMusicPlayer
player.repeatMode = .all
```

```
let album: MPMediaItemCollection = ...
let albumQueueDescriptor = MPMusicPlayerMediaItemQueueDescriptor(itemCollection: album)
albumQueueDescriptor.startItem = album.items[3]
player.setQueue(with: albumQueueDescriptor)
```

```
let catalogSongStoreID: String = ...
let catalogQueueDescriptor = MPMusicPlayerStoreQueueDescriptor(storeIDs: [catalogSongStoreID])
player.prepend(catalogQueueDescriptor)
```

```
let recentlyPlayedSongPlayParameters: [String: Any] = ...
let recentlyPlayedSongQueueDescriptor = MPMusicPlayerPlayParametersQueueDescriptor(
    playParametersQueue: [recentlyPlayedSongPlayParameters]
)
player.append(recentlyPlayedSongQueueDescriptor)
```



```
let player = MPMusicPlayerController.systemMusicPlayer
player.repeatMode = .all
```

```
let album: MPMediaItemCollection = ...
let albumQueueDescriptor = MPMusicPlayerMediaItemQueueDescriptor(itemCollection: album)
albumQueueDescriptor.startItem = album.items[3]
player.setQueue(with: albumQueueDescriptor)
```

```
let catalogSongStoreID: String = ...
let catalogQueueDescriptor = MPMusicPlayerStoreQueueDescriptor(storeIDs: [catalogSongStoreID])
player.prepend(catalogQueueDescriptor)
```

```
let recentlyPlayedSongPlayParameters: [String: Any] = ...
let recentlyPlayedSongQueueDescriptor = MPMusicPlayerPlayParametersQueueDescriptor(
    playParametersQueue: [recentlyPlayedSongPlayParameters]
)
player.append(recentlyPlayedSongQueueDescriptor)
```

```
// Advanced Queue Editing using applicationQueuePlayer

let player = MPMusicPlayerController.applicationQueuePlayer
player.perform(
    queueTransaction: { (mutableQueue: MPMusicPlayerControllerMutableQueue) in
        // Remove 3rd item
        mutableQueue.remove(mutableQueue.items[2])

        // Add song with storeID after 5th item
        let songStoreID: String = ...
        let songQueueDescriptor = MPMusicPlayerStoreQueueDescriptor(storeIDs: [songStoreID])
        mutableQueue.insert(songQueueDescriptor, after: mutableQueue.items[4])
    },
    completionHandler: { (queue: MPMusicPlayerControllerQueue, error: Error?) in
        ...
    }
)
)
```

```
// Advanced Queue Editing using applicationQueuePlayer
```

```
let player = MPMusicPlayerController.applicationQueuePlayer
player.perform(
    queueTransaction: { (mutableQueue: MPMusicPlayerControllerMutableQueue) in
        // Remove 3rd item
        mutableQueue.remove(mutableQueue.items[2])

        // Add song with storeID after 5th item
        let songStoreID: String = ...
        let songQueueDescriptor = MPMusicPlayerStoreQueueDescriptor(storeIDs: [songStoreID])
        mutableQueue.insert(songQueueDescriptor, after: mutableQueue.items[4])
    },
    completionHandler: { (queue: MPMusicPlayerControllerQueue, error: Error?) in
        ...
    }
)
```

```
// Advanced Queue Editing using applicationQueuePlayer

let player = MPMusicPlayerController.applicationQueuePlayer
player.perform(
    queueTransaction: { (mutableQueue: MPMusicPlayerControllerMutableQueue) in
        // Remove 3rd item
        mutableQueue.remove(mutableQueue.items[2])

        // Add song with storeID after 5th item
        let songStoreID: String = ...
        let songQueueDescriptor = MPMusicPlayerStoreQueueDescriptor(storeIDs: [songStoreID])
        mutableQueue.insert(songQueueDescriptor, after: mutableQueue.items[4])
    },
    completionHandler: { (queue: MPMusicPlayerControllerQueue, error: Error?) in
        ...
    }
)
)
```

