

# Ideas

- Barcode scanner / data

## Initial Tasks

Julian Set up postgres DB and H2.

James - set up github repo and project / java start code.

Malcolm - test behavior of annotation libraries when things like arguments are missed.

Malcolm - status codes for error messages “official” where possible

Everyone - see if you can find a file listing albums/artist/year/genre.

## API structure

Base URL /api/v1/album...

→ get all albums in stock

/instock

return: array of albums [{id, title, artist, year, genre, stock}, ... ]

error status:

→ get all albums by a given artist

GET /artist

parameter: artist

return: array of albums [{id, title, artist, year, genre, stock}, ... ]

error status: 404 unknown artist

→ get all albums by a given release year

GET /year

parameter: year

return: array of albums [{id, title, artist, year, genre, stock}, ... ]

error status: invalid year (not a number, out of range)

→ get all albums by a given genre

GET /genre

Parameter: genre

return: array of albums [{id, title, artist, year, genre, stock}, ... ]

Error: 404 - unknown genre

→ get album information by album title

GET /title

parameter: title

return: array of albums [{id, title, artist, year, genre, stock}, ... ]

error: missing or invalid title string

→ add new album into the database

POST /

parameters: title, artist, year, genre.

return: album {id, title, artist, year, genre, stock}

error: if (artist, year and title) match another record - return error.

→ update album details

PUT /

parameters: id, title, artist, year, genre.

error: if (artist, year and title) match a record other than the one being edited - return error.

error: unknown id

→ update stock amounts of a particular album

PATCH /stock

parameters: id, stock

error: unknown id

error: stock  $\geq$  0

→ delete albums from the inventory

DELETE /

parameters: id

error: unknown id

# Service Layer Functionality

getAlbumsInStock()

- returns list of all albums in stock
- findByStockGreaterThan(0)

getAlbumsByArtist( String artist)

- Returns a list of all albums by artist
- findAllAlbumsByArtistContainingIgnoreCase(String artist)

getAlbumByReleaseYear( int year)

- Returns list of all albums for given year
- findAllAlbumsByReleaseYear( int year)

getAlbumsByGenre( Genre genre)

- Returns a list of all albums by genre
- findAllAlbumsByGenre( Genre genre)

getAlbumsByTitle(String title)

- Returns list of albums with the given title
- findAllAlbumsByTitleContainingIgnoreCase( String title)

insertAlbum( Album album )

- Calls repo.save(album)

updateAlbumById( Album album)

- album = findById(id)
- repo.save(album)

updateAlbumStockById( long id, int stock )

- album = findById(id)
- repo.save(album)

deleteAlbumById( long id )

- album = findById
- repo.delete(album)

# Database Tables

## Albums

ID	integer - auto-increment
title	string
artist	string
year	integer
genre	integer/string
stock	integer