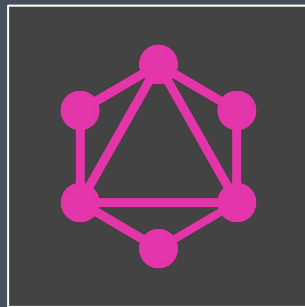


Harmonizing APIs

A comparison of GraphQL and OpenAPI
using the Spotify API

Fosdem 2025
Martin Bonnin

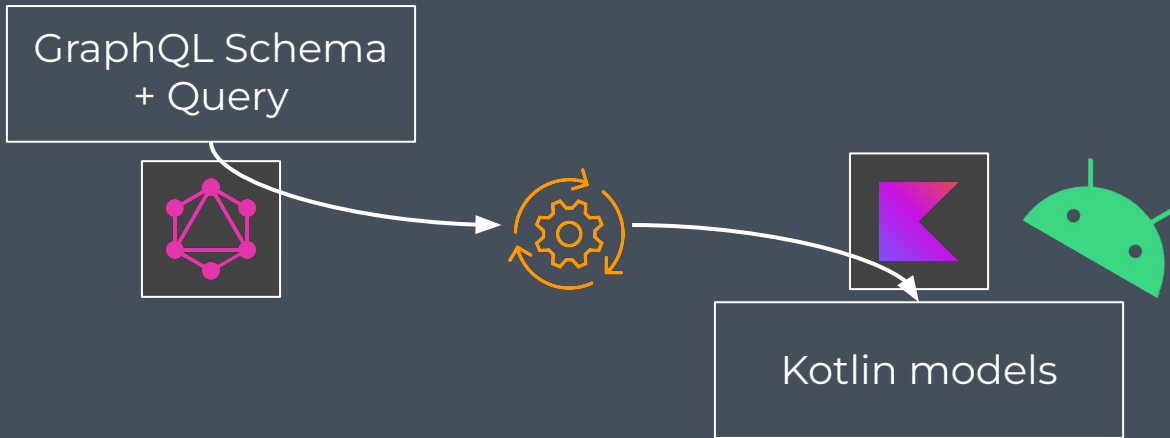




APOLLO

[apollographql/apollo-kotlin](https://github.com/apollographql/apollo-kotlin)

Apollo Kotlin



Spotify showcase

[apollographql/spotify-showcase](https://github.com/apollographql/spotify-showcase)

spotify-showcase Public

Watch 6 Fork 17 Starred 127

main 31 Branches 0 Tags

Go to file Add file Code

README Code of conduct MIT license Security

APOLLO

React + Apollo Spotify Showcase

APOLLO

Afternoon Acoustic

Afternoon Acoustic

#	TITLE	ALBUM	DATE ADDED
1	Truffle Change - The Roots	Edges Run	28 days ago
2	Where's My Love - Acoustic 3PM	Where's My Love	28 days ago
3	Nothing Arrived - Live from Seattle/London		

Architecture

The overall API architecture is made up of two GraphQL servers, one exposing subscription/mutation functionality and the other exposing query functionality. Both GraphQL servers use the Spotify REST API as their datasource, but we are hosting the subscription server on dedicated infrastructure (Railway) and the other on serverless functions (Netlify).

The Apollo Router routes incoming traffic from the client application and integrates with GraphOS to receive schema updates and report usage metrics.

About

A Spotify clone that showcases the Apollo GraphQL platform.

graphql apollo-client apollo-server apollo-graphql

Readme MIT license Code of conduct Security policy Activity Custom properties 127 stars 6 watching 17 forks Report repository

Releases

No releases published
[Create a new release](#)

Contributors 13

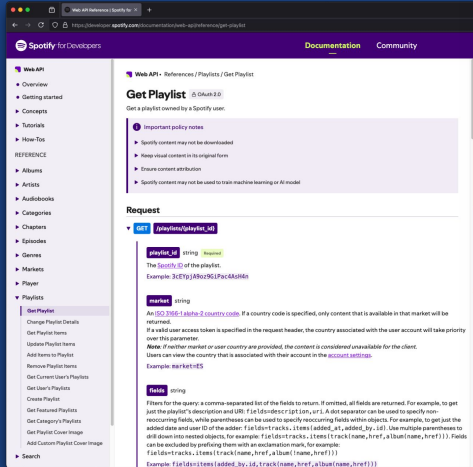
Deployments 12

production 12 hours ago
[+ 11 deployments](#)

Languages

- TypeScript 98.7%
- JavaScript 0.4%
- Shell 0.4%
- CSS 0.2%
- HTML 0.1%
- Dockerfile 0.1%
- Other 0.1%

Spotify REST API



<https://developer.spotify.com/documentation/web-api/reference/web-api/open-api-schema.yaml>

