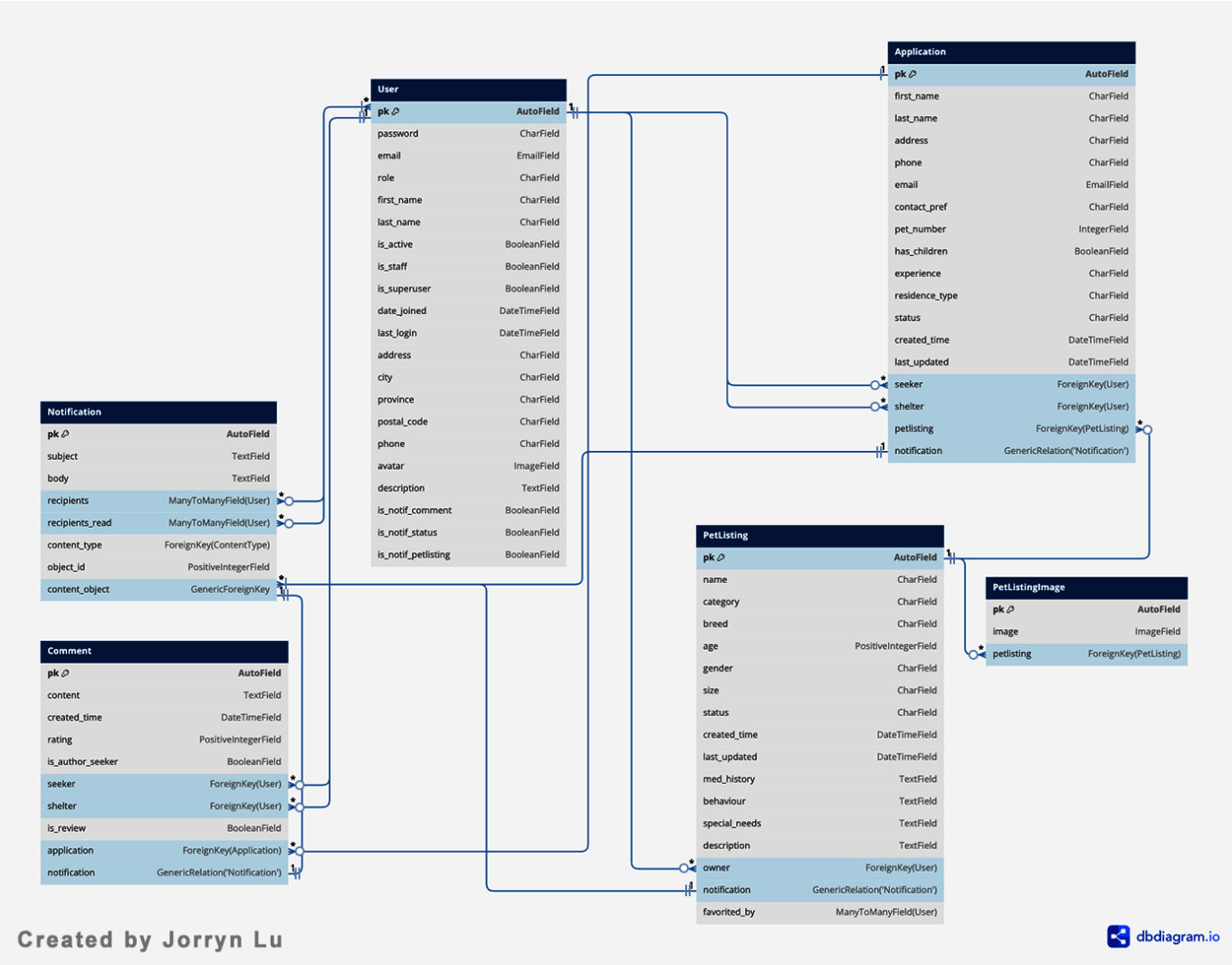


# CSC309 Group\_0346 PetPal Overview

The following link is our repository on github, which contains the same code as our P2 submission but with a complete commit history:

<https://github.com/lujorryn/PetPal-G0346>

## Model Entity Relationship Diagram



## Simple JWT Token-based Authentication

We use simple JWT's TokenObtainPairView to implement our login function.

