SwiftCloud API Documentation

Overview

Prepared for:

- Authorized API Clients

Prepared by:

- SwiftCloud Development Team (cynic123)

Introduction

SwiftCloud is a comprehensive music analytics platform dedicated to Taylor Swift's discography. It offers various APIs to interact with and analyze song data, providing insights into song plays, popularity trends, and more. The platform is divided into four main microservices, each responsible for different aspects of the analytics:

- **Songs**: Handles song search by various criteria.
- **Search**: Provides advanced search functionality across the data.
- **Popularity**: Manages and analyzes the popularity of songs, artists, and albums.
- **Trends**: Analyzes and provides trending information.

This documentation provides detailed information on the available endpoints, including request and response formats, example requests, and expected outputs.

Preliminary Endpoint

API Base

Endpoint: `/api`

Method: `GET`

Description: This endpoint serves as the base for all API requests. It can be used to verify that the API server is running and accessible.

Request:

- No request parameters.

Response:

- **Status 200**: Confirms that the API server is running.

Body: JSON object confirming the API server status.

```
{
  "message": "API server is running"
}
```

Example Request:

```
curl -X GET http://localhost:3000/api
```

Example Response:

```
{
  "message": "API server is running"
}
```

Songs Service

Health Check

Endpoint: `/songs/health`

Method: `GET`

Description: This endpoint is used to check the health status of the Songs microservice.

Request

- No request parameters.

Response:

- **Status 200**: Returns the health status.
- **Status 500**: Returns an error if the service is not reachable or there is an internal server error.

Sample Request:

```
curl -X GET http://localhost:3000/api/songs/health
```

Sample Response:

```
{
  "status": "Welcome to Songs Service!"
}
```

Get All Songs

Endpoint: `/songs/all`

Method: `GET`

Description: This endpoint returns a list of all songs.

Request:

- No request parameters.

Response:

- **Status 200**: Returns a list of all songs.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl -X GET http://localhost:3000/api/songs/all
```

Sample Response:

```
[
  {
    "writers": ["Taylor Swift", "Aaron Dessner"],
    "plays": [
      { "month": "June", "count": 68 },
      { "month": "July", "count": 6 },
      { "month": "August", "count": 61 }
    ],
    "id": "66abb3ba47a72962db72bdbe",
    "title": "The 1",
    "artist": "Taylor Swift",
    "album": "Folklore",
    "year": 2020
  },
  {
    "writers": ["Taylor Swift", "Max Martin", "Shellback"],
    "plays": [
      { "month": "June", "count": 27 },
      { "month": "July", "count": 30 },
      { "month": "August", "count": 32 }
    ],
    "id": "66abb3ba47a72962db72bdbf",
    "title": "22",
    "artist": "Taylor Swift",
    "album": "Red",
    "year": 2012
  },
  {
    "writers": ["Taylor Swift", "Louis Bell", "Frank Dukes"],
    "plays": [
      { "month": "June", "count": 90 },
      { "month": "July", "count": 17 },
      { "month": "August", "count": 87 }
    ],
    "id": "66abb3ba47a72962db72bdc0",
    "title": "Afterglow",
    "artist": "Taylor Swift",
    "album": "Lover",
    "year": 2019
  },
  // ...continued
]
```

Get Songs by Year

Endpoint: `/songs/year/:year`

Method: `GET`

Description: This endpoint returns a list of songs for a specified year.

Request Parameter:

- `year` (number, required): The year for which songs are to be fetched. Must be a valid year between 1900 and the current year.

Response:

- **Status 200**: Returns a list of songs for the specified year.
- **Status 400**: Returns an error if the year parameter is missing, not a number, or out of range.
- **Status 404**: Returns an error if no songs are found for the specified year.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl -X GET http://localhost:3000/api/songs/year/2020
```

Sample Response:

```
[
  {
    "writers": ["Taylor Swift", "Aaron Dessner"],
    "plays": [
      { "month": "June", "count": 68 },
      { "month": "July", "count": 6 },
      { "month": "August", "count": 61 }
    ],
    "id": "66abb3ba47a72962db72bdbe",
    "title": "The 1",
    "artist": "Taylor Swift",
    "album": "Folklore",
    "year": 2020
  },
  {
    "writers": ["Taylor Swift", "Jack Antonoff"],
    "plays": [
      { "month": "June", "count": 84 },
      { "month": "July", "count": 64 },
      { "month": "August", "count": 59 }
    ],
    "id": "66abb3ba47a72962db72bdd7",
    "title": "Cardigan",
    "artist": "Taylor Swift",
    "album": "Folklore",
    "year": 2020
  }
]
```

Get Songs by Artist

Endpoint: `/songs/artist/:artist`

Method: `GET`

Description: This endpoint returns a list of songs by a specified artist. It supports both partial and full name matching, meaning it will return songs by artists whose names start with the given prefix or fully match the provided artist name.

Request Parameter:

- `artist` (string, required): The name of the artist for which songs are to be fetched.

Response:

- **Status 200**: Returns a list of songs by the specified artist.
- **Status 400**: Returns an error if the artist parameter is missing.
- **Status 500**: Returns an error if there is an internal server error.