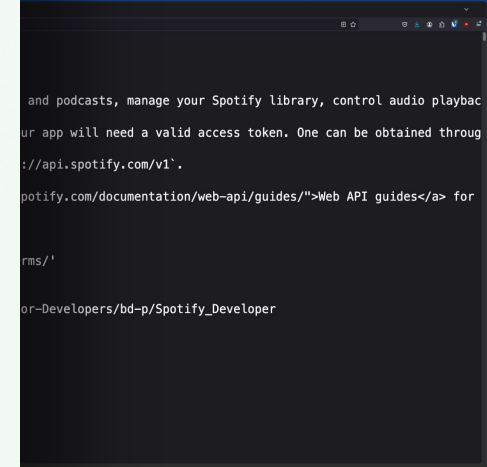


Spotify Web API 1.0.0

Share these results

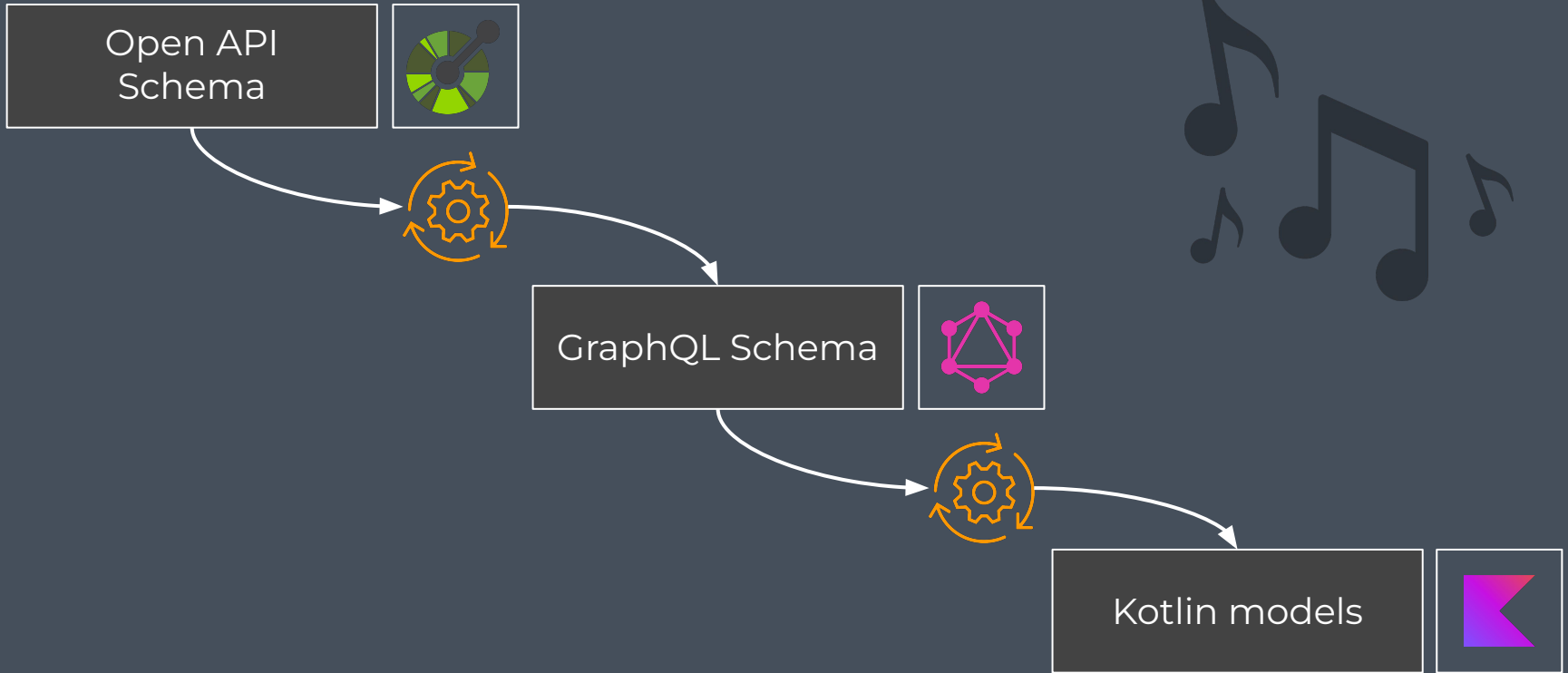
You deserve a round of API-lause!

But don't REST on your laurels



<https://developer.spotify.com/documentation/web-api/reference/web-api/open-api-schema.yaml>

Codegen all the things!



Let's dive in!

Spotify schema

```
openapi: '3.0.3'
info:
  version: '1.0.0'
  title: 'Spotify Web API'
servers:
  - url: https://api.spotify.com/v1
paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          description: A playlist response
          content:
            application/json:
              schema:
                type: object
                description: A playlist
                properties:
                  id:
                    type: string
                    description: The Spotify ID for the playlist.
                  description:
                    type: string
                    description: The playlist description. _Only returned for modified, verified playlists, otherwise_ `null`.
                  name:
                    type: string
                    description: The name of the playlist.
                  tracks:
                    type: array
                    description: The tracks in the playlist.
                    items:
                      type: object
                      description: A track.
                      properties:
                        id:
                          type: string
                          # more properties ...
```



Our goal today



```
openapi: '3.0.3'
info:
  version: '1.0.0'
  title: 'Spotify Web API'
servers:
  - url: https://api.spotify.com/v1
paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          description: A playlist response
          content:
            application/json:
              schema:
                type: object
                description: A playlist
                properties:
                  id:
                    type: string
                    description: The Spotify ID for the playlist.
                  description:
                    type: string
                    description: The playlist description. _Only returned for modified, verified p
                  name:
                    type: string
                    description: The name of the playlist.
                  tracks:
                    type: array
                    description: The tracks in the playlist.
                    items:
                      type: object
                      description: A track.
                      properties:
                        id:
                          type: string
                          # more properties ...
```