---

### LOGIN

**Endpoint:** `/api/token/`

**Description:** Generates a refresh and an access token if given the correct email and password combination.

**Methods:** `POST`

**Required Payload:** `email`, `password`

**Permissions:** Any user

#### ERRORS:

- **400 Bad Request:** Missing `email` and/or `password`
- **401 Unauthorized:** Incorrect username and password combination

**SUCCESS:** Return a refresh token and an access token. Example:

```
{  "refresh": "eyJhbGciOiJIUzI1NiIs...",
  "access": "eyJhbGciOiJIUzI1NiIs..." }
```

---

## The accounts app

The accounts app defines a custom `PetPalUser` model derived from Django's `AbstractUser` that includes all extra fields specified in the ERD diagram. The app includes `signup` (creating a new `PetPalUser`) and `delete account` functions.

---

### SIGNUP

**Endpoint:** `/api/accounts/new-account`

**Description:** Creates a shelter account (new `PetPalUser`) if `account_type` is `shelter`, otherwise create a seeker account.

**Methods:** `POST`

**Required Payload:** `email`, `password`, `confirm_password`, `account_type`

**Optional Payload:** `first_name`, `last_name`

**Permissions:** Any user

#### ERRORS:

- **400 Bad Request:**
  - Invalid email address
  - Missing any field in required payload
  - `password` and `confirm_password` don't match
  - User with this email already exists

**SUCCESS:** Return a message that indicates user <user\_id> SEEKER/SHELTER successfully created.

Example:

```
{  "msg": "User 7: SEEKER successfully created"  }
```

---

## DELETE ACCOUNT

**Endpoint:** /api/accounts/<int:account\_id>

**Description:** Delete an account if account is owned by the user and all its owned items (e.g. seeker: applications shelter: petlistings etc)

**Methods:** DELETE

**Permissions:** Account owner

**ERRORS:**

- 401 Unauthorized
  - Authentication credentials were not provided. (user not logged in)
- 403 Forbidden:
  - account\_id does not match with request.user.id

**SUCCESS:** Return a message that indicates user <user\_id> deleted. Example:

```
{  "msg": "User 1 deleted"  }
```

---

## The seekers app

The seekers app corresponds to users looking for a pet. The app provides the functions to view and update a user's own profile, view their favorite pets, and add/delete pets from their favorites.

---

## ALL SEEKER USERS

**Endpoint:** /api/seekers

**Description:** Forbids anyone from looking at the list of all seeker users.

**Methods:** GET

**Permissions:** None

**ERRORS:**

- 403 Forbidden
  - Every user, authenticated or not, should receive this error.

---

## USER DETAIL VIEW

**Endpoint:** /api/seekers/<int:account\_id>

**Description:** Get the details of a seeker and update the user's data if needed.

**Methods:** GET PUT

**Optional Payload (PUT):** email, address, city, province, postal\_code, phone, avatar,

`is_notif_comment, is_notif_status, is_notif_petlisting`

**Permissions:** Account owner has permissions for GET and PUT requests, only GET for shelters with an active application

**ERRORS:**

- **401 Unauthorized**
  - Authentication credentials were not provided (user not logged in).
- **403 Forbidden:**
  - `account_id` does not match with `request.user.id`
- **400 Bad Request:**
  - User tried to update a field incorrectly.
- **404 Forbidden:**
  - Account with `account_id` does not exist.

**SUCCESS:** Return the data of `<account_id>`. Example:

```
{
  "msg": "Seeker details",
  "user_data": {
    "email": "seeker2@example.com",
    "address": "",
    "city": "",
    "province": "",
    "postal_code": "",
    "phone": "",
    "avatar": null,
    "is_notif_comment": true,
    "is_notif_status": true,
    "is_notif_petlisting": true
  },
  "fav_pets": []
}
```

---

## USER PET FAVORITES

**Endpoint:** `/api/seekers/<int:account_id>/favorites`

**Description:** Get the list of favorite pets by the user with id `account_id`

**Methods:** `GET`

**Permissions:** Account owner only

**ERRORS:**

- **401 Unauthorized**
  - Authentication credentials were not provided (user not logged in).
- **403 Forbidden:**
  - `account_id` does not match with `request.user.id`
- **404 Not Found:**
  - account with `account_id` not found

**SUCCESS:** Return a list with `<account_id>` favorites. Example:

```
{
  "page": {
    "current": 1,
    "has_next": false,
    "has_previous": false
  },
  "msg": "seeker1@example.com Favorites",
  "data": [
    {
      "id": 1,
      "name": "Buddy",
      "category": "D",
      "breed": "Golden Retriever",
      "age": 3,
      "gender": "M",
      "size": "L",
      "status": "",
      "med_history": "None",
      "behaviour": "Friendly",
      "special_needs": "",
      "description": "A very friendly dog."
    }
  ]
}
```

