Api Documentation

Overview

Api Documentation

Version information

Version : 1.0

License information

License : Apache 2.0
License URL : http://www.apache.org/licenses/LICENSE-2.0
Terms of service : urn:tos

URI scheme

Host : localhost:8080
BasePath : /

Tags

- auth-controller : Auth Controller
- basic-error-controller : Basic Error Controller
- operation-handler : Operation Handler
- user-controller : User Controller
- web-mvc-links-handler : Web Mvc Links Handler

Paths

links

```
GET /actuator
```

Responses

HTTP Code	Description	Schema
200	OK	< string, < string, Link > map > map
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`
- `application/vnd.spring-boot.actuator.v2+json`

Tags

- web-mvc-links-handler

handle

```
GET /actuator/health
```

Parameters

Type	Name	Description	Schema
Body	**body** *optional*	body	< string, string > map

Responses

HTTP Code	Description	Schema
200	OK	object
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`
- `application/vnd.spring-boot.actuator.v2+json`

Tags

- operation-handler

handle

```
GET /actuator/health/{component}
```

Parameters

Type	Name	Description	Schema
Body	**body** *optional*	body	< string, string > map

Responses

HTTP Code	Description	Schema
200	OK	object
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`
- `application/vnd.spring-boot.actuator.v2+json`

Tags

- operation-handler

handle

```
GET /actuator/health/{component}/{instance}
```

Parameters

Type	Name	Description	Schema
Body	**body** *optional*	body	< string, string > map

Responses

HTTP Code	Description	Schema
200	OK	object
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`
- `application/vnd.spring-boot.actuator.v2+json`

Tags

- operation-handler

handle

```
GET /actuator/info
```

Parameters

Type	Name	Description	Schema
Body	**body** *optional*	body	< string, string > map

Responses

HTTP Code	Description	Schema
200	OK	object
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`
- `application/vnd.spring-boot.actuator.v2+json`

Tags

- operation-handler

authenticateUser

```
POST /auth/login
```

Parameters

Type	Name	Description	Schema
Body	**loginRequest** *required*	loginRequest	LoginRequest

Responses

HTTP Code	Description	Schema
200	OK	object
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

HTTP Code	Description	Schema
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- auth-controller

registerUser

```
POST /auth/signup
```

Parameters

Type	Name	Description	Schema
Body	**signUpRequest** *required*	signUpRequest	SignUpRequest

Responses

HTTP Code	Description	Schema
200	OK	object
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- auth-controller

error

```
POST /error
```

Responses

HTTP Code	Description	Schema
200	OK	< string, object > map
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- basic-error-controller

error

```
GET /error
```

Responses

HTTP Code	Description	Schema
200	OK	< string, object > map
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`

Tags

- basic-error-controller

error

```
PUT /error
```

Responses

HTTP Code	Description	Schema
200	OK	< string, object > map
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

HTTP Code	Description	Schema
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- basic-error-controller

error

```
DELETE /error
```

Responses

HTTP Code	Description	Schema
200	OK	< string, object > map
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- `*/*`

Tags

- basic-error-controller

error

```
PATCH /error
```

Responses

HTTP Code	Description	Schema
200	OK	< string, object > map
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- basic-error-controller

error

```
HEAD /error
```

Responses

HTTP Code	Description	Schema
200	OK	< string, object > map
204	No Content	No Content
401	Unauthorized	No Content

HTTP Code	Description	Schema
403	Forbidden	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- basic-error-controller

error

```
OPTIONS /error
```

Responses

HTTP Code	Description	Schema
200	OK	< string, object > map
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- basic-error-controller

Create User

```
POST /user
```

Parameters

Type	Name	Description	Schema
Body	**korisnik** *required*	korisnik	UserRequest

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- user-controller

Update User By Id

```
PUT /user
```

Parameters

Type	Name	Description	Schema
Body	**user** *required*	user	User

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`
- `application/json`

Tags

- user-controller

Get All Users

```
GET /user/all
```

Responses

HTTP Code	Description	Schema
200	OK	< User > array

HTTP Code	Description	Schema
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`
- `application/json`