Metadata

```
openapi: '3.0.3'  
info:  
  version: '1.0.0'  
  title: 'Spotify Web API'  
servers:  
  - url: https://api.spotify.com/v1
```



Metadata

```
openapi: '3.0.3'  
info:  
  version: '1.0.0'  
  title: 'Spotify Web API'  
servers:  
  - url: https://api.spotify.com/v1
```



No metadata

```
schema {  
  query: Query  
}  
  
type Query {  
  # stuff goes here ...  
}
```



Paths

```
openapi: '3.0.3'
info:
  version: '1.0.0'
  title: 'Spotify Web API'
servers:
  - url: https://api.spotify.com/v1
paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          in: path
          schema:
            description: The Spotify ID of the
playlist.
            type: string
      responses:
        '200':
          description: A playlist response
          content:
            application/json:
              schema:
                type: object
                description: A playlist
```



Paths

```
openapi: '3.0.3'
info:
  version: '1.0.0'
  title: 'Spotify Web API'
servers:
  - url: https://api.spotify.com/v1
paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          in: path
          schema:
            description: The Spotify ID of the
playlist.
            type: string
      responses:
        '200':
          description: A playlist response
          content:
            application/json:
              schema:
                type: object
                description: A playlist
```



Root fields

```
type Query {
  getPlaylist(id: String): GetPlaylist
}

"""
A playlist
"""
type GetPlaylist {
  # more fields ...
}
```



Properties

```
content:
  application/json:
    schema:
      type: object
      description: A playlist
      properties:
        id:
          type: string
          description: The Spotify ID for the
playlist.
        description:
          type: string
          description: The playlist description.
_Only returned for modified, verified
playlists, otherwise_ `null`.
        name:
          type: string
          description: The name of the playlist.
```



Fields



Properties

```
content:
  application/json:
    schema:
      type: object
      description: A playlist
      properties:
        id:
          type: string
          description: The Spotify ID for the
playlist.
        description:
          type: string
          description: The playlist description.
_Only returned for modified, verified
playlists, otherwise_ `null`.
        name:
          type: string
          description: The name of the playlist.
```



Fields

```
"""
A playlist
"""
type GetPlaylist {
  """
  The Spotify ID of the playlist.
  """
  id: String
  """
  The playlist description. _Only returned for
modified, verified playlists, otherwise_ `null`
  """
  description: String
  """
  The name of the playlist.
  """
  name: String
}
```



Arrays

```
name:  
  type: string  
  description: The name of the playlist.  
tracks:  
  type: array  
  description: The tracks in the playlist.  
  items:  
    type: object  
    description: A track.  
    properties:  
      id:  
        type: string  
      # more properties ...
```



Arrays

```
name:  
  type: string  
  description: The name of the playlist.  
tracks:  
  type: array  
  description: The tracks in the playlist.  
  items:  
    type: object  
    description: A track.  
    properties:  
      id:  
        type: string  
        # more properties ...
```



Lists

```
type GetPlaylist {  
  # more fields ...  
  ""  
  The tracks in the playlist.  
  ""  
  tracks: [GetPlaylistTrack]  
}  
  
""  
A track  
""  
type GetPlaylistTrack {  
  id: String  
  # more fields ...  
}
```



```

openapi: '3.0.3'
info:
  version: '1.0.0'
  title: 'Spotify Web API'
servers:
  - url: https://api.spotify.com/v1
paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          description: A playlist response
          content:
            application/json:
              schema:
                type: object
                description: A playlist
                properties:
                  id:
                    type: string
                    description: The Spotify ID for the playlist.
                  description:
                    type: string
                    description: The playlist description. _Only returned for modified, verified
playlists, otherwise_ `null`.
                  name:
                    type: string
                    description: The name of the playlist.
                  tracks:
                    type: array
                    description: The tracks in the playlist.
                    items:
                      type: object
                      description: A track.
                      properties:
                        id:
                          type: string
                          # more properties ...

```



```

schema {
  query: Query
}

type Query {
  getPlaylist(id: String): GetPlaylist
}

"""
A playlist
"""
type GetPlaylist {
  """
  The Spotify ID of the playlist.
  """
  id: String
  """
  The playlist description. _Only returned for modified, verified playlists, otherwise_ `null`
  """
  description: String
  """
  The name of the playlist.
  """
  name: String
  """
  The tracks in the playlist.
  """
  tracks: [GetPlaylistTrack]
}

"""
A track
"""
type GetPlaylistTrack {
  """
  The Spotify ID for the track.
  """
  id: String
  # more fields ...
}