---

## ADD/DELETE FAVORITES

**Endpoint:** `/api/seekers/<int:account_id>/favorites/<int:pet_id>`

**Description:** Add or delete a pet with `pet_id` on `account_id`'s favorites

**Methods:** `POST`, `DELETE`

**Permissions:** Account owner only

### ERRORS:

- **401 Unauthorized**
  - Authentication credentials were not provided (user not logged in).
- **403 Forbidden:**
  - `account_id` does not match with `request.user.id`.
- **404 Not Found:**
  - account with `account_id` or pet with `pet_id` not found
- **409 Conflict**
  - `pet_id` is already in favorites (add), or is not in favorites (delete)

**SUCCESS:** Return a message of add/delete successful and `pet_id` Examples:

```
{ "msg": "Added 4 to favorites" }
```

```
{ "msg": "Removed 4 from favorites" }
```

---

## The shelters app

The shelters app corresponds to shelters that have pets up for adoption. The app provides the functions to view and update a shelter's own profile.

---

### ALL SHELTERS

**Endpoint:** `/api/shelters`

**Description:** Provides a list view of all shelters.

**Methods:** `GET`

**Permissions:** Any logged-in users

### ERRORS:

- **401 Unauthorized**
  - Authentication credentials were not provided (user not logged in).

**SUCCESS:** Return a list of the shelters. Example:

```
{
  "page": {
    "current": 1,
    "has_next": true,
    "has_previous": false
  },
  "data": [
    {
      "id": 4,
      "email": "shelter1@example.com",
      "address": "",
      "city": "Vancouver",
      "province": "",
      "postal_code": "",
      "phone": "",
      "avatar": null,
      "description": ""
    },
    {
      "id": 5,
      "email": "shelter2@example.com",
      "address": "",
      "city": "",
      "province": "",
      "postal_code": "",
      "phone": "",
      "avatar": null,
      "description": ""
    }
  ]
}
```

```
]
}
```

---

## SHELTER DETAIL VIEW

**Endpoint:** `/api/shelters/<int:account_id>`

**Description:** Provides a view of the shelter with `account_id` to see details or update (if permitted)

**Methods:** `GET`, `PUT`

**Permissions:** Any logged-in user has access to `GET`, only account holders can make a `PUT` request.

**Optional Payload (PUT):** `email`, `address`, `city`, `province`, `postal_code`, `phone`, `avatar`, `description`, `is_notif_comment`, `is_notif_status`, `is_notif_petlisting`

### ERRORS:

- **401 Unauthorized**
  - Authentication credentials were not provided (user not logged in).
- **403 Forbidden:**
  - `account_id` does not match with `request.user.id`.
- **404 Forbidden:**
  - Account with `account_id` does not exist.
- **400 Bad Request:**
  - User tried to update a field incorrectly.

**SUCCESS:** Return a the details of a shelter. Example:

```
{
  "msg": "Shelter Detail",
  "data": {
    "email": "shelter1@example.com",
    "address": "",
    "city": "Vancouver",
    "province": "",
    "postal_code": "",
    "phone": "",
    "avatar": null,
    "description": "",
    "is_notif_comment": true,
    "is_notif_status": true,
    "is_notif_petlisting": true
  }
}
```

---

## The petlistings app

The petlisting app defines a `PetListing` model with its fields specified in the ERD diagram. The app implements the following 5 endpoints that support creating, updating, viewing, and deleting Petlistings:

---

### GET Petlisting

**Endpoint:** `/api/petlistings`

**Description:** Get a petlisting

**Methods:** `GET`

**Optional Payload:** `category, age, status, gender, size, shelter_email, name`

**Permissions:**

- Any logged in user

#### ERRORS:

- **400 Bad Request:**
  - Any optional field doesn't match the model's field options
- **401 Unauthorized**
  - Authentication credentials were not provided. (user not logged in)

**SUCCESS:** Return a Json of pets that fit the filter requirements

---

### CREATE Petlisting

**Endpoint:** `/api/petlistings`

**Description:** Create a petlisting

**Methods:** `POST`

**Required Payload:** `name, category, gender, size`

**Optional Payload:** `age, breed, description, med_history, behaviour, special_needs`

**Permissions:**

- Any logged in user and is a shelter

#### ERRORS:

- **400 Bad Request:**
  - Missing any field in required payload
- **401 Unauthorized**
  - Authentication credentials were not provided. (user not logged in)
  - User is not a shelter