```
// Advanced Queue Editing using applicationQueuePlayer

let player = MPMusicPlayerController.applicationQueuePlayer
player.perform(
    queueTransaction: { (mutableQueue: MPMusicPlayerControllerMutableQueue) in
        // Remove 3rd item
        mutableQueue.remove(mutableQueue.items[2])

        // Add song with storeID after 5th item
        let songStoreID: String = ...
        let songQueueDescriptor = MPMusicPlayerStoreQueueDescriptor(storeIDs: [songStoreID])
        mutableQueue.insert(songQueueDescriptor, after: mutableQueue.items[4])
    },
    completionHandler: { (queue: MPMusicPlayerControllerQueue, error: Error?) in
        ...
    }
)
)
```

```
// Advanced Queue Editing using applicationQueuePlayer

let player = MPMusicPlayerController.applicationQueuePlayer
player.perform(
    queueTransaction: { (mutableQueue: MPMusicPlayerControllerMutableQueue) in
        // Remove 3rd item
        mutableQueue.remove(mutableQueue.items[2])

        // Add song with storeID after 5th item
        let songStoreID: String = ...
        let songQueueDescriptor = MPMusicPlayerStoreQueueDescriptor(storeIDs: [songStoreID])
        mutableQueue.insert(songQueueDescriptor, after: mutableQueue.items[4])
    },
    completionHandler: { (queue: MPMusicPlayerControllerQueue, error: Error?) in
        ...
    }
)
)
```

```
// Advanced Queue Editing using applicationQueuePlayer

let player = MPMusicPlayerController.applicationQueuePlayer
player.perform(
    queueTransaction: { (mutableQueue: MPMusicPlayerControllerMutableQueue) in
        // Remove 3rd item
        mutableQueue.remove(mutableQueue.items[2])

        // Add song with storeID after 5th item
        let songStoreID: String = ...
        let songQueueDescriptor = MPMusicPlayerStoreQueueDescriptor(storeIDs: [songStoreID])
        mutableQueue.insert(songQueueDescriptor, after: mutableQueue.items[4])
    },
    completionHandler: { (queue: MPMusicPlayerControllerQueue, error: Error?) in
        ...
    }
)
)
```

Timeframe

Timeframe

Timeframe

iOS 10.3



Music user tokens (Fall)¹

Storefront country code²

Catalog playback (with store IDs)

¹ API in SKCloudServiceController exists under a different name.

² As an approximation, you can use `Locale.current.regionCode`.

Timeframe

iOS 10.3



iOS 11



Music user tokens (Fall)¹

Storefront country code²

Catalog playback (with store IDs)

Music user tokens

Storefront country code

Catalog playback (with store IDs)

Catalog playback (with parameters)³

¹ API in `SKCloudServiceController` exists under a different name.

² As an approximation, you can use `Locale.current.regionCode`.

³ Available in an upcoming iOS 11 Beta.

Summary

Leverage Apple Music's content to keep your users engaged

Get music user token from iOS

Use it to call Apple Music API

Use Media Player to control playback

More Information

<https://developer.apple.com/wwdc17/502>

Related Sessions

What's New in Audio

Grand Ballroom B

Tuesday 1:50PM

What's New in Foundation

Hall 2

Wednesday 11:00AM

Introducing AirPlay 2

Executive Ballroom

Thursday 4:10PM

Labs

MusicKit Lab

Technology Lab G

Tue 4:10PM-6:00PM

Audio Lab

Technology Lab F

Tue 4:10PM-6:00PM

Foundation Lab

Technology Lab C

Wed 1:00PM-3:10PM

MusicKit Lab

Technology Lab F

Wed 3:10PM-6:00PM

Audio Lab

Technology Lab G

Thur 1:00PM-3:00PM