```



Nullability!

```
openapi: '3.0.3'
info:
  version: '1.0.0'
  title: 'Spotify Web API'
servers:
  - url: https://api.spotify.com/v1
paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          required: true
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          description: A playlist response
          content:
            application/json:
              schema:
                type: object
                description: A playlist
                properties:
                  id:
                    type: string
                    nullable: false
                    description: The Spotify ID for the playlist.
                  description:
                    type: string
                    nullable: true
                    description: The playlist description. _Only returned for modified, verified
                    playlists, otherwise_ `null`.
                  name:
                    type: string
                    nullable: false
                    description: The name of the playlist.
                  tracks:
                    type: array
                    nullable: false
                    description: The tracks in the playlist.
                    items:
                      type: object
                      description: A track.
                      properties:
                        id:
                          type: string
                          nullable: false
                          # more properties ...
```



```
schema {
  query: Query
}

type Query {
  getPlaylist(id: String!): GetPlaylist
}

"""
A playlist
"""
type GetPlaylist {
  """
  The Spotify ID of the playlist.
  """
  id: String!
  """
  The playlist description. _Only returned for modified, verified playlists, otherwise_ `null`
  """
  description: String
  """
  The name of the playlist.
  """
  name: String!
  """
  The tracks in the playlist.
  """
  tracks: [GetPlaylistTrack!]!
}

"""
A track
"""
type GetPlaylistTrack {
  """
  The Spotify ID for the track.
  """
  id: String!
  # more fields ...
}
```



```

openapi: '3.0.3'
info:
  version: '1.0.0'
  title: 'Spotify Web API'
servers:
  - url: https://api.spotify.com/v1
paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          required: true
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          description: A playlist response
          content:
            application/json:
              schema:
                type: object
                description: A playlist
                properties:
                  id:
                    type: string
                    nullable: false
                    description: The Spotify ID for the playlist.
                  description:
                    type: string
                    nullable: true
                    description: The playlist description. _Only returned for modified, verified
playlists, otherwise_ `null`.
                  name:
                    type: string
                    nullable: false
                    description: The name of the playlist.
                  tracks:
                    type: array
                    nullable: false
                    description: The tracks in the playlist.
                    items:
                      type: object
                      description: A track.
                      properties:
                        id:
                          type: string
                          nullable: false
                          # more properties ...

```



```

schema {
  query: Query
}

type Query {
  getPlaylist(id: String!): GetPlaylist
}

"""
A playlist
"""
type GetPlaylist {
  """
  The Spotify ID of the playlist.
  """
  id: String!
  """
  The playlist description. _Only returned for modified, verified playlists, otherwise_ `null`
  """
  description: String
  """
  The name of the playlist.
  """
  name: String!
  """
  The tracks in the playlist.
  """
  tracks: [GetPlaylistTrack!]
}

"""
A track
"""
type GetPlaylistTrack {
  """
  The Spotify ID for the track.
  """
  id: String!
  # more fields ...
}

```



```

openapi: '3.0.3'
info:
  version: '1.0.0'
  title: 'Spotify Web API'
servers:
  - url: https://api.spotify.com/v1
paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          required: true
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          description: A playlist response
          content:
            application/json:
              schema:
                type: object
                description: A playlist
                properties:
                  id:
                    type: string
                    nullable: false
                    description: The Spotify ID for the playlist.
                  description:
                    type: string
                    nullable: true
                    description: The playlist description. _Only returned for modified, verified
                    playlists, otherwise_ `null`.
                  name:
                    type: string
                    nullable: false
                    description: The name of the playlist.
                  tracks:
                    type: array
                    nullable: false
                    description: The tracks in the playlist.
                    items:
                      type: object
                      description: A track.
                      properties:
                        id:
                          type: string
                          nullable: false
                          # more properties ...

```



```

schema {
  query: Query
}

type Query {
  getPlaylist(id: String!): GetPlaylist
}

"""
A playlist
"""
type GetPlaylist {
  """
  The Spotify ID of the playlist.
  """
  id: String!
  """
  The playlist description. _Only returned for modified, verified playlists, otherwise_ `null`
  """
  description: String
  """
  The name of the playlist.
  """
  name: String!
  """
  The tracks in the playlist.
  """
  tracks: [GetPlaylistTrack!]
}

"""
A track
"""
type GetPlaylistTrack {
  """
  The Spotify ID for the track.
  """
  id: String!
  # more fields ...
}

```

New objects for every path? 🤔



OpenAPI components

§ 4.8.7 Components Object

Holds a set of reusable objects for different aspects of the OAS. All objects defined within the Components Object will have no effect on the API unless they are explicitly referenced from outside the Components Object.