**SUCCESS:** Return a message that indicates petlisting created and returns listing details.

---

### GET Petlisting detail

**Endpoint:** `/api/petlistings/<int:pet_id>`

**Description:** Get a petlisting details

**Methods:** `GET`

**Permissions:**

- Any logged in user

#### ERRORS:

- **400 Bad Request:**



- Listing doesn't exist
- **401 Unauthorized**
  - Authentication credentials were not provided. (user not logged in)

**SUCCESS:** Return a message that indicates success and returns listing details.

---

### PUT Petlisting detail

**Endpoint:** `/api/petlistings/<int:pet_id>`

**Description:** Update a petlisting details

**Methods:** **PUT**

**Required Payload:** `name, category, gender, size`

**Optional Payload:** `age, breed, description, med_history, behaviour, special_needs`

**Permissions:**

- Any logged in user and is the shelter that created the listing

#### ERRORS:

- **400 Bad Request:**
  - Missing any field in required payload
  - Listing doesn't exist
- **401 Unauthorized**
  - Authentication credentials were not provided. (user not logged in)
  - User is not a shelter

**SUCCESS:** Return a message that indicates petlisting updated and returns listing details.

---

### DELETE Petlisting

**Endpoint:** `/api/petlistings/<int:pet_id>`

**Description:** Delete a petlisting

**Methods:** **DELETE**

**Permissions:**

- Any logged in user and is the shelter which created the listing

#### ERRORS:

- **400 Bad Request:**
  - Listing doesn't exist
- **401 Unauthorized**
  - Authentication credentials were not provided. (user not logged in)
  - User is not the shelter that created the listing

**SUCCESS:** Return a message that indicates petlisting is deleted

---

## The applications app

Allows the user to view, create, and modify applications for a petlisting.

Shelters can view all of the applications for a petlisting.

Seekers can view their applications only.

---

## ENDPOINTS

### Endpoint:

`/api/application`

### Description:

Create an application for a petlisting or view a list of applications

### Methods:

`GET, POST`

POST request:

### Required Payload:

- `first_name, last_name, address, phone, email, contact_pref, pet_number, has_children, experience, residence_type, status, petlisting_id`

### Permissions:

- Shelter:
  - must be authenticated
- Seeker:
  - must be authenticated

### ERRORS:

- **400 Bad Request:**
  - Invalid data in the payload.
  - User invalid (when trying to retrieve the current user from the database).
  - PetListing invalid (when trying to retrieve the petlisting from the database).
  - PetListing unavailable (when the status of the petlisting is not 'AV' - available).
- **401 Unauthorized:**
  - Authentication credentials were not provided (user not logged in).

### SUCCESS:

Return a redirect url and success status code:

```
{
  "redirect_url": "/api/applications/<application_id>", "status": 201
}
```

GET request:

### Permissions:

- Shelter:
  - must be authenticated
- Seeker:
  - must be authenticated

### ERRORS:

- **400 Bad Request:**
  - Invalid status or sort option provided in the query parameters.
  - Invalid user role (when the user role is not 'SHELTER' or 'SEEKER').
  - Sorting option not supported.
- **401 Unauthorized:**
  - Authentication credentials were not provided (user not logged in).

### SUCCESS:

- Shelter:
  - Return a paginated list of the applications for all of the shelter's pet listings with a status 200
- Seeker:
  - Return a paginated list of the user's applications with a status 200

```
{
  "count": 35,
  "next": "http://localhost:8000/api/applications?page=2",
  "previous": null,
  "results": {
    "data": [
      {
        "id": 37,
        "first_name": "John",
        "last_name": "Doe",
        "address": "123 Main St",
        "phone": "555-1234-9999999999999999",
        "email": "shelter1@example.com",
        "contact_pref": "P",
        "pet_number": 2,
        "has_children": false,
        "experience": "EX",
        "residence_type": "C",
        "status": "P",
        "created_time": "2023-11-12T01:09:20.055224Z",

```

```
        "last_updated": "2023-11-12T01:09:20.055265Z",
        "seeker": 4,
        "shelter": 4,
        "petlisting": 1
      },
      ...
    ]
  }
}
```

---

**Endpoint:**

/api/applications/pet/<int:pet\_id>

**Description:**

List all applications for a pet listing (pet\_id)

**Methods:**

GET

GET request:

**Permissions:**

- Shelter:
  - must be authenticated
- Seeker:
  - not allowed

**ERRORS:**

- **400 Bad Request:**
  - Invalid pet\_id was provided.
- **401 Unauthorized:**
  - Shelter does not own the pet listing.