Example Request:

```
curl -X GET http://localhost:3000/api/songs/artist/sugar
```

Example Response:

```
[
    {
        "writers": [
            "Taylor Swift",
            "Patrick Monahan"
        ],
        "plays": [
            {
                "month": "June",
                "count": 19
            },
            {
                "month": "July",
                "count": 22
            },
            {
                "month": "August",
                "count": 7
            }
        ],
        "id": "66abb3ba47a72962db72bdc6",
```

```json
        "title": "Babe",
        "artist": "Sugarland\nfeaturing Taylor Swift",
        "album": "Bigger",
        "year": 2018
    }
]
```

Get Songs by Writer

Endpoint: `/songs/writer/:writer`

Method: `GET`

Description: This endpoint returns a list of songs written by a specified writer. It supports both partial and full name matching, meaning it will return songs written by writers whose names start with the given prefix or fully match the provided writer name.

Request Parameter:

- `writer` (string, required): The name of the writers for which songs are to be fetched.

Response:

- **Status 200**: Returns a list of songs by writers whose names start with the specified prefix or fully match the given writer name.
- **Status 400**: Returns an error if the writer parameter is missing.
- **Status 404**: Returns an error if no songs are found for the specified writer.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request

```
curl -X GET http://localhost:3000/api/songs/writer/max
```

Sample Response:

```
[
    {
        "writers": [
            "Taylor Swift",
            "Max Martin",
            "Shellback"
        ],
        "plays": [
            {
                "month": "June",
                "count": 27
            },
            {
                "month": "July",
                "count": 30
            },
            {
                "month": "August",
                "count": 32
            }
        ],
        "id": "66abb3ba47a72962db72bdbf",
        "title": "22",
        "artist": "Taylor Swift",
        "album": "Red",
        "year": 2012
    },
    {
        "writers": [
            "Taylor Swift",
            "Max Martin"
        ],
        "plays": [
            {
                "month": "June",
                "count": 78
            },
            {
                "month": "July",
                "count": 78
            },
            {
                "month": "August",
                "count": 97
            }
        ],
        "id": "66abb3ba47a72962db72bdc2",
        "title": "All You Had to Do Was Stay",
        "artist": "Taylor Swift",
        "album": "1989",
        "year": 2014
    },
    {
        "writers": [
            "Taylor Swift",
            "Max Martin",
            "Shellback"
        ],
        "plays": [
            {
                "month": "June",
                "count": 10
            },
            {
                "month": "July",
                "count": 75
            },
            {
                "month": "August",
                "count": 29
            }
        ],
        "id": "66abb3ba47a72962db72bdc8",
        "title": "Bad Blood (album version)",
        "artist": "Taylor Swift",
        "album": "1989",
        "year": 2014
    }
    // ...continued
]
```

Get Songs by Album

Endpoint: `/songs/album/:album`

Method: `GET`

Description: This endpoint returns a list of songs from a specified album. It supports both partial and full name matching, meaning it will return songs from albums whose names start with the given prefix or fully match the provided album name.

Request Parameter:

- `album` (string, required): The name of the album for which songs are to be fetched.

Response:

- **Status 200**: Returns a list of songs from the specified album.
- **Status 400**: Returns an error if the `album` parameter is not provided.
- **Status 404**: Returns an error if no songs are found for the given album name.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl -X GET http://localhost:3000/api/songs/album/spea
```

Sample Response:

```
[
    {
        "writers": [
            "Taylor Swift"
        ],
        "plays": [
            {
                "month": "June",
                "count": 50
            },
            {
                "month": "July",
                "count": 38
            },
            {
                "month": "August",
                "count": 63
            }
        ],
        "id": "66ab3f325944ca757a354bc8",
        "title": "Back to December",
        "artist": "Taylor Swift",
        "album": "Speak Now",
        "year": 2010
    },
    {
        "writers": [
            "Donna Weiss",
            "Jackie DeShannon"
        ],
        "plays": [
            {
                "month": "June",
                "count": 52
            },
            {
                "month": "July",
                "count": 56
            },
            {
                "month": "August",
                "count": 79
            }
        ],
        "id": "66ab3f325944ca757a354bcf",
        "title": "Bette Davis Eyes (live cover)",
        "artist": "Taylor Swift",
        "album": "Speak Now World Tour - Live",
        "year": 2011
    },
    {
        "writers": [
            "Taylor Swift"
        ],
        "plays": [
            {
                "month": "June",
                "count": 5
            },
            {
                "month": "July",
                "count": 110
            },
            {
                "month": "August",
                "count": 71
            }
        ],
        "id": "66ab3f325944ca757a354bd0",
        "title": "Better than Revenge",
        "artist": "Taylor Swift",
        "album": "Speak Now",
        "year": 2010
    }
    // ...continued
]
```

Get Songs by Month

Endpoint: `/songs/month/:month`

Method: `GET`

Description: This endpoint returns a list of songs that were played in a specified month. The month can be provided in any case (e.g., "august", "AUGUST", "aUg") and the system will recognize it correctly.

Request Parameter:

- `month` (string, required): The month for which songs are to be fetched.

Response:

- **Status 200**: Returns a list of songs played in the specified month.
- **Status 400**: Returns an error if the `month` parameter is not provided.
- **Status 404**: Returns an error if no songs are found for the given month.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl -X GET http://localhost:3000/api/songs/month/aUg
```

Sample Response:

```
[
    {
        "writers": [
            "Taylor Swift",
            "Aaron Dessner"
        ],
        "plays": [
            {
                "month": "August",
                "count": 61
            }
        ],
        "id": "66ab3f325944ca757a354bc8",
        "title": "The 1",
        "artist": "Taylor Swift",
        "album": "Folklore",
        "year": 2020
    },
    {
        "writers": [
            "Taylor Swift",
            "Max Martin",
            "Shellback"
        ],
        "plays": [
            {
                "month": "August",
                "count": 32
            }
        ],
        "id": "66ab3f325944ca757a354bcf",
        "title": "22",
        "artist": "Taylor Swift",
        "album": "Red",
        "year": 2012
    },
    {
        "writers": [
            "Taylor Swift",
            "Louis Bell",
            "Frank Dukes"
        ],
        "plays": [
            {
                "month": "August",
                "count": 87
            }
        ],

        "id": "66ab3f325944ca757a354bd0",
        "title": "Afterglow",
        "artist": "Taylor Swift",
        "album": "Lover",
        "year": 2019
    }
    // ...continued
]
```

Search Service

Health Check

Endpoint: `/search/health`

Method: `GET`

Description: This endpoint is used to check the health status of the Search microservice.