§ 4.8.7.1 Fixed Fields

Field Name	Type	Description
schemas	Map[string, Schema Object]	An object to hold reusable Schema Objects .
responses	Map[string, Response Object Reference Object]	An object to hold reusable Response Objects .
parameters	Map[string, Parameter Object Reference Object]	An object to hold reusable Parameter Objects .
examples	Map[string, Example Object Reference Object]	An object to hold reusable Example Objects .
requestBodies	Map[string, Request Body Object Reference Object]	An object to hold reusable Request Body Objects .
headers	Map[string, Header Object Reference Object]	An object to hold reusable Header Objects .
securitySchemes	Map[string, Security Scheme Object Reference Object]	An object to hold reusable Security Scheme Objects .
	Map[string, Link Object Reference Object]	An object to hold reusable Link Objects .

```
paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          description: A playlist response
          content:
            application/json:
              schema:
                type: object
                description: A playlist
                properties:
                  id:
                    type: string
                    description: The Spotify ID for the playlist.
                  description:
                    type: string
                    description: The playlist description. _Only returned for modified, verified
playlists, otherwise_ `null`.
                  name:
                    type: string
                    description: The name of the playlist.
                  tracks:
                    type: array
                    description: The tracks in the playlist.
                    items:
                      type: object
                      description: A track.
                      properties:
                        id:
                          type: string
                          # more properties ...
```



```
paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          required: true
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          $ref: '#/components/responses/OnePlaylist'

components:
  responses:
    OnePlaylist:
      description: A playlist
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/PlaylistObject'

schemas:
  PlaylistObject:
    type: object
    x-spotify-docs-type: PlaylistObject
    properties:
      # properties here
    tracks:
      type: array
      description: The tracks of the playlist.
      items:
        $ref: '#/components/schemas/PlaylistTrackObject'

  PlaylistTrackObject:
    type: object
    x-spotify-docs-type: PlaylistTrackObject
    properties:
      added_at:
```




```
paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          required: true
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          $ref: '#/components/responses/OnePlaylist'

components:
  responses:
    OnePlaylist:
      description: A playlist
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/PlaylistObject'

schemas:
  PlaylistObject:
    type: object
    x-spotify-docs-type: PlaylistObject
    properties:
      # properties here
    tracks:
      type: array
      description: The tracks of the playlist.
      items:
        $ref: '#/components/schemas/PlaylistTrackObject'

  PlaylistTrackObject:
    type: object
    x-spotify-docs-type: PlaylistTrackObject
    properties:
      added_at:
```



```

paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          required: true
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          $ref: '#/components/responses/OnePlaylist'

components:
  responses:
    OnePlaylist:
      description: A playlist
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/PlaylistObject'

schemas:
  PlaylistObject:
    type: object
    x-spotify-docs-type: PlaylistObject
    properties:
      # properties here
    tracks:
      type: array
      description: The tracks of the playlist.
      items:
        $ref: '#/components/schemas/PlaylistTrackObject'

  PlaylistTrackObject:
    type: object
    x-spotify-docs-type: PlaylistTrackObject
    properties:
      added_at:

```

```

schema {
  query: Query
}

type Query {
  getPlaylist(id: String!) OnePlaylist
}

"""
A playlist
"""
type OnePlaylist {
  """
  The Spotify ID of the playlist.
  """
  id: String!
  """
  The playlist description. _Only returned for modified, verified playlists, otherwise_ `null`
  """
  description: String
  """
  The name of the playlist.
  """
  name: String!
  """
  The tracks in the playlist.
  """
  tracks: [PlaylistTrackObject!]!
}

"""
A track
"""
type PlaylistTrackObject {
  """
  The Spotify ID for the track.
  """
  id: String!
  # more fields ...
}

```



allOf for interfaces

```
components:
  schemas:
    SimplifiedAlbumObject:
      allOf:
        - $ref: '#/components/schemas/AlbumBase'
        - type: object
          required:
            - artists
          properties:
            artists:
              type: array
              items:
                $ref:
                  '#/components/schemas/SimplifiedArtistObject'
```

```
interface AlbumBase {
  id: String
  name: String
}

type SimplifiedAlbumObject implements AlbumBase {
  id: String
  name: String
  artists: [SimplifiedArtistObject]
}
```



oneOf for unions

```
components:
  schemas:
    QueueObject:
      type: object
      x-spotify-docs-type: QueueObject
      properties:
        currently_playing:
          oneOf:
            - $ref: '#/components/schemas/TrackObject'
            - $ref: '#/components/schemas/EpisodeObject'
```

```
type QueueObject {
  currently_playing: CurrentlyPlaying
}
```

```
union CurrentlyPlaying = TrackObject | EpisodeObject
```



```

paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          required: true
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          $ref: '#/components/responses/OnePlaylist'

components:
  responses:
    OnePlaylist:
      description: A playlist
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/PlaylistObject'

schemas:
  PlaylistObject:
    type: object
    x-spotify-docs-type: PlaylistObject
    properties:
      # properties here
    tracks:
      type: array
      description: The tracks of the playlist.
      items:
        $ref: '#/components/schemas/PlaylistTrackObject'

  PlaylistTrackObject:
    type: object
    x-spotify-docs-type: PlaylistTrackObject
    properties:
      added_at:

```

```

schema {
  query: Query
}

type Query {
  getPlaylist(id: String!): GetPlaylist
}

"""
A playlist
"""
type GetPlaylist {
  """
  The Spotify ID of the playlist.
  """
  id: String!
  """
  The playlist description. _Only returned for modified, verified playlists, otherwise_ `null`
  """
  description: String
  """
  The name of the playlist.
  """
  name: String!
  """
  The tracks in the playlist.
  """
  tracks: [GetPlaylistTrack!]!
}

"""
A track
"""
type GetPlaylistTrack {
  """
  The Spotify ID for the track.
  """
  id: String!
  # more fields ...
}

```



```

paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          required: true
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          $ref: '#/components/responses/OnePlaylist'

components:
  responses:
    OnePlaylist:
      description: A playlist
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/PlaylistObject'

schemas:
  PlaylistObject:
    type: object
    x-spotify-docs-type: PlaylistObject
    properties:
      # properties here
    tracks:
      type: array
      description: The tracks of the playlist.
      items:
        $ref: '#/components/schemas/PlaylistTrackObject'

  PlaylistTrackObject:
    type: object
    x-spotify-docs-type: PlaylistTrackObject
    properties:
      added_at:

```



```

schema {
  query: Query
}

type Query {
  getPlaylist(id: String!) OnePlaylist
}

"""
A playlist
"""
type OnePlaylist {
  """
  The Spotify ID of the playlist.
  """
  id: String!
  """
  The playlist description. _Only returned for modified, verified playlists, otherwise_ `null`
  """
  description: String
  """
  The name of the playlist.
  """
  name: String!
  """
  The tracks in the playlist.
  """
  tracks: [PlaylistTrackObject!]!
}

"""
A track
"""
type PlaylistTrackObject {
  """
  The Spotify ID for the track.
  """
  id: String!
  # more fields ...
}

```





Success!

