

API Documentation

All APIs are hosted through an express server that is run alongside the frontend React application. Below is listed all the API endpoints and the effects of calling them:

Note: all endpoints follow the URL of the website i.e. "http://localhost:3000/<endpoint>"

Parts APIs

Create Part:

Endpoint: "/Parts/"

Request Type: POST

Request Body: {name}

Creates a part with the name specified in the request body.

Get Parts:

Endpoint: "/Parts/" optionally "?name=<partName>"

Request Type: GET

Request Body: None

Gets all parts or optionally any part containing the specified partName variable.

Update Part:

Endpoint: "/Parts/{partID}"

Request Type: PATCH

Request Body: {name}

Updates the part name specified in the body when given the part id.

Delete Part:

Endpoint: "/Parts/{partID}"

Request Type: DELETE

Request Body: None

Deletes the part specified by the id in the endpoint.

ContainedBy APIs

Create ContainedBy:

Endpoint: "/Parts/{part id}/Containers/{container id}"

Request Type: POST

Request Body: {identifier, quantity}

Creates a relationship between a container and a part and assigns the values given in the request body.

Get ContainedBy:

Endpoint: "/Parts/Containers/" optionally

"?name=<partName>,containerName=<containerName>"

Request Type: GET

Request Body: None

Gets all containerBy relationships from the database. If the name or containerName are specified, will get parts or containers that match.

Update ContainedBy:

Endpoint: "/Parts/{part id}/Containers/{container id}"

Request Type: PATCH

Request Body: {quantity}

Updates the containedBy relationship with the quantity specified in the request body.

Delete ContainedBy:

Endpoint: "/Parts/{part id}/Containers/{container id}"

Request Type: DELETE

Request Body: None

Deletes the relationship between the specified part and container.

Containers APIs

Create Container:

Endpoint: "/Containers/"

Request Type: POST

Request Body: {name, size, location, description}

Creates a container with the name, size, location and description specified in the request body.

Get Containers:

Endpoint: "/Containers/" optionally "?name=<containerName>"

Request Type: GET

Request Body: None

Gets all the containers in the database with a name optionally included to get all containers containing that name.

Get Container:

Endpoint: "/Containers/{container id}"

Request Type: GET

Request Body: None

Gets the container from the database with the specified container ID.

Update Container:

Endpoint: "/Containers/{container id}"

Request Type: PATCH

Request Body: {name, size, location, description}

Updates the container with the id specified in the endpoint and updates the values given in the request body.

Delete Container:

Endpoint: "/Containers/{container id}"

Request Type: DELETE

Request Body: None

Deletes the container with the id specified in the endpoint.

Categories APIs

Create Category:

Endpoint: "/Categories/"

Request Type: POST

Request Body: {name}

Creates a category with the name specified in the request body.

Get Categories:

Endpoint: "/Categories/"

Request Type: GET

Request Body: None

Gets all categories in the database.

Get Category:

Endpoint: "/Categories/{category id}"

Request Type: GET

Request Body: None

Gets the category from the database with the specified category ID.

Update Category:

Endpoint: "/Categories/{category id}"

Request Type: PATCH

Request Body: {name}

Updates the category with the id specified in the endpoint and updates the name given in the request body.

Delete Category:

Endpoint: "/Categories/{category id}"

Request Type: DELETE

Request Body: None

Deletes the category with the id specified in the endpoint.

CategorizedBy APIs

Create CategorizedBy:

Endpoint: "/Categorized/Categories/{category id}/Parts/{part id}"

Request Type: POST

Request Body: None

Creates a relationship between the given category and part. This is displayed on the Parts page of the React application.

Delete CategorizedBy:

Endpoint: "/Categorized/Categories/{category id}/Parts/{part id}"

Request Type: DELETE

Request Body: None

Deletes the relationship between the given category and part.

Users APIs

Create User:

Endpoint: "/Users/"

Request Type: POST

Request Body: {username, password}

Creates a user with the specified username and password when specified

Get Users:

Endpoint: "/Users/"

Request Type: GET

Request Body: None

Gets all users in the database

Get User:

Endpoint: "/Users/{user id}"

Request Type: GET

Request Body: None

Gets user in the database with the specified ID.

Update User:

Endpoint: "/Users/{user id}"

Request Type: PATCH

Request Body: {column, value}

Updates the user specified by the user id. Column corresponds to either username or password and value is the desired update value.

Delete User:

Endpoint: "/Users/{user id}"

Request Type: DELETE

Request Body: None

Deletes user specified by the user id.

Export API

Export Data:

Endpoint: "/Export/"

Request Type: GET

Request Body: None

Exports the data in the database to a CSV file and downloads it on the frontend application.