Request Parameters:

- None

Response:

- **Status 200**: Returns a status message confirming the search service is operational.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl -X GET http://localhost:3000/api/search/health
```

Sample Response:

```
{
  "status": "Welcome to Search Service!"
}
```

Advanced Search

Endpoint: `/search/advanced`

Method: `POST`

Description: This endpoint enables comprehensive searching within the SwiftCloud platform. Users can search for songs, artists, albums, or writers using a variety of criteria, including specific filters, sorting options, and pagination controls.

Request:

- **Body Parameters**:
    - `query` (string, required): The search term to partially or fully match a song title, artist, album, or writer.
    - `filters` (array): Array of filter criteria. Each filter must include. Each filter must include:
        - `field` (string, required): One of `title`, `artist`, `album`, `writers` `or year.`
        - `operator` (string, required): The operator to use for filtering. One of `eq` (equals), `gte` (greater than or equal to), `lte` (less than or equal to), or `contains` (partial match).
        - `value` (variable type, required if operator is `eq`, `gte`, or `lte`): Must be a number if operator is `gte` or `lte`. If `field` is `year`, `value` must be a number or numeric string within the range >= 1900 and <= current year, and `operator` cannot be `contains`.
    - `sort` (object): Sorting criteria with the following properties:
        - `field` (string): The field to sort by. Must be one of the fields listed above for filters.
        - `order` (string): The sorting order, either `asc` (ascending) or `desc` (descending).
    - `limit` (number, optional, default = 10): The maximum number of results to return.
    - `offset` (number, optional, default = 10): The starting point for the results.

Response:

- **Status 200**: Returns the search results that match the criteria.
- **Status 400**: Returns an error if the request parameters are invalid.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl --location 'http://localhost:3000/api/search/advanced' \
--header 'Content-Type: application/json' \
--data '{
  "query": "Max",
  "filters": [
    { "field": "year", "operator": "gte", "value": "2007" }
  ],
  "sort": { "field": "writers", "order": "asc" },
  "limit": 20,
  "offset": 0
}'
```

Sample Response:

```
{
    "results": [
        {
            "writers": [
                "Taylor Swift",
                "Max Martin",
                "Shellback"
            ],
            "plays": [
                {
                    "month": "June",
                    "count": 66
                },
                {
                    "month": "July",
                    "count": 16
                },
                {
                    "month": "August",
                    "count": 4
                }
            ],
            "id": "66acdb2d8ed575e50d508579",
            "title": "I Did Something Bad",
            "artist": "Taylor Swift",
            "album": "Reputation",
            "year": 2017
        },
        {
            "writers": [
                "Taylor Swift",
                "Max Martin",
                "Shellback"
            ],
            "plays": [
                {
                    "month": "June",
                    "count": 99
                },
                {
                    "month": "July",
                    "count": 17
                },
                {
                    "month": "August",
                    "count": 46
                }
            ],
            "id": "66acdb2d8ed575e50d5085d7",
            "title": "We Are Never Ever Getting Back Together",
            "artist": "Taylor Swift",
            "album": "Red",
            "year": 2012
        },
        // ...continued
    ],
    "total_results": 22
}
```

Autocomplete

Endpoint: `/search/autocomplete`

Method: `POST`

Description: This endpoint provides a list of suggestions that either begin with or exactly match the given query string. It is designed to empower typeahead functionality on the frontend. Each suggestion includes a value representing the matched entity and a type indicating the entity's category.

Request:

- **Body Parameters**:
    - `query` (string, required): The search term for which suggestions are needed.
    - `limit` (number, optional, default = 0): The maximum number of suggestions to return.

Response:

- **Status 200**: Returns a list of suggestions matching the query.
- **Status 400**: Returns an error if the request parameters are invalid.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl --location 'http://localhost:3000/api/search/autocomplete' \
--header 'Content-Type: application/json' \
--data '{
    "query": "tay",
    "limit": 5
}'
```

Sample Response:

```
{
    "suggestions": [
        {
            "value": "Taylor Swift",
            "type": "writer"
        },
        {
            "value": "Taylor Swift",
            "type": "album"
        },
        {
            "value": "Taylor Swift",
            "type": "artist"
        },
        {
            "value": "Taylor Swift\n(Deluxe edition)",
            "type": "album"
        },
        {
            "value": "Taylor Swift\nand Idris Elba",
            "type": "artist"
        }
    ]
}
```

Popularity Service

Health Check

Endpoint: `/popularity/health`

Method: `GET`

Description: This endpoint is used to check the health status of the Popularity microservice.

Request:

- No request parameters.

Response:

- **Status 200**: Returns a status message confirming the popularity service is operational.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl -X GET http://localhost:3000/api/popularity/health
```

Sample Response:

```
{
    "status": "Welcome to Popularity Service!"
}
```

Get Most Popular Songs Monthly

Endpoint: `/popularity/songs/most_monthly`

Method: `GET`

Description: Retrieves the most popular songs on a monthly basis. The results are sorted by play count in descending order.