Or is it?...

```

paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          required: true
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          $ref: '#/components/responses/OnePlaylist'

components:
  responses:
    OnePlaylist:
      description: A playlist
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/PlaylistObject'

schemas:
  PlaylistObject:
    type: object
    x-spotify-docs-type: PlaylistObject
    properties:
      # properties here
    tracks:
      type: array
      description: The tracks of the playlist.
      items:
        $ref: '#/components/schemas/PlaylistTrackObject'

  PlaylistTrackObject:
    type: object
    x-spotify-docs-type: PlaylistTrackObject
    properties:
      added_at:

```

```

schema {
  query: Query
}

type Query {
  getPlaylist(id: String!): OnePlaylist
}

"""
A playlist
"""
type OnePlaylist {
  """
  The Spotify ID of the playlist.
  """
  id: String!
  """
  The playlist description. _Only returned for modified, verified playlists, otherwise_ `null`
  """
  description: String
  """
  The name of the playlist.
  """
  name: String!
  """
  The tracks in the playlist.
  """
  tracks: [PlaylistTrackObject!]
}

"""
A track
"""
type PlaylistTrackObject {
  """
  The Spotify ID for the track.
  """
  id: String!
  # more fields ...
}

```



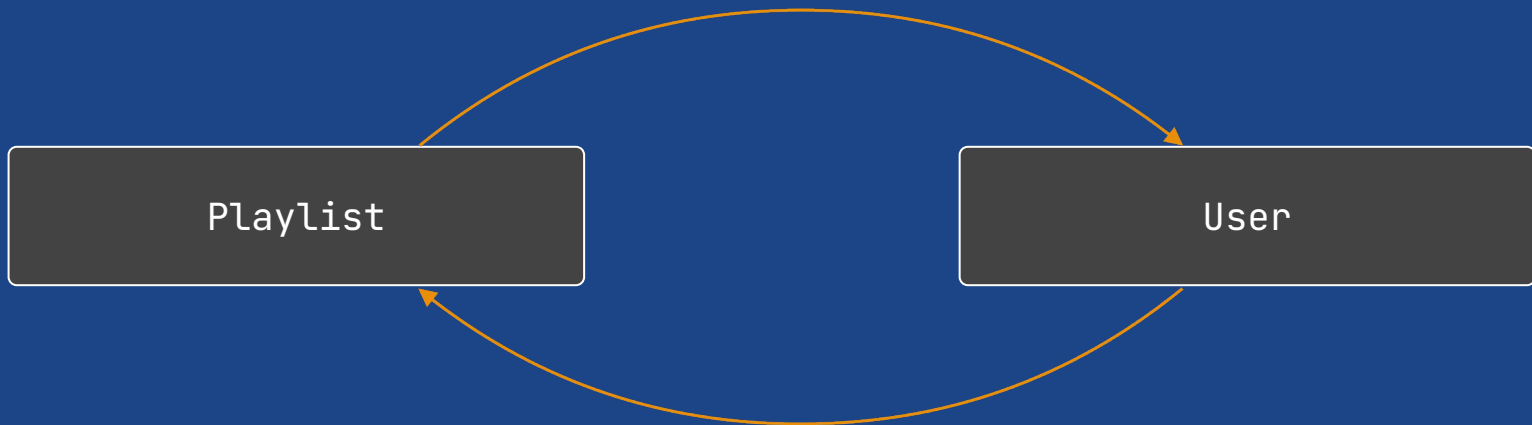
GraphQL is a “query” language

```
query GetPlaylist {  
  getPlaylist(id: "42") {  
    id  
    name  
    owner {  
      name  
      playlists {  
        name  
      }  
    }  
  }  
}
```