Tags

- user-controller

Get User By Id

```
GET /user/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	User
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`

- `application/json`

Tags

- user-controller

Delete User By Id

```
DELETE /user/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	string
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- `*/*`
- `application/json`

Tags

- user-controller

Definitions

Link

Name	Schema
href *optional*	string
templated *optional*	boolean

LoginRequest

Name	Schema
email *optional*	string
password *optional*	string

Map«string,Link»

Type : < string, Link > map

ModelAndView

Name	Schema
empty *optional*	boolean
model *optional*	object
modelMap *optional*	< string, object > map
reference *optional*	boolean

Name	Schema
status *optional*	enum (100 CONTINUE, 101 SWITCHING_PROTOCOLS, 102 PROCESSING, 103 CHECKPOINT, 200 OK, 201 CREATED, 202 ACCEPTED, 203 NON_AUTHORITATIVE_INFORMATION, 204 NO_CONTENT, 205 RESET_CONTENT, 206 PARTIAL_CONTENT, 207 MULTI_STATUS, 208 ALREADY_REPORTED, 226 IM_USED, 300 MULTIPLE_CHOICES, 301 MOVED_PERMANENTLY, 302 FOUND, 302 MOVED_TEMPORARILY, 303 SEE_OTHER, 304 NOT_MODIFIED, 305 USE_PROXY, 307 TEMPORARY_REDIRECT, 308 PERMANENT_REDIRECT, 400 BAD_REQUEST, 401 UNAUTHORIZED, 402 PAYMENT_REQUIRED, 403 FORBIDDEN, 404 NOT_FOUND, 405 METHOD_NOT_ALLOWED, 406 NOT_ACCEPTABLE, 407 PROXY_AUTHENTICATION_REQUIRED, 408 REQUEST_TIMEOUT, 409 CONFLICT, 410 GONE, 411 LENGTH_REQUIRED, 412 PRECONDITION_FAILED, 413 PAYLOAD_TOO_LARGE, 413 REQUEST_ENTITY_TOO_LARGE, 414 URI_TOO_LONG, 414 REQUEST_URI_TOO_LONG, 415 UNSUPPORTED_MEDIA_TYPE, 416 REQUESTED_RANGE_NOT_SATISFIABLE, 417 EXPECTATION_FAILED, 418 I_AM_A_TEAPOT, 419 INSUFFICIENT_SPACE_ON_RESOURCE, 420 METHOD_FAILURE, 421 DESTINATION_LOCKED, 422 UNPROCESSABLE_ENTITY, 423 LOCKED, 424 FAILED_DEPENDENCY, 426 UPGRADE_REQUIRED, 428 PRECONDITION_REQUIRED, 429 TOO_MANY_REQUESTS, 431 REQUEST_HEADER_FIELDS_TOO_LARGE, 451 UNAVAILABLE_FOR_LEGAL_REASONS, 500 INTERNAL_SERVER_ERROR, 501 NOT_IMPLEMENTED, 502 BAD_GATEWAY, 503 SERVICE_UNAVAILABLE, 504 GATEWAY_TIMEOUT, 505 HTTP_VERSION_NOT_SUPPORTED, 506 VARIANT_ALSO_NEGOTIATES, 507 INSUFFICIENT_STORAGE, 508 LOOP_DETECTED, 509 BANDWIDTH_LIMIT_EXCEEDED, 510 NOT_EXTENDED, 511 NETWORK_AUTHENTICATION_REQUIRED)
view *optional*	View

Name	Schema
viewName *optional*	string

SignUpRequest

Name	Schema
email *optional*	string
name *optional*	string
password *optional*	string
surname *optional*	string

User

Name	Schema
email *optional*	string
id *optional*	integer (int64)
name *optional*	string
password *optional*	string
role *optional*	string
surname *optional*	string

UserRequest

Name	Schema
email *optional*	string
name *optional*	string
password *optional*	string
surname *optional*	string
username *optional*	string

View

Name	Schema
contentType *optional*	string

Api Documentation

Overview

Api Documentation

Version information

Version : 1.0

License information

License : Apache 2.0
License URL : http://www.apache.org/licenses/LICENSE-2.0
Terms of service : urn:tos