Request Parameters:

- `limit` (number, optional, default = 50): The maximum number of results to return.
- `offset` (number, optional, default = 0): The number of results to skip before starting to collect the result set.

Response:

- **Status 200**: Returns a list of the most popular songs for each month.
- **Status 400**: Returns an error if the limit or offset parameters are not valid numbers.
- **Status 404**: Returns an error if no songs are found.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl --location
'http://localhost:3000/api/popularity/songs/most_monthly?limit=5&offset=
0'
```

Sample Response:

```
{
    "June": {
        "songs": [
            {
                "title": "Santa Baby (cover)",
                "artist": "Taylor Swift",
                "play_count": 110,
                "rank": 1
            },
            {
                "title": "Soon You'll Get Better",
                "artist": "Taylor Swift featuring Dixie Chicks",
                "play_count": 110,
                "rank": 2
            },
            {
                "title": "American Girl (cover)",
                "artist": "Taylor Swift",
                "play_count": 109,
                "rank": 3
            },
            {
                "title": "Miss Americana & the Heartbreak Prince",
                "artist": "Taylor Swift",
                "play_count": 109,
                "rank": 4
            },
            {
                "title": "The Way I Loved You",
                "artist": "Taylor Swift",
                "play_count": 107,
                "rank": 5
            }
        ]
    },
    "July": {
        "songs": [
            {
                "title": "Better than Revenge",
                "artist": "Taylor Swift",
                "play_count": 110,
                "rank": 1
            },
            {
                "title": "Christmases When You Were Mine",
                "artist": "Taylor Swift",
                "play_count": 110,
                "rank": 2
            },
            {
                "title": "Epiphany",
                "artist": "Taylor Swift",
                "play_count": 110,
                "rank": 3
            },
            {
                "title": "Lover (Remix)",
                "artist": "Taylor Swift featuring Shawn Mendes",
                "play_count": 110,
                "rank": 4
            },
            {
                "title": "Hold On (live cover)",
                "artist": "Jack Ingram featuring Taylor Swift",
                "play_count": 109,
                "rank": 5
            }
        ]
    },
    "August": {
        "songs": [
            {
                "title": "I Wish You Would",
                "artist": "Taylor Swift",
                "play_count": 110,
                "rank": 1
            },
            {
                "title": "Last Christmas (cover)",
                "artist": "Taylor Swift",
                "play_count": 110,
                "rank": 2
            },
            {
                "title": "Never Grow Up",
                "artist": "Taylor Swift",
                "play_count": 110,
                "rank": 3
            },
            {
                "title": "Style",
                "artist": "Taylor Swift",
                "play_count": 110,
                "rank": 4
            },
            {
                "title": "The Best Day",
                "artist": "Taylor Swift",
                "play_count": 110,
                "rank": 5
            }
        ]
    }
}
```

Get Most Popular Songs All Time

Endpoint: `/popularity/songs/most_all_time`

Method: `GET`

Description: Retrieves the most popular songs of all time. The results are sorted by play count in descending order.

Request Parameters:

- `limit` (number, optional, default = 50): The maximum number of results to return. Must be a number.
- `offset` (number, optional, default = 0): The number of results to skip before starting to collect the result set. Must be a number.

Response:

- **Status 200**: Returns a list of the most popular songs of all time.
- **Status 400**: Returns an error if the limit or offset parameters are not valid numbers.
- **Status 404**: Returns an error if no songs are found.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl --location
'http://localhost:3000/api/popularity/songs/most_all_time?limit=3&offset
=0'
```

Sample Response:

```
{
    "June": {
        "songs": [
            {
                "title": "Santa Baby (cover)",
                "artist": "Taylor Swift",
                "play_count": 110,
                "rank": 1
            },
            {
                "title": "Soon You'll Get Better",
                "artist": "Taylor Swift featuring Dixie Chicks",
                "play_count": 110,
                "rank": 2
            },
            {
                "title": "American Girl (cover)",
                "artist": "Taylor Swift",
                "play_count": 109,
                "rank": 3
            }
        ]
    },
    "July": {
        "songs": [
            {
                "title": "Better than Revenge",
                "artist": "Taylor Swift",
                "play_count": 110,
                "rank": 1
            },
            {
                "title": "Christmases When You Were Mine",
                "artist": "Taylor Swift",
                "play_count": 110,
                "rank": 2
            },
            {
                "title": "Epiphany",
                "artist": "Taylor Swift",
                "play_count": 110,
                "rank": 3
            }
        ]
    },
    "August": {
        "songs": [
            {
                "title": "I Wish You Would",
                "artist": "Taylor Swift",
                "play_count": 110,
                "rank": 1
            },
            {
                "title": "Last Christmas (cover)",
                "artist": "Taylor Swift",
                "play_count": 110,
                "rank": 2
            },
            {
                "title": "Never Grow Up",
                "artist": "Taylor Swift",
                "play_count": 110,
                "rank": 3
            }
        ]
    }
}
```

Get Song Popularity

Endpoint: `/popularity/songs/song`

Method: `GET`

Description: Fetches the overall rankings of song(s) whose titles contain the provided query string or fully match the provided title, aggregated across all available months in the input dataset.

Request Parameters:

- `title` (string, required): The title or partial title of the song to search for.