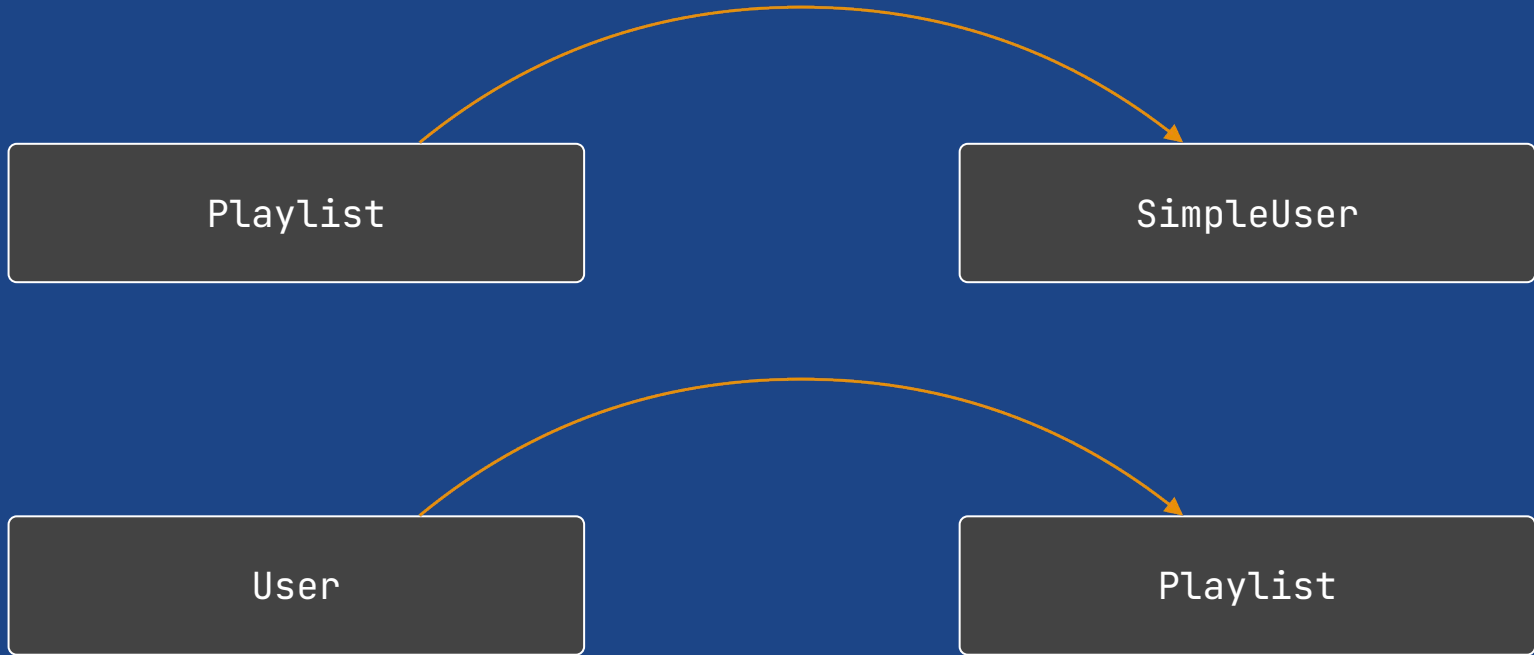


```
{  
  "data": {  
    "getPlaylist": {  
      "id": "42",  
      "name": "Belgian Classics 🇧🇪",  
      "owner": {  
        "name": "Martin",  
        "playlists": [  
          { "name": "Java Metal 🙌" },  
          { "name": "Belgian Classics 🇧🇪" }  
        ]  
      }  
    }  
  }  
}
```

Cycles everywhere!



Cycles everywhere!



REST is about resources

GET /v1/playlists/42



```
{
  "id": "42",
  "name": "Belgian Classics 🇧🇪",
  "owner": {
    "external_urls": {
      "spotify": "string"
    },
    "followers": {
      "href": "string",
      "total": 0
    },
    "href": "string",
    "id": "string",
    "type": "user",
    "uri": "string",
    "display_name": "string"
  },
  "public": false,
  "snapshot_id": "string",
  "tracks": {
    "href": "https://api.spotify.com/v1/me/shows?offset=0&limit=20",
    "limit": 20,
    "next": "https://api.spotify.com/v1/me/shows?offset=1&limit=1",
    "offset": 0,
    "previous": "https://api.spotify.com/v1/me/shows?offset=1&limit=1",
    "total": 4,
    "items": [
      {
        "added_at": "string",
        "added_by": {
          "external_urls": {
            "spotify": "string"
          },
          "followers": {
            "href": "string",
            "total": 0
          },
          "href": "string",
          "id": "string",
          "type": "user",
          "uri": "string"
        },
        "is_local": false,
        "track": {
          "album": {
            "album_type": "compilation",
            "total_tracks": 9,
            "available_markets": [
              "CA",
              "BR",
              "IT"
            ],
            "external_urls": {
```

Several variants

TrackObject

SimplifiedTrackObject

LinkedTrackObject

PlaylistTrackObject

Several variants

TrackObject

SimplifiedTrackObject

LinkedTrackObject

Same GraphQL entity

PlaylistTrackObject



*“One does not ~~simply~~
automatically convert OpenAPI to
idiomatic GraphQL”*

– Martin Bonnin

Guidelines for better codegen



✓ Use ✓

`operationId`

Components

`nullable`

`required: true`

`allOf` for inheritance

`oneOf` for unions



⊘ Do not use ⊘

`anyOf`

non-string enums

nested `oneOf`

Wrap up

GraphQL 	OpenAPI 
Statically typed	Statically typed
Schema + query language	Schema language
No versioning	Versioned
Single endpoint	Multiple endpoints
Interfaces, unions	allOf, oneOf
Enums	enum
Concise	Flexible

Thanks!

GraphQL Conf 2025

September 08-10

Amsterdam

graphql.org/conf/2025

Merci!

