

# Project 1 API Walkthrough

Aaron Milner

## Books

- Purpose: Keep a collection of books storing relevant information such as name, author, and year.

## Endpoints:

- GET /books: Returns all books
- GET /books/{id}: Returns a specific book based on the id
- POST /books: Creates a new book in the database
- PUT/PATCH /books/{id}: Modifies an already existing book based on the id
- DELETE /books/{id}: Removes a book from the database based on the id

## Contractors

- Purpose: Keep a collection of contractors from various companies and keep a collection of their location in the business structure.

## Endpoints:

- GET /contractors: Retrieves a list of all contractors
- GET /contractors/{id}: Returns a specific contractor based on the id
- POST /contractors: Creates a new contractor in the database
- PUT/PATCH /contractors/{id}: Updates the record of an existing contractor based on the id
- DELETE /contractors/{id}: Deletes a contractor record based on id

## Device-Registry

- Purpose: Keep a collection of all the devices deployed in your organization; it also keeps a record of where the device is.

## Endpoints:

- GET /devices: Returns a list of all devices in the database
- GET /devices/{id}: Retrieves a single device based on the id
- POST /devices: Creates a new device in the database
- PUT/PATCH /devices/{id}: Updates the record of an existing device in the database
- DELETE /devices/{id}: Removes a device's record in the database based on id

# Inventory

- Purpose: A collection of a stores inventory. It has information on how much of an item is on the shelf and in storage as well as the SKU.



## Endpoints:

- GET /stocks: Gets a list of items are considered "stock"
- GET /stocks/{id}: Returns a single item based on its id
- POST /stocks: Creates a new "stock" item
- PUT/PATCH /stocks/{id}: Updates an existing item based on id
- DELETE /stocks/{id}: Removes an item from the "stock" list

# Movies

- Purpose: Keep a collection of movies and their information such as the genre and the actors that played in it.

## Endpoints:

- GET /movies: Retrieves a list of all movies in the database
- GET /movies/{id}: Returns a specific movie based on the id
- POST /movies: Used to add a new movie to the database
- PUT/PATCH /movies/{id}: Updates a movie in the database based on the id
- DELETE /movies/{id}: Deletes a movie from the database based on id

# Orders

- Purpose: Securely have a collection of orders and their information. This includes the customers name & address and the packages current and last location.

## Endpoints:

- GET /orders: Returns all orders in the database
- GET /orders/{id}: Returns a specific order based on the id
- POST /orders: Creates a new order in the database
- PUT /orders/{id}: Updates an existing order based on the id
- DELETE /orders/{id}: Removes an order from the database based on the id

## Org. Membership

- Purpose: Keep a registry of members in your organization and if they are up to date on their dues.

## Endpoints:

- GET /members: Lists all of the members that are in the database
- GET /members/{id}: Retrieves a specific member based on the id
- POST /members: Adds a new member in the database
- PUT/PATCH /members/{id}: Updates an existing member based on their id
- DELETE /members/{id}: Removes a member from your org.'s database based on id

# Reservations

- Purpose: A secure collection of reservations of locations and how many members will be attending each reservation.



## Endpoints:

- GET /reservations: Returns all reservations that are stored in the database
- GET /reservations/{id}: Returns a specific reservation based on the id
- POST /reservations: Used to create a new reservation
- PUT /reservations/{id}: Updates an existing reservation
- DELETE /reservations/{id}: Deletes a reservation from the database based on id

# Scores

- Purpose: Keep a record of a students assignment and class scores in both a numerical and alphabetical format.

## Endpoints:

- GET /scores: Retrieves a list of all scores stored in the database
- GET /scores/{id}: Returns a single score based on the id
- POST /scores: Creates a new student and their corresponding scores in the database
- PUT/PATCH /scores/{id}: Updates the scores of a student already entered into the database
- DELETE /scores/{id}: Removes the score of a student in the database based on id

# Tasks

- Purpose: A collection of the tasks that are both complete and incomplete and what the time requirement is for each one. It is extremely helpful for managing large workloads.

## Endpoints:

- GET /tasks: Returns all tasks that have been entered into the database
- GET /tasks/{id}: Returns a specific single task based on the id
- POST /tasks: Adds a new task to the database
- PUT/PATCH /tasks/{id}: Updates the record for a current task; this will also be used to mark if a task is completed
- DELETE /tasks/{id}: Removes a task from the database based on the id