Response:

- **Status 200**: Returns a list of songs that match the searched title.
- **Status 400**: Returns an error if the title query parameter is not provided.
- **Status 404**: Returns an error if no songs are found.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl --location
'http://localhost:3000/api/popularity/songs/song?title=bet'
```

Sample Response:

```
{
    "rankings": [
        {
            "title": "Betty",
            "artist": "Taylor Swift",
            "play_count": 205,
            "rank": 45
        },
        {
            "title": "Bette Davis Eyes (live cover)",
            "artist": "Taylor Swift",
            "play_count": 187,
            "rank": 65
        },
        {
            "title": "Better than Revenge",
            "artist": "Taylor Swift",
            "play_count": 186,
            "rank": 66
        }
    ]
}
```

Alternative Sample Response (Single Match):

```
{
    "rankings": [
        {
            "title": "Two Is Better Than One",
            "artist": "Boys Like Girls\nfeaturing Taylor Swift",
            "play_count": 90,
            "rank": 153
        }
    ]
}
```

Get Most Popular Albums Monthly

Endpoint: `/popularity/albums/most_monthly`

Method: `GET`

Description: Retrieves the most popular albums for each month, based on the play count. The results are sorted by play count in descending order for each month.

Request Parameters:

- `limit` (number, optional, default = 50): The maximum number of albums to return for each month.
- `offset` (number, optional, default =: 0): The number of albums to skip before starting to collect the result set.

Response:

- **Status 200**: Returns the most popular albums for each month.
- **Status 400**: Returns an error if the limit or offset parameters are not valid numbers.
- **Status 404**: Returns an error if no albums are found.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl --location
'http://localhost:3000/api/popularity/albums/most_monthly?limit=3&offset
=0'
```

Sample Response:

```
{
    "June": {
        "albums": [
            {
                "name": "Lover",
                "artist": "Taylor Swift",
                "play_count": 1141,
                "rank": 1
            },
            {
                "name": "Fearless",
                "artist": "Taylor Swift",
                "play_count": 862,
                "rank": 2
            },
            {
                "name": "Folklore",
                "artist": "Taylor Swift",
                "play_count": 818,
                "rank": 3
            }
        ]
    },
    "July": {
        "albums": [
            {
                "name": "Lover",
                "artist": "Taylor Swift",
                "play_count": 1006,
                "rank": 1
            },
            {
                "name": "Speak Now",
                "artist": "Taylor Swift",
                "play_count": 905,
                "rank": 2
            },
            {
                "name": "Folklore",
                "artist": "Taylor Swift",
                "play_count": 894,
                "rank": 3
            }
        ]
    },
    "August": {
        "albums": [
            {
                "name": "Reputation",
                "artist": "Taylor Swift",
                "play_count": 1019,
                "rank": 1
            },
            {
                "name": "Folklore",
                "artist": "Taylor Swift",
                "play_count": 992,
                "rank": 2
            },
            {
                "name": "Lover",
                "artist": "Taylor Swift",
                "play_count": 983,
                "rank": 3
            }
        ]
    }
}
```

Get Most Popular Albums of All Time

Endpoint: `/popularity/albums/most_all_time`

Method: `GET`

Description: Fetches the overall rankings of albums whose

Request Parameters:

- `limit` (optional, default: 50): The maximum number of albums to return. Must be a number.
- `offset` (optional, default: 0): The number of albums to skip before starting to collect the result set. Must be a number.

Response:

- **Status 200**: Returns the most popular albums of all time.
- **Status 400**: Returns an error if the limit or offset parameters are not valid numbers.
- **Status 404**: Returns an error if no albums are found.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl --location
'http://localhost:3000/api/popularity/albums/most_all_time?limit=3&offset=0'
```

Sample Response:

```
{
    "albums": [
        {
            "name": "Lover",
            "artist": "Taylor Swift",
            "play_count": 3130,
            "rank": 1
        },
        {
            "name": "Folklore",
            "artist": "Taylor Swift",
            "play_count": 2704,
            "rank": 2
        },
        {
            "name": "Reputation",
            "artist": "Taylor Swift",
            "play_count": 2555,
            "rank": 3
        }
    ]
}
```

Get Album Popularity

Endpoint: `/popularity/albums/album`

Method: `GET`

Description: Retrieves the popularity rankings of album(s) that contain the provided query string or fully match the provided name, aggregated across all available months in the input dataset.

Request Parameters:

- `name` (string, required): The name or the partial name of the album to search for.

Response:

- **Status 200**: Returns the popularity rankings of the matching albums.
- **Status 400**: Returns an error if the name parameter is not provided.
- **Status 404**: Returns an error if no matching albums are found.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl --location
'http://localhost:3000/api/popularity/albums/album?name=spe'
```

Sample Response:

```
{
    "rankings": [
        {
            "name": "Speak Now",
            "artist": "Taylor Swift",
            "play_count": 2532,
            "rank": 4
        },
        {
            "name": "Speak Now World Tour - Live",
            "artist": "Taylor Swift",
            "play_count": 858,
            "rank": 10
        },
        {
            "name": "Speak Now\n(Deluxe edition)",
            "artist": "Taylor Swift",
            "play_count": 442,
            "rank": 16
        }
    ]
}
```

Get Most Popular Artists Monthly

Endpoint: `/popularity/artists/most_monthly`

Method: `GET`

Description: Retrieves the most popular artists for each month based on play counts.

Request Parameters:

- `limit` (optional, default = 0): The maximum number of artists to return per month. Must be a number.
- `offset` (optional, default = 0): The offset for pagination. Must be a number.

