

Food Trucks Nearby

Sheldon Mattson

Service Layer

Overview

My application is being built with React on the front-end while Express and NodeJS make up the back end, with MongoDB as the database. A single REST server API will bridge the calls from the front-end and the database to populate pages, regulate data, and verify data.

Users

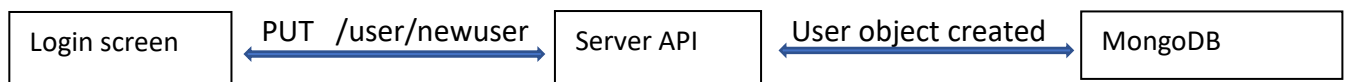
A login/sign out button is featured on all pages of the application.

Create Account

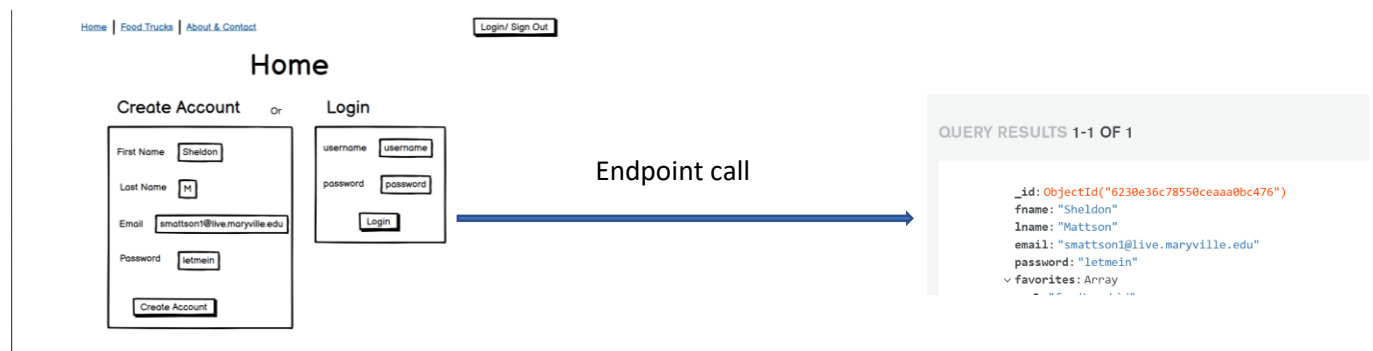
Method: PUT

When a user creates an account the system will check for any restrictions, such as password length, and create a user object in the database using the provided information. If there is an error with information provided or the system an error will occur.

Example:



Successful:



User will not see the data created, rather they will be logged into account with appropriate home screen.

Error:

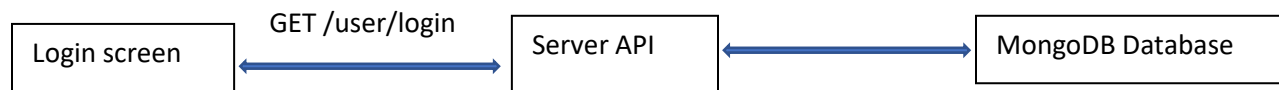
```
{
  "code": 400,
  "error message": "Password is not long enough. Please try a different one."
}
```

