RESTful API Design

-returns in JSON format

Models:

**-Topics**

-Name

-Basic Image relating to Topic

-Link(s) to Book(s)

-Link(s) to Movie(s)

-Links to Music

/topics

GET Topics:

GET /topics – returns list of all topics

Response:

Topics - array

GET Topics Image:

GET /topics/image – returns image for this topic

**-Books**

-Name

-Author

-Description

-Year of Release

-Image of Cover

-Link(s) to Book's Topic(s)

-Link(s) to Similar Movies

-Link(s) to Similar Music

/books

/books/<id>

-every book needs an ID

Query String:

api\_key – string, required

GET Books:

GET /books – returns list of books

Response:

Books – array

{

}

path parameters for the following HTTP requests:

book\_id -> integer, required

GET Book Details:

GET books/<id> - returns details about specific book

Response:

About – array

{

id – integer

name – string

author – string

description – string

publication\_year – integer

book\_cover - string

}

GET Book Name: is this needed??

GET books/<id>/name - returns name of book

Response:

Id – integer

Name - string

GET Book Author:

GET books/<id>/author – returns author of the book

Response:

Id – integer

Author - string

GET Book Description:

GET books/<id>/description – returns description for the book

Response:

Id – integer

Description – string

GET Book Publication Year:

GET books/<id>/publication\_year – returns publication year for the book

Response:

Id – integer

Publication\_year - integer

GET Book Cover:

GET books/<id>/cover – returns book cover

Response:

Id – integer

Image - string

**-Movies**

-Name

-Description

-Year of Release

-Trailer of Movie (if available)

-Image of Movie Poster (if trailer not available)

-Link(s) to Movie's Topic(s)

-Link(s) to Similar Book(s)

-Link(s) to Similar Music

resources: movie name, movie overview, release year, trailer, movie poster, links to related media

-need an ID for each movie so when the client calls to get movies, they'll get the ID for each movie and then they can use that ID to call /movies/<id> to get more details

/movies - list of movies

/movies/<id> - returns the details about a specific movie

Query String:

api\_key -> string, required

GET Movies:

GET /movies - returns list of movies

Response:

Movies – array

{

Id – integer

name – string

description – string

release\_year – integer

trailer – string, optional

image – string, optional

}

-Path parameters for the following HTTP requests:

movie\_id -> Integer, required

GET Movie Details:

GET /movies/<id> - returns details for this movie

Response:

Id – integer

name – string

description – string

release\_year – integer

trailer – string, optional

image – string, optional

links – array ??? do we need to return these links??

{

}

GET Movie Description:

GET /movies/<id>/description - returns description for a specific movie

Response:

Id - integer

Description - string

GET Movie Release Year:

GET /movies/<id>/release\_year

Response:

Id – integer

Release\_year - integer

GET Movie Trailer (if available):

GET /movies/<id>/trailer – returns link to movie trailer if available

Response:

Id - integer

Site – string

Size – integer (360, 480, 720, 1080)

404 error message: return status\_message (string) saying there’s no movie trailer

GET Movie Poster (if no trailer):

GET /movies/<id>/image – returns link to image of movie poster

Response:

Id – integer

Image\_link - string

404 error message: return status\_message (string) saying there’s no movie poster

**-Music**

-Name of Song

-Artist

-Album

-Year of Release

-Song Preview Player

-Link(s) to Song's Topic(s)

-Link(s) to Similar Movie(s)

-Link(s) to Similar Book(s)

-each song needs an ID

Query String:

api\_key – string, required

GET Music:

GET /music – returns list of all songs

Response:

Songs – array

{

id – integer

name – string

artist – string

album – string

year - integer

}

path parameters for following HTTP requests:

song\_id – integer

GET Song Details:

GET/music/<id> - returns info about this song

Response:

Id – integer

name – string

artist – string

album – string

year – integer

GET Song Artist:

GET/music/<id>/artist – returns artist for this song

Response:

Id – integer

Artist – string

GET Song Album:

GET/music/<id>/album – returns album this song is on

Response:

Id – integer

Album - string

GET Song Release Year:

GET/music/<id>/release\_year – returns year this song was released

Response:

Id – integer

Release\_year - integer

GET Song Preview:

GET/music/<id>/preview – returns preview of this song

Response:

Id – integer

Preview – string (link to preview)