Response:

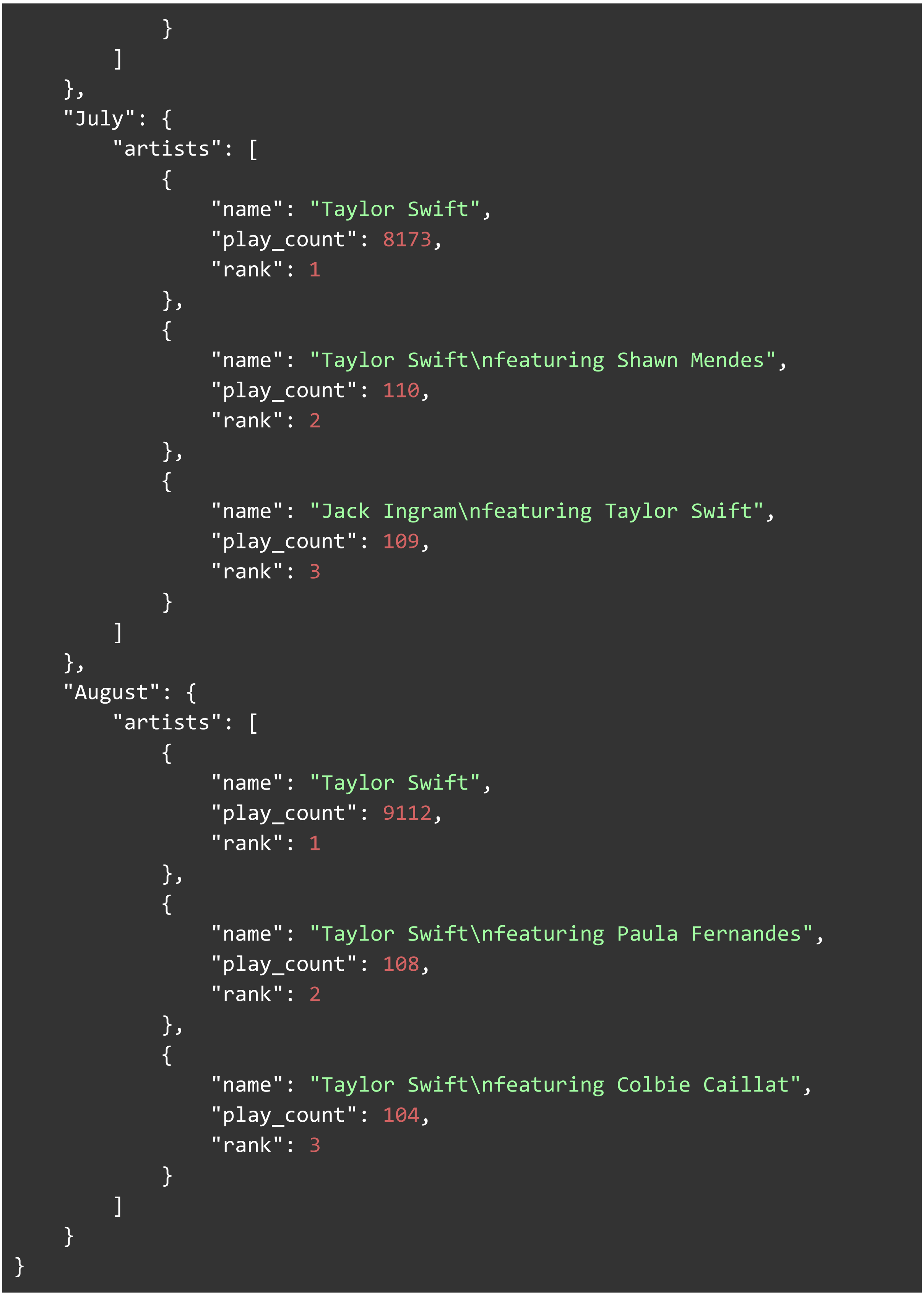
- **Status 200**: Returns the most popular artists for each month.
- **Status 400**: Returns an error if the limit or offset parameters are not numbers.
- **Status 404**: Returns an error if no artists are found.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl --location
'http://localhost:3000/api/popularity/artists/most_monthly?limit=3&offset=0'
```

Sample Response:

```
{
    "June": {
        "artists": [
            {
                "name": "Taylor Swift",
                "play_count": 8097,
                "rank": 1
            },
            {
                "name": "Taylor Swift\nfeaturing Dixie Chicks",
                "play_count": 110,
                "rank": 2
            },
            {
                "name": "B.o.B\nfeaturing Taylor Swift",
                "play_count": 104,
                "rank": 3
```



```
            }
        ]
    },
    "July": {
        "artists": [
            {
                "name": "Taylor Swift",
                "play_count": 8173,
                "rank": 1
            },
            {
                "name": "Taylor Swift\nfeaturing Shawn Mendes",
                "play_count": 110,
                "rank": 2
            },
            {
                "name": "Jack Ingram\nfeaturing Taylor Swift",
                "play_count": 109,
                "rank": 3
            }
        ]
    },
    "August": {
        "artists": [
            {
                "name": "Taylor Swift",
                "play_count": 9112,
                "rank": 1
            },
            {
                "name": "Taylor Swift\nfeaturing Paula Fernandes",
                "play_count": 108,
                "rank": 2
            },
            {
                "name": "Taylor Swift\nfeaturing Colbie Caillat",
                "play_count": 104,
                "rank": 3
            }
        ]
    }
}
```

Get Most Popular Artists of All Time

Endpoint: `/popularity/artists/most_all_time`

Method: `GET`

Description: Retrieves the most popular artists of all time based on play counts.

Request Parameters:

- `limit` (optional, default = 0): The maximum number of artists to return per month. Must be a number.
- `offset` (optional, default = 0): The offset for pagination. Must be a number.

Response:

- **Status 200**: Returns the most popular artists of all time.
- **Status 400**: Returns an error if the limit or offset parameters are not numbers.
- **Status 404**: Returns an error if no artists are found.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl --location
'http://localhost:3000/api/popularity/artists/most_all_time?limit=5&offset=0'
```