URI scheme

Host : localhost:8081
BasePath : /

Tags

- album-controller : Album Controller
- basic-error-controller : Basic Error Controller
- genre-controller : Genre Controller
- operation-handler : Operation Handler
- performer-controller : Performer Controller
- playlist-controller : Playlist Controller
- song-controller : Song Controller
- song-playlist-controller : Song Playlist Controller
- web-mvc-links-handler : Web Mvc Links Handler

Paths

links

```
GET /actuator
```

Responses

HTTP Code	Description	Schema
200	OK	< string, < string, Link > map > map
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`
- `application/vnd.spring-boot.actuator.v2+json`

Tags

- web-mvc-links-handler

handle

```
GET /actuator/health
```

Parameters

Type	Name	Description	Schema
Body	**body** *optional*	body	< string, string > map

Responses

HTTP Code	Description	Schema
200	OK	object
401	Unauthorized	No Content
403	Forbidden	No Content

HTTP Code	Description	Schema
404	Not Found	No Content

Produces

- `application/json`
- `application/vnd.spring-boot.actuator.v2+json`

Tags

- operation-handler

handle

```
GET /actuator/health/{component}
```

Parameters

Type	Name	Description	Schema
Body	**body** *optional*	body	< string, string > map

Responses

HTTP Code	Description	Schema
200	OK	object
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`
- `application/vnd.spring-boot.actuator.v2+json`

Tags

- operation-handler

handle

```
GET /actuator/health/{component}/{instance}
```

Parameters

Type	Name	Description	Schema
Body	**body** *optional*	body	< string, string > map

Responses

HTTP Code	Description	Schema
200	OK	object
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`
- `application/vnd.spring-boot.actuator.v2+json`

Tags

- operation-handler

handle

```
GET /actuator/info
```

Parameters

Type	Name	Description	Schema
Body	**body** *optional*	body	< string, string > map

Responses

HTTP Code	Description	Schema
200	OK	object
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `application/json`
- `application/vnd.spring-boot.actuator.v2+json`

Tags

- operation-handler

create

```
POST /album
```

Parameters

Type	Name	Description	Schema
Body	**albumRequest** *required*	albumRequest	AlbumRequest

Responses

HTTP Code	Description	Schema
200	OK	string

HTTP Code	Description	Schema
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- album-controller

Get All Albums

```
GET /album/all
```

Responses

HTTP Code	Description	Schema
200	OK	< Album > array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`
- `application/json`

Tags

- album-controller

findAlbumById

```
GET /album/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	Album
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`

Tags

- album-controller

update

```
PUT /album/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)
Body	**album** *required*	album	AlbumRequest

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- album-controller

deleteAlbum

```
DELETE /album/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	string
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- `*/*`

Tags

- album-controller

error

```
POST /error
```

Responses

HTTP Code	Description	Schema
200	OK	< string, object > map
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- basic-error-controller

error

```
GET /error
```

Responses

HTTP Code	Description	Schema
200	OK	< string, object > map
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`

Tags

- basic-error-controller

error

```
PUT /error
```

Responses

HTTP Code	Description	Schema
200	OK	< string, object > map
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- basic-error-controller

error

```
DELETE /error
```

Responses

HTTP Code	Description	Schema
200	OK	< string, object > map
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- `*/*`

Tags

- basic-error-controller

error

```
PATCH /error
```

Responses

HTTP Code	Description	Schema
200	OK	< string, object > map
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- basic-error-controller

error

```
HEAD /error
```

Responses

HTTP Code	Description	Schema
200	OK	< string, object > map
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- basic-error-controller

error

```
OPTIONS /error
```

Responses

HTTP Code	Description	Schema
200	OK	< string, object > map
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- basic-error-controller

create

```
POST /genre
```

Parameters

Type	Name	Description	Schema
Body	**genreRequest** *required*	genreRequest	GenreRequest

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- genre-controller

Get All Genres

```
GET /genre/all
```

Responses

HTTP Code	Description	Schema
200	OK	< Genre > array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`
- `application/json`

Tags

- genre-controller

findGenreById