- <https://spec.graphql.org/draft/>
- <https://json-schema.org/specification>
- <https://spec.openapis.org/oas/latest.html>
- <https://developer.spotify.com/documentation/web-api/reference>
- <https://github.com/apollographql/spotify-showcase>
- <https://github.com/apollographql/apollo-kotlin>



Martin Bonnin

```
openapi: '3.0.3'
info:
  version: '1.0.0'
  title: 'Spotify Web API'
servers:
  - url: https://api.spotify.com/v1
paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          description: A playlist response
          content:
            application/json:
              schema:
                type: object
                description: A playlist
                properties:
                  id:
                    type: string
                    description: The Spotify ID for the playlist.
                  description:
                    type: string
                    description: The playlist description. _Only returned for modified, verified
                    playlists, otherwise_ `null`.
                  name:
                    type: string
                    description: The name of the playlist.
                  tracks:
                    type: array
                    description: The tracks in the playlist.
                    items:
                      type: object
                      description: A track.
                      properties:
                        id:
                          type: string
                          # more properties ...
```



```
openapi: '3.0.3'
info:
  version: '1.0.0'
  title: 'Spotify Web API'
servers:
  - url: 'https://api.spotify.com/v1'
paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          description: A playlist response
          content:
            application/json:
              schema:
                type: object
                description: A playlist
                properties:
                  id:
                    type: string
                    description: The Spotify ID for the playlist.
                  description:
                    type: string
                    description: The playlist description. _Only returned for modified, verified
                    playlists, otherwise_ `null`.
                  name:
                    type: string
                    description: The name of the playlist.
                  tracks:
                    type: array
                    description: The tracks in the playlist.
                    items:
                      type: object
                      description: A track.
                      properties:
                        id:
                          type: string
                          # more properties ...
```

```
schema {
  query: Query
}

type Query {
  # stuff goes here ...
}
```



```
openapi: '3.0.3'
info:
  version: '1.0.0'
  title: 'Spotify Web API'
servers:
  - url: https://api.spotify.com/v1
paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          description: A playlist response
          content:
            application/json:
              schema:
                type: object
                description: A playlist
                properties:
                  id:
                    type: string
                    description: The Spotify ID for the playlist.
                  description:
                    type: string
                    description: The playlist description. _Only returned for modified, verified
                    playlists, otherwise_ `null`.
                  name:
                    type: string
                    description: The name of the playlist.
                  tracks:
                    type: array
                    description: The tracks in the playlist.
                    items:
                      type: object
                      description: A track.
                      properties:
                        id:
                          type: string
                          # more properties ...
```



```
schema {
  query: Query
}

type Query {
  getPlaylist(id: String!): GetPlaylist
}

"""
A playlist
"""
type GetPlaylist {
  # more fields ...
}
```



```
openapi: '3.0.3'
info:
  version: '1.0.0'
  title: 'Spotify Web API'
servers:
  - url: https://api.spotify.com/v1
paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          description: A playlist response
          content:
            application/json:
              schema:
                type: object
                description: A playlist
                properties:
                  id:
                    type: string
                    description: The Spotify ID for the playlist.
                  description:
                    type: string
                    description: The playlist description. _Only returned for modified, verified
playlists, otherwise_ `null`.
                  name:
                    type: string
                    description: The name of the playlist.
                  tracks:
                    type: array
                    description: The tracks in the playlist.
                    items:
                      type: object
                      description: A track.
                      properties:
                        id:
                          type: string
                          # more properties ...
```



```
schema {
  query: Query
}

type Query {
  getPlaylist(id: String!): GetPlaylist
}

"""
A playlist
"""
type GetPlaylist {
  """
  The Spotify ID of the playlist.
  """
  id: String!
  """
  The playlist description. _Only returned for modified, verified playlists, otherwise_ `null`
  """
  description: String!
  """
  The name of the playlist.
  """
  name: String!
}
```



```

openapi: '3.0.3'
info:
  version: '1.0.0'
  title: 'Spotify Web API'
servers:
  - url: https://api.spotify.com/v1
paths:
  /playlists/{id}:
    get:
      operationId: get-playlist
      description: Get a playlist by id.
      parameters:
        - name: id
          in: path
          schema:
            description: The Spotify ID of the playlist.
            type: string
      responses:
        '200':
          description: A playlist response
          content:
            application/json:
              schema:
                type: object
                description: A playlist
                properties:
                  id:
                    type: string
                    description: The Spotify ID for the playlist.
                  description:
                    type: string
                    description: The playlist description. _Only returned for modified, verified
playlists, otherwise_ `null`.
                  name:
                    type: string
                    description: The name of the playlist.
                tracks:
                  type: array
                  description: The tracks in the playlist.
                  items:
                    type: object
                    description: A track.
                    properties:
                      id:
                        type: string
                        # more properties ...

```



```

schema {
  query: Query
}

type Query {
  getPlaylist(id: String!): GetPlaylist
}

"""
A playlist
"""
type GetPlaylist {
  """
  The Spotify ID of the playlist.
  """
  id: String!
  """
  The playlist description. _Only returned for modified, verified playlists, otherwise_ `null`
  """
  description: String!
  """
  The name of the playlist.
  """
  name: String!
  """
  The tracks in the playlist.
  """
  tracks: [GetPlaylistTrack!]
}

"""
A track
"""
type GetPlaylistTrack {
  """
  The Spotify ID for the track.
  """
  id: String!
  # more fields ...
}

```