Sample Response:

```
{
    "artists": [
        {
            "name": "Taylor Swift",
            "play_count": 25382,
            "rank": 1
        },
        {
            "name": "B.o.B\nfeaturing Taylor Swift",
            "play_count": 274,
            "rank": 2
        },
        {
            "name": "Zayn and Taylor Swift",
            "play_count": 258,
            "rank": 3
        },
        {
            "name": "Taylor Swift\nfeaturing Paula Fernandes",
            "play_count": 243,
            "rank": 4
        },
        {
            "name": "Taylor Swift\nfeaturing Colbie Caillat",
            "play_count": 235,
            "rank": 5
        }

    ]
}
```

Get Artist Popularity

Endpoint: `/popularity/artists/artist`

Method: `GET`

Description: Retrieves the popularity of artists whose names contain the provided query string or fully match the provided name, aggregated across all available months in the input dataset.

Request Parameters:

- `name` (string, required): The name or the partial name of the artist to search for.

Response:

- **Status 200**: Returns the rankings of artists matching the provided name.
- **Status 400**: Returns an error if the name parameter is not provided.
- **Status 404**: Returns an error if no artists are found.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl --location
'http://localhost:3000/api/popularity/artists/artist?name=ed'
```

Sample Response:

```
{
    "rankings": [
        {
            "name": "Taylor Swift\nfeaturing Ed Sheeran",
            "play_count": 175,
            "rank": 11
        },
        {
            "name": "Taylor Swift\nfeaturing Ed Sheeran and Future",
            "play_count": 97,
            "rank": 18
        }
    ]
}
```

Trends Service

Health Check

Endpoint: `/trends/health`

Method: `GET`

Description: This endpoint is used to check the health status of the Trends microservice.

Request:

- No request parameters.

Response:

- **Status 200**: Returns a status message confirming the trends service is operational.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl -X GET http://localhost:3000/api/popularity/health
```

Sample Response:

```
{
    "status": "Welcome to Popularity Service!"
}
```

Get Overall Trends

Endpoint: `/trends/overall`

Method: `GET`

Description: Retrieves the overall trends of the top 10 artists with their top 10 songs. It also includes statistics such as total plays, average plays per song, and growth rate per month for both artists and songs where applicable.

Request Parameters: None

Response:

- **Status 200**: Returns the top 10 artists with their top 10 songs, along with overall statistics.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl --location 'http://localhost:3000/api/trends/overall'
```

Sample Response:

```
{
    "total_plays": 28694,
    "average_plays_per_song": 166.826,
    "top_artists": [
        {
            "name": "Taylor Swift",
            "total_plays": 25382,
            "average_plays_per_song": 55.662,
            "growth_rate_per_month": -0.056,
            "top_songs": [
                {
                    "title": "Style",
                    "growth_rate_per_month": -0.032
                },
                {
                    "title": "Beautiful Ghosts",
                    "growth_rate_per_month": 0
                },
                {
                    "title": "Peace",
                    "growth_rate_per_month": 0.025
                }
                // ...continued
            ]
        },
        {
            "name": "B.o.B featuring Taylor Swift",
            "total_plays": 274,
            "average_plays_per_song": 91.333,
            "growth_rate_per_month": 0.193,
            "top_songs": [
                {
                    "title": "Both of Us",
                    "growth_rate_per_month": 0.193
                }
                // ...continued
            ]
        },
        {
            "name": "Zayn and Taylor Swift",
            "total_plays": 258,
            "average_plays_per_song": 86,
            "growth_rate_per_month": 0.454,
            "top_songs": [
                {
                    "title": "I Don't Wanna Live Forever",
                    "growth_rate_per_month": 0.454
                }
                // ...continued
            ]
        }
        // ...continued
    ]
}
```

Get Trends for a Specific Period

Endpoint: `/trends/period`

Method: `GET`

Description: Retrieves the trends of the top 10 artists with their top 10 songs, for a specific period defined by the start and end months. It also includes statistics such as total plays, average plays per song, and growth rate per month for both artists and songs where applicable.

Request Parameters:

- `start_month` (number, required): The starting month of the period (range 1-12, where 1 represents January and 12 represents December).
- `end_month` (number, required): The ending month of the period (range 1-12, where 1 represents January and 12 represents December).

Response:

- **Status 200**: Returns the top 10 artists with their top 10 songs for the specified period, along with overall statistics.
- **Status 400**: Returns an error if the start month or end month is not provided or if the values are invalid.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl --location
'http://localhost:3000/api/trends/period?start_month=5&end_month=6'
```

Sample Response:

```
{
    "total_plays": 9111,
    "average_plays_per_song": 52.971,
    "top_artists": [
        {
            "name": "Taylor Swift",
            "total_plays": 8097,
            "average_plays_per_song": 53.27,
            "growth_rate_per_month": 0,
            "top_songs": [
                {
                    "title": "Santa Baby (cover)",
                    "growth_rate_per_month": 0
                },
                {
                    "title": "American Girl (cover)",
                    "growth_rate_per_month": 0
                },
                {
                    "title": "Miss Americana & the Heartbreak Prince",
                    "growth_rate_per_month": 0
                }
                // ...continued
            ]
        },
        {
            "name": "Taylor Swift featuring Dixie Chicks",
            "total_plays": 110,
            "average_plays_per_song": 110,
            "growth_rate_per_month": 0,
            "top_songs": [
                {
                    "title": "Soon You'll Get Better",
                    "growth_rate_per_month": 0
                }
                // ...continued
            ]
        },
        {
            "name": "B.o.B featuring Taylor Swift",
            "total_plays": 104,
            "average_plays_per_song": 104,
            "growth_rate_per_month": 0,
            "top_songs": [
                {
                    "title": "Both of Us",
                    "growth_rate_per_month": 0
                }
                // ...continued
            ]
        }
        // ...continued
    ]
}
```

Get Trending Songs

Endpoint: `/trends/songs`

Method: `GET`

Description: Retrieves the trending songs for the specified number of recent months. For instance, if the specified period is 2 months and the current month is August, the response will encompass data for June and July. The response also includes details such as the total plays, average plays per month, monthly growth rate, and other relevant information.