```
GET /genre/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	Genre

HTTP Code	Description	Schema
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`

Tags

- genre-controller

update

```
PUT /genre/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)
Body	**genre** *required*	genre	GenreRequest

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- genre-controller

deleteGenre

```
DELETE /genre/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	string
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- `*/*`

Tags

- genre-controller

create

```
POST /performer
```

Parameters

Type	Name	Description	Schema
Body	**performerRequest** *required*	performerRequest	PerformerRequest

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- performer-controller

Get All Performers

```
GET /performer/all
```

Responses

HTTP Code	Description	Schema
200	OK	< Performer > array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`
- `application/json`

Tags

- performer-controller

findPerformerById

```
GET /performer/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	Performer
401	Unauthorized	No Content
403	Forbidden	No Content

HTTP Code	Description	Schema
404	Not Found	No Content

Produces

- `*/*`

Tags

- performer-controller

update

```
PUT /performer/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)
Body	**performer** *required*	performer	PerformerRequest

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- performer-controller

deletePerformer

```
DELETE /performer/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	string
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- `*/*`

Tags

- performer-controller

create

```
POST /playlist
```

Parameters

Type	Name	Description	Schema
Body	**playlistRequest** *required*	playlistRequest	PlaylistRequest

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- playlist-controller

Get All Playlists

```
GET /playlist/all
```

Responses

HTTP Code	Description	Schema
200	OK	< Playlist > array

HTTP Code	Description	Schema
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`
- `application/json`

Tags

- playlist-controller

findplaylistById

```
GET /playlist/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	Playlist
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`

Tags

- playlist-controller

update

```
PUT /playlist/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)
Body	**playlist** *required*	playlist	PlaylistRequest

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- playlist-controller

deleteplaylist

```
DELETE /playlist/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	string
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- `*/*`

Tags

- playlist-controller

create

```
POST /song
```

Parameters

Type	Name	Description	Schema
Body	**songRequest** *required*	songRequest	SongRequest

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- song-controller

create

```
POST /song-playlist
```

Parameters

Type	Name	Description	Schema
Body	**songPlaylistRequest** *required*	songPlaylistRequest	SongPlaylistRequest

Responses

HTTP Code	Description	Schema
200	OK	string

HTTP Code	Description	Schema
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- song-playlist-controller

Get All SongPlaylists

```
GET /song-playlist/all
```

Responses

HTTP Code	Description	Schema
200	OK	< SongPlaylist > array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`
- `application/json`

Tags

- song-playlist-controller

findSongPlaylistById

```
GET /song-playlist/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	SongPlaylist
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`

Tags

- song-playlist-controller

update

```
PUT /song-playlist/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)
Body	**songPlaylist** *required*	songPlaylist	SongPlaylistRequest

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- song-playlist-controller

deleteSongPlaylist

```
DELETE /song-playlist/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	string
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- `*/*`

Tags

- song-playlist-controller

Get All Songs

```
GET /song/all
```

Responses

HTTP Code	Description	Schema
200	OK	< Song > array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`
- `application/json`

Tags

- song-controller

findSongById

```
GET /song/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	Song
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`

Tags

- song-controller

update

```
PUT /song/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)
Body	**song** *required*	song	SongRequest

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- song-controller

deleteSong

