# CSC309 Phase 2 Documentation (Group 1160)

In order to run the Django server:
1. Pull the repository to your local.
2. Locate the `'petpal'` directory in the terminal.
3. Run '`chmod +x startup.sh'` and '`chmod +x run.sh`' to make the scripts executable.
4. Run '`./startup.sh`' in the terminal.
5. Run '`./run.sh`' in the terminal

After running the server, you can run `'make'` in the terminal to delete your virtual environment and empty the database.

Now that you have the server running, below are the endpoints for each of the apps:

## Accounts:

**Models:**

**CustomUser Model**
**Fields:**

- **username (string): The unique username for the user.**
- **email (email): The unique email address for the user.**
- **password (string): The password for the user.**
- **seeker (boolean, default=False): Indicates whether the user is a pet seeker.**
- **shelter (boolean, default=False): Indicates whether the user is a shelter.**
- **groups (many-to-many relationship with auth.Group): The groups this user belongs to. A user will get all permissions granted to each of their groups.**
- **user_permissions (many-to-many relationship with auth.Permission): Specific permissions for this user.**

**Shelter Model**
**Fields:**

- **user (one-to-one relationship with CustomUser, related name='shelter_user'): Reference to the corresponding user for the shelter.**
- **shelter_name (string): The name of the shelter.**
- **location (string): The location of the shelter.**
- **mission_statement (text): A text field describing the mission statement of the shelter.**

**PetSeeker Model**
**Fields:**

- **user (one-to-one relationship with CustomUser, related name='seeker_user'): Reference to the corresponding user for the pet seeker.**
- **seeker_name (string): The name of the pet seeker.**
- **location (string): The location of the pet seeker.**
- **preferences (string): A field to store the preferences of the pet seeker.**
- **profile_picture (image, optional): An optional image field to store the profile picture of the pet seeker.**

**Token-based Authentication:**

- **Endpoint: /accounts/login/**
- **Method: POST**
- **Description: Obtains a JSON Web Token (JWT) for authentication.**
- **Payload:**
  **{ "username": "your_username", "password": "your_password" }**
- **Response:**
  **{ "access": "your_access_token", "refresh": "your_refresh_token" }**
- **Authentication: Public**

- **Endpoint: /accounts/login/refresh/**
- **Method: POST**
- **Description: Refreshes a JSON Web Token (JWT).**
- **Payload:**
  **{ "refresh": "your_refresh_token" }**
- **Response:**
  **{ "access": "your_new_access_token" }**
- **Authentication: Public**

**User Registration and Profile Management:**
**Shelter:**

- **Endpoint: /accounts/shelters/**
- **Method: POST**
- **Description: Registers a new shelter account (Only username, email, password and name required).**
- **Payload:**
  **{ "username": "shelter_username", "email": "shelter_email@example.com", "password": "shelter_password", "shelter_name": "Shelter Name", "location": "Shelter Location", "mission_statement": "Shelter Mission Statement" }**
- **Authentication: Public**

- **Endpoint: /accounts/shelters/<str:username>/**
- **Method: PUT**
- **Description: Updates the information of an existing shelter account.**
- **Payload:**
  **{ "shelter_name": "Updated Shelter Name", "location": "Updated Shelter Location", "mission_statement": "Updated Shelter Mission Statement" }**
- **Authentication: Respective shelter only (JWT required)**

- **Endpoint: /accounts/shelters/profile/<str:username>/**
- **Method: GET**
- **Description: Retrieves the profile information of a shelter.**
- **Response:**

```
{
        "id": 1,
        "user": {
           "id": 1,
           "username": "yoyo",
           "email": "yoyo@gmail.com",
           "seeker": false,
           "shelter": true
        },
        "shelter_name": "yoyo",
        "location": "yo",
        "mission_statement": "yo"
    }
```

- **Authentication: Any authenticated user (seeker or shelter) (JWT required)**

**Pet Seeker:**

- **Endpoint: /accounts/petseekers/**
- **Method: POST**
- **Description: Registers a new pet seeker account.**
- **Payload:**
  **{ "username": "seeker_username", "email": "seeker_email@example.com", "password": "seeker_password", "seeker_name": "Pet Seeker Name", "preferences": "Pet Seeker Preference", "location": "Pet Seeker Location", "profile_picture": "base64_encoded_image" // Optional }**
- **Authentication: Public**

- **Endpoint: /accounts/petseekers/<str:username>/**
- **Method: PUT**

- **Description: Updates the information of an existing pet seeker account.**
- **Payload:**
  **{ "seeker_name": "Updated Pet Seeker Name", "location": "Updated Pet Seeker Location", "profile_picture": "base64_encoded_image" // Optional }**
- **Authentication: Respective pet seeker only (JWT required)**

- **Endpoint: /accounts/petseekers/profile/<str:username>/**
- **Method: GET**
- **Description: Retrieves the profile information of a pet seeker.**
- **Response:**

```
{
   "id": 1,
   "user": {
      "id": 2,
      "username": "seekeri2",
      "email": "seekeri2@gmail.com",
      "seeker": true,
      "shelter": false
   },
   "seeker_name": "seekeri2",
   "location": "seekeri2",
   "profile_picture": "/static/seekeri2_profile.jpg",
   "preferences": "my_preferences"
}
```