**SUCCESS:**

- Return a paginated list of the applications for a petlisting with a status 200.

```
{
  "count": 35,
  "next": "http://localhost:8000/api/applications/pet/1?page=2",
  "previous": null,
  "results": {
    "data": [
```

```
{
  {
    "id": 37,
    "first_name": "John",
    "last_name": "Doe",
    "address": "123 Main St",
    "phone": "555-1234-9999999999999999999",
    "email": "shelter1@example.com",
    "contact_pref": "P",
    "pet_number": 2,
    "has_children": false,
    "experience": "EX",
    "residence_type": "C",
    "status": "P",
    "created_time": "2023-11-12T01:09:20.055224Z",
    "last_updated": "2023-11-12T01:09:20.055265Z",
    "seeker": 4,
    "shelter": 4,
    "petlisting": 1
  },
  ...
]
}
```

---

**Endpoint:**

/api/applications/<int:app\_id>

**Description:**

View or Edit the specified application

**Methods:**

GET, PUT

GET request:

**Permissions:**

- Shelter:
  - must be authenticated and own the application
- Seeker:
  - must be authenticated and own the application

**ERRORS:**

- **400 Bad Request:**
  - Invalid app\_id was provided.
- **401 Unauthorized:**

- Shelter does not own the pet listing for the application
- Seeker is not the author of the application

**SUCCESS:**

- Return the details of the application with a status 200.

```
{
  "data": {
    "id": 12,
    "first_name": "John",
    "last_name": "Doe",
    "address": "123 Main St",
    "phone": "555-1234",
    "email": "shelter1@example.com",
    "contact_pref": "P",
    "pet_number": 2,
    "has_children": false,
    "experience": "EX",
    "residence_type": "C",
    "status": "D",
    "created_time": "2023-11-11T15:12:47.599277Z",
    "last_updated": "2023-11-11T15:13:17.139519Z",
    "seeker": 4,
    "shelter": 4,
    "petlisting": 1
  }
}
```

PUT request:

**Required Payload:**

- **status**

**Permissions:**

- Shelter:
  - must be authenticated and own the application
- Seeker:
  - must be authenticated and own the application

**ERRORS:**

- **400 Bad Request:**
  - Invalid app\_id was provided.
  - Seeker wanted to update the application with a status other than 'w'
  - Shelter wanted to update the application with a status other than 'A' or 'D'

- **401 Unauthorized:**
  - Shelter does not own the pet listing for the application
  - Seeker is not the author of the application

**SUCCESS:**

- Return a redirect url and success status code:

```
{
  "redirect_url": "/api/applications/<application_id>", "status": 201
}
```

---

## The comments app

The comments app defines a Comment model with its fields specified in the ERD diagram. The app implements the following 5 endpoints that support creating, listing and viewing comments:

---

**CREATE COMMENT**

**Endpoint:** `/api/comments`

**Description:** Create a comment regarding an application or a shelter

**Methods:** `POST`

**Required Payload:** `is_review`, `content`, `recipient_email`

**Optional Payload:** `rating`, `application` (required if `is_review=True`)

**Permissions:**

- Application comments: application seeker and application's petlisting's owner
- Shelter reviews: Any logged in user

**ERRORS:**

- **400 Bad Request:**
  - SEEKER-SEEKER comments and SHELTER-SHELTER comments not supported
  - Missing any field in required payload
  - Application comment with no valid application\_id
- **401 Unauthorized**
  - Authentication credentials were not provided. (user not logged in)
  - Application comments: User does not match permissions specified above

**SUCCESS:** Return a message that indicates Review/Comment created and returns comment details.

Example:

```
{
  "msg": "Review Created",
  "data": {
    "To": "seeker1@example.com",
    "From": "shelter1@example.com",
  }
}
```

```
    "Message": "Awesome!",  
    "rating": 5  
  }  
}
```

---

**Endpoint:** `/api/comments/applications`

**Description:** List all comments of user's all applications with pagination by application supported

**Methods:** GET

**Fields:** `id, content, created_time, rating=null, is_author_seeker, seeker(email display), shelter(email display), is_review=False, application(id display)`