Request Parameters:

- `months` (number, required, default = 50): The number of months to look back for trending songs (1-12).
- `limit` (number, optional, default = 50): The maximum number of songs to retrieve.

Response:

- **Status 200**: Returns a list of trending songs for the specified period.
- **Status 400**: Returns an error if the `months` parameter is not provided, is not a number, or is outside the valid range. Also, returns an error if the `limit` parameter is not a number.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl --location
'http://localhost:3000/api/trends/songs?months=2&limit=5'
```

Sample Response:

```
[
    {
        "title": "Beautiful Ghosts",
        "artist": "Taylor Swift",
        "total_plays": 206,
        "growth_rate_per_month": 0.05999999865889549
    },
    {
        "title": "Christmas Tree Farm",
        "artist": "Taylor Swift",
        "total_plays": 204,
        "growth_rate_per_month": -0.03799999877810478
    },
    {
        "title": "I Don't Wanna Live Forever",
        "artist": "Zayn and Taylor Swift",
        "total_plays": 204,
        "growth_rate_per_month": -0.01899999938905239
    },
    {
        "title": "A Perfectly Good Heart",
        "artist": "Taylor Swift",
        "total_plays": 202,
        "growth_rate_per_month": 0.019999999552965164
    },
    {
        "title": "Hold On (live cover)",
        "artist": "Jack Ingram featuring Taylor Swift",
        "total_plays": 200,
        "growth_rate_per_month": 0.1979999989271164
    }
]
```

Get Trending Artists

Endpoint: `/trends/artists`

Method: `GET`

Description: Retrieves the trending artists for the specified number of recent months. For instance, if the specified period is 2 months and the current month is August, the response will encompass data for June and July. The response also includes details such as the total plays, average plays per month, monthly growth rate, and other relevant information.

Request Parameters:

- `months` (number, required): The number of months to look back for trending artists (1-12).
- `limit` (number, optional, default = 50): The maximum number of artists to retrieve.

Response:

- **Status 200**: Returns a list of trending artists for the specified period.
- **Status 400**: Returns an error if the `months` parameter is not provided, is not a number, or is outside the valid range. Also, returns an error if the `limit` parameter is not a number.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request:

```
curl --location
'http://localhost:3000/api/trends/artists?months=2&limit=3'
```

Sample Response:

```
[
    {
        "name": "Taylor Swift",
        "total_plays": 16270,
        "growth_rate_per_month": 0.009
    },
    {
        "name": "Zayn and Taylor Swift",
        "total_plays": 204,
        "growth_rate_per_month": -0.019
    },
    {
        "name": "Jack Ingram featuring Taylor Swift",
        "total_plays": 200,
        "growth_rate_per_month": 0.198
    },
    {
        "name": "B.o.B\nfeaturing Taylor Swift",
        "total_plays": 199,
        "growth_rate_per_month": -0.0870
    },
    {
        "name": "Taylor Swift\nfeaturing Dixie Chicks",
        "total_plays": 188,
        "growth_rate_per_month": -0.291
    }

]
```

Get Trending Albums

Endpoint: `/trends/albums`

Method: `GET`

Description: Retrieves the trending albums for the specified number of recent months. For instance, if the specified period is 2 months and the current month is August, the response will encompass data for June and July. The response also includes details such as the total plays, average plays per month, monthly growth rate, and other relevant information.

Request Parameters:

- `months` (number, required): The number of months to look back for trending albums (1-12).
- `limit` (number, optional, default = 50): The maximum number of albums to retrieve.

Response:

- **Status 200**: Returns a list of trending albums for the specified period.
- **Status 400**: Returns an error if the `months` parameter is not provided, is not a number, or is outside the valid range. Also, returns an error if the `limit` parameter is not a number.
- **Status 500**: Returns an error if there is an internal server error.

Sample Request

```
curl --location
'http://localhost:3000/api/trends/albums?months=2&limit=5'
```

Sample Response:

```
[
    {
        "name": "Lover",
        "artist": "Taylor Swift",
        "total_plays": 2147,
        "growth_rate_per_month": -0.11800000071525574
    },
    {
        "name": "Folklore",
        "artist": "Taylor Swift",
        "total_plays": 1712,
        "growth_rate_per_month": 0.09300000220537186
    },
    {
        "name": "Fearless",
        "artist": "Taylor Swift",
        "total_plays": 1661,
        "growth_rate_per_month": -0.0729999989271164
    },
    {
        "name": "Speak Now",
        "artist": "Taylor Swift",
        "total_plays": 1575,
        "growth_rate_per_month": 0.35100001096725464
    },
    {
        "name": "Reputation",
        "artist": "Taylor Swift",
        "total_plays": 1536,
        "growth_rate_per_month": -0.05299999937415123
    }
]
```