```
DELETE /song/{id}
```

Parameters

Type	Name	Description	Schema
Path	**id** *required*	id	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	string
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Produces

- `*/*`

Tags

- song-controller

Definitions

Album

Name	Schema
id *optional*	integer (int64)
performer *optional*	integer (int64)
released *optional*	string (date-time)
title *optional*	string

AlbumRequest

Name	Schema
performer *optional*	integer (int64)

Name	Schema
released *optional*	string (date-time)
title *optional*	string

Genre

Name	Schema
id *optional*	integer (int64)
title *optional*	string

GenreRequest

Name	Schema
id *optional*	integer (int64)
title *optional*	string

Link

Name	Schema
href *optional*	string
templated *optional*	boolean

Map«string,Link»

Type : < string, Link > map

ModelAndView

Name	Schema
empty *optional*	boolean
model *optional*	object
modelMap *optional*	< string, object > map
reference *optional*	boolean

Name	Schema
status *optional*	enum (100 CONTINUE, 101 SWITCHING_PROTOCOLS, 102 PROCESSING, 103 CHECKPOINT, 200 OK, 201 CREATED, 202 ACCEPTED, 203 NON_AUTHORITATIVE_INFORMATION, 204 NO_CONTENT, 205 RESET_CONTENT, 206 PARTIAL_CONTENT, 207 MULTI_STATUS, 208 ALREADY_REPORTED, 226 IM_USED, 300 MULTIPLE_CHOICES, 301 MOVED_PERMANENTLY, 302 FOUND, 302 MOVED_TEMPORARILY, 303 SEE_OTHER, 304 NOT_MODIFIED, 305 USE_PROXY, 307 TEMPORARY_REDIRECT, 308 PERMANENT_REDIRECT, 400 BAD_REQUEST, 401 UNAUTHORIZED, 402 PAYMENT_REQUIRED, 403 FORBIDDEN, 404 NOT_FOUND, 405 METHOD_NOT_ALLOWED, 406 NOT_ACCEPTABLE, 407 PROXY_AUTHENTICATION_REQUIRED, 408 REQUEST_TIMEOUT, 409 CONFLICT, 410 GONE, 411 LENGTH_REQUIRED, 412 PRECONDITION_FAILED, 413 PAYLOAD_TOO_LARGE, 413 REQUEST_ENTITY_TOO_LARGE, 414 URI_TOO_LONG, 414 REQUEST_URI_TOO_LONG, 415 UNSUPPORTED_MEDIA_TYPE, 416 REQUESTED_RANGE_NOT_SATISFIABLE, 417 EXPECTATION_FAILED, 418 I_AM_A_TEAPOT, 419 INSUFFICIENT_SPACE_ON_RESOURCE, 420 METHOD_FAILURE, 421 DESTINATION_LOCKED, 422 UNPROCESSABLE_ENTITY, 423 LOCKED, 424 FAILED_DEPENDENCY, 426 UPGRADE_REQUIRED, 428 PRECONDITION_REQUIRED, 429 TOO_MANY_REQUESTS, 431 REQUEST_HEADER_FIELDS_TOO_LARGE, 451 UNAVAILABLE_FOR_LEGAL_REASONS, 500 INTERNAL_SERVER_ERROR, 501 NOT_IMPLEMENTED, 502 BAD_GATEWAY, 503 SERVICE_UNAVAILABLE, 504 GATEWAY_TIMEOUT, 505 HTTP_VERSION_NOT_SUPPORTED, 506 VARIANT_ALSO_NEGOTIATES, 507 INSUFFICIENT_STORAGE, 508 LOOP_DETECTED, 509 BANDWIDTH_LIMIT_EXCEEDED, 510 NOT_EXTENDED, 511 NETWORK_AUTHENTICATION_REQUIRED)
view *optional*	View

Name	Schema
viewName *optional*	string

Performer

Name	Schema
id *optional*	integer (int64)
info *optional*	string
name *optional*	string

PerformerRequest

Name	Schema
info *optional*	string
name *optional*	string

Playlist

Name	Schema
created *optional*	string (date-time)
createdBy *optional*	integer (int64)
id *optional*	integer (int64)
title *optional*	string

PlaylistRequest

Name	Schema
created *optional*	string (date-time)
createdBy *optional*	integer (int64)
title *optional*	string

Song

Name	Schema
album *optional*	integer (int64)
audio *optional*	integer (int64)
genre *optional*	integer (int64)
id *optional*	integer (int64)
title *optional*	string

SongPlaylist

Name	Schema
id *optional*	integer (int64)
playlist *optional*	integer (int64)
song *optional*	integer (int64)

SongPlaylistRequest

Name	Schema
playlist *optional*	integer (int64)
song *optional*	integer (int64)

SongRequest

Name	Schema
album *optional*	integer (int64)
audio *optional*	integer (int64)
genre *optional*	integer (int64)
playlistId *optional*	integer (int64)
title *optional*	string

View

Name	Schema
contentType *optional*	string