**Permissions:** Logged in user can view their own application comments

**ERRORS:**

- **401 Unauthorized**
  - Authentication credentials were not provided. (user not logged in)

**SUCCESS:** Return a page indicator (current page, has\_next, has\_previous) and all the data fields.

---

**Endpoint:** `/api/comments/applications/<int:app_id>`

**Description:** List all comments of a user's specific application with pagination supported

**Methods:** GET

**Fields:** `id, content, created_time, rating=null, is_author_seeker, seeker(email display), shelter(email display), is_review=False, application(id display)`

**Permissions:** Not the application seeker or the application's petlisting owner

**ERRORS:**

- **400 Bad Request:**
  - Application with `app_id` does not exist
- **401 Unauthorized**
  - Authentication credentials were not provided. (user not logged in)
  - Not the application seeker or the application's petlisting owner

**SUCCESS:** Return a page indicator (current page, has\_next, has\_previous) and all the data fields.

---

**Endpoint:** `/api/comments/applications/<int:msg_id>`

**Description:** View a comment's contents

**Methods:** GET

**Fields:** `id, content, created_time, rating, is_author_seeker, seeker(email display), shelter(email display), is_review, application(id display)`

**Permissions:** Author/Recipient (seeker or shelter of Comment object) of comment

**ERRORS:**

- **400 Bad Request:**
  - Comment with `msg_id` does not exist
- **401 Unauthorized**
  - Authentication credentials were not provided. (user not logged in)



- Not author/recipient

**SUCCESS:** Return a page indicator (current page, has\_next, has\_previous) and all the data fields.

```
{
  "data": {
    "id": 1,
    "content": "I would love to see Buddy in person!",
    "created_time": "2023-11-10T19:33:02.452885Z",
    "rating": null,
    "is_author_seeker": true,
    "seeker": "seeker1@example.com",
    "shelter": "shelter1@example.com",
    "is_review": false,
    "application": 1
  }
}
```

---

**Endpoint:** `/api/comments/shelter/<int:shelter_id>`

**Description:** List all comments of a shelter with pagination supported

**Methods:** `GET`

**Fields:** `id, content, created_time, rating=null, is_author_seeker, seeker(email display), shelter(email display), is_review=False, application(id display)`

**Permissions:** Any logged in user

**ERRORS:**

- **400 Bad Request:**
  - Shelter with `shelter_id` does not exist
- **401 Unauthorized**
  - Authentication credentials were not provided. (user not logged in)

**SUCCESS:** Return a page indicator (current page, has\_next, has\_previous) and all the data fields.

---

## The notification app

The notifications app defines a Notification model with its fields specified in the ERD diagram. The app implements the following 4 endpoints that support reading, updating, and deleting notifications:

---

### GET Notification

**Endpoint:** `/api/notifications`

**Description:** Get all notification

**Methods:** `GET`

**Permissions:**

- User has the notifications

**ERRORS:**

- **400 Bad Request:**
- **401 Unauthorized**
  - Authentication credentials were not provided. (user not logged in)

**SUCCESS:** Return a message that indicates success and returns the list of user's notifications

---

### GET Notification Details

**Endpoint:** `/api/notifications/<int:note_id>`

**Description:** Get a specific notification

**Methods:** **GET**

**Permissions:**

- User has the notification

### ERRORS:

- **400 Bad Request:**
  - Notification doesn't exist
- **401 Unauthorized**
  - Authentication credentials were not provided. (user not logged in)

**SUCCESS:** Return a message that indicates success and returns the user's notification

---

### PUT Notification

**Endpoint:** `/api/notifications/<int:note_id>`

**Description:** Update a specific notification to read

**Methods:** **PUT**

**Permissions:**

- User has the notification

### ERRORS:

- **400 Bad Request:**
  - Notification doesn't exist
- **401 Unauthorized**
  - Authentication credentials were not provided. (user not logged in)

**SUCCESS:** Return a message that indicates success

---

### DELETE Notification

**Endpoint:** `/api/notifications/<int:note_id>`

**Description:** Delete a specific notification

**Methods:** **DELETE**

**Permissions:**

- User has the notification

**ERRORS:**

- **400 Bad Request:**
  - Notification doesn't exist
- **401 Unauthorized**
  - Authentication credentials were not provided. (user not logged in)

**SUCCESS:** Return a message that indicates success of deletion

---