Login to Account

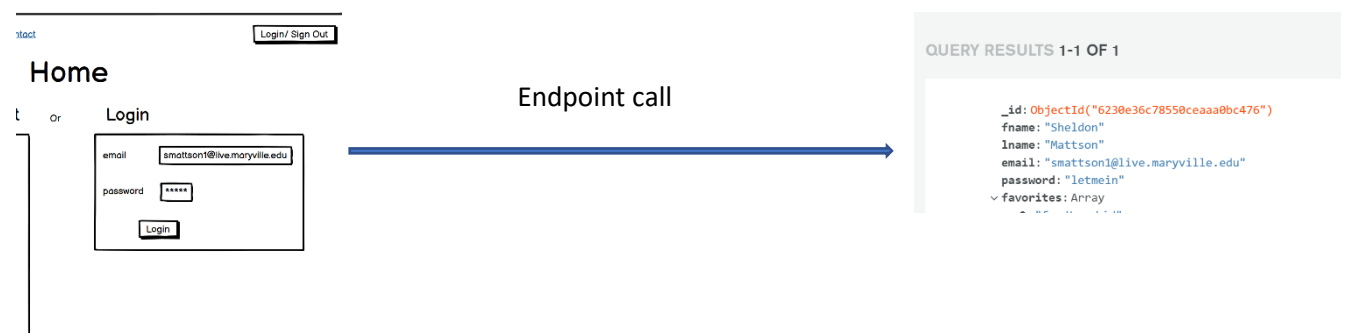
Method: GET

This endpoint will be called when a user clicks the login button and enters information to login to an account. The login credentials will be checked and verified with current users. If the user is found, they will be logged into account. If not, an error will occur.

Example:



Successful:



Error:

```
{
  "code": 401,
  "error message": "Invalid login information. Please enter a valid email or password."
}
```

Food Trucks

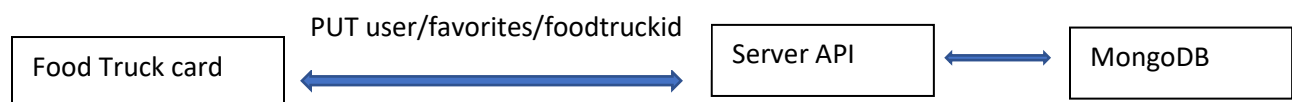
Favorites

The favorites feature allows users to save their favorite food trucks to a list. Favorited food trucks will populate on the main page of a logged in user and indicate as such on the food trucks page.

Method: PUT

Pressing the favorite button/icon will add a food truck to the users' favorites if not already favorited.

Example:



Success:

```
{
  "id": string,
  ....,
  "favorites": array [
    {foodtruckid: string}
  ]
}
```

Error:

```
{
  "code": 400,
  "error message": "An error has occurred. Please try again."
}
```

Method: DELETE

Pressing the favorite button/icon of a food truck indicated as a favorite will remove it from the list.

Example:



Success:

```
{
  "id": string,
  ...,
  "favorites": array [
  ]
}
```

Error:

```
{
  "code": 400,
  "error message": "An error has occurred. Please try again."
}
```

Search

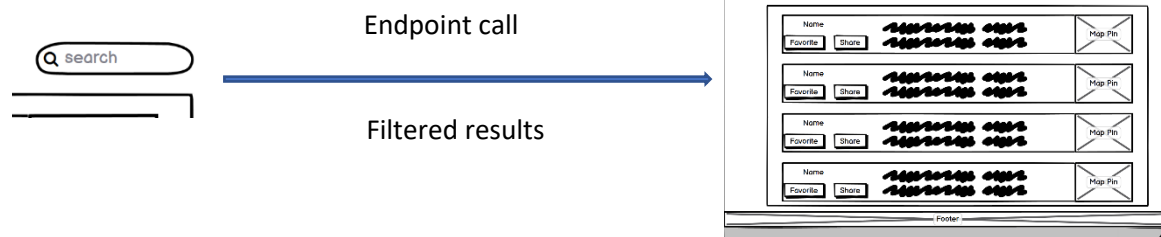
Method: GET

On the main food trucks page, a user may need to filter what is provided. For instance, Mexican food trucks or looking for an exact name. The service will query for data objects that contain matching tags.

Example:



Successful:



Error:

```
{
  "code": 404,
```

```
    "error message": "No matches found."  
}
```

Additional

The system automatically calls the GET method to populate pages and favorites with informational cards of the food trucks on load. Successful methods yield food truck cards while any errors in populating fields will yield an "Error 404: Page Not Found" error.