- **Authentication: Respective pet seeker and only shelters with whom the seeker has an active application (JWT required)**

**List Shelters:**

- **Endpoint: /accounts/all_shelters/**
- **Method: GET**
- **Description: Retrieves a list of shelters.**
- **Response:**

```
{
   "count": 1,
   "next": null,
   "previous": null,
   "results": [
      {
         "id": 1,
         "user": {
            "id": 1,
            "username": "yoyo",
                "email": "yoyo@gmail.com",
                "seeker": false,
                "shelter": true
            },
            "shelter_name": "yoyo",
            "location": "yo",
            "mission_statement": "yo"
        }, …// other shelters
    ]
}
```

- **Authentication: Public**

**User Deletion:**

- **Endpoint: /accounts/shelters/<str:username>/**
- **Method: DELETE**
- **Description: Deletes a shelter account and all associated data.**
- **Authentication: Respective shelter only (JWT required)**

- **Endpoint: /accounts/petseekers/<str:username>/**
- **Method: DELETE**
- **Description: Deletes a pet seeker account and all associated data.**
- **Authentication: Respective pet seeker only (JWT required)**

## Pet Listings:

**Models:**

**Pets Model**
**Fields:**

- **shelter (foreign key to CustomUser):** The shelter to which the pet belongs. Each pet is associated with a unique shelter through a foreign key relationship with the CustomUser model.
- **name (string):** The name of the pet.
- **age (integer):** The age of the pet in years.
- **species (string)**: The species of the pet (e.g., dog, cat).
- **breed (string)**: The breed of the pet.
- **description (text):** A detailed description of the pet.
- **image (image field):** An image field allowing the upload of a picture of the pet.
- **location (string):** The current location of the pet.
- **color (string):** The color of the pet's fur or feathers.
- **date_added (datetime, auto-generated on creation):** The date and time when the pet was added to the system.

- **size (choice field, default 'Small', choices: 'Small', 'Medium', 'Large'):** The size category of the pet
- **gender (choice field, default 'Male', choices: 'Male', 'Female'):** The gender of the pet.
- **status (choice fielddefault 'Available', choices: 'Available', 'Pending', 'Adopted'):** The adoption status of the pet.

## Pet Creation

- **Endpoint: /pets/create/**
- **Method: POST**
- **Payload:**
  **{"name": "Fluffy", "age": 3, "species": "Dog", "breed": "Golden Retriever", "description": "A friendly and playful dog looking for a loving home.", "location": "Cityville", "color": "Golden", "image": "path/to/image.jpg", "date_added": "2023-11-14T10:30:00"}**
- **Description: Creates a new pet for the logged in Shelter**
- **Response:**

  ```
  {
     "id": 2,
     "shelter": 1,
     "name": "Fluffy",
     "age": 2,
     "species": "Dog",
     "breed": "Golden Retriever",
     "description": "A friendly and playful dog looking for a loving home.",
     "image": null,
     "location": "Cityville",
     "color": "Golden",
     "date_added": "2023-11-15T19:04:34.840825Z",
     "size": "Small",
     "gender": "Male",
     "status": "Available"
  }
  ```
  **id, shelter, date_added may differ based on Logged in Shelter**

- **Authentication: Only shelter (JWT required)**

## Pet Delete

- **Endpoint: /pets/<int:pet_id>**
- **Method: DELETE**
- **Description: Delete the pet with given pet_id**
- **Authentication: Only shelter (JWT required)**

## Pet Update

- **Endpoint: /pets/<int:pet_id>**
- **Method: PUT**
- **Description: Update the pet with given pet_id**

- **Payload:**
  **{"name": "Toofe", "age": 10, status=”Adopted”}**
- **Response**

```
{
   "id": 2,
   "shelter": 1,
   "name": "Toofe",
   "age": 10,
   "species": "Dog",
   "breed": "Golden Retriever",
   "description": "A friendly and playful dog looking for a loving home.",
   "image": null,
   "location": "Cityville",
   "color": "Golden",
   "date_added": "2023-11-15T19:04:34.840825Z",
   "size": "Small",
   "gender": "Male",
   "status": "Adopted”
}
```

  **id, shelter, date_added may differ based on Logged in Shelter**

- **Authentication: Only shelter (JWT required)**

**Pet Details**

- **Endpoint: /pets/<int:pet_id>**
- **Method: GET**
- **Payload: {id=2}**
- **Description: Get the pet details with given pet_id**
- **Response:**
- **Response**

```
{
   "id": 2,
   "shelter": 1,
   "name": "Toofe",
   "age": 10,
   "species": "Dog",
   "breed": "Golden Retriever",
   "description": "A friendly and playful dog looking for a loving home.",
   "image": null,
   "location": "Cityville",
   "color": "Golden",
   "date_added": "2023-11-15T19:04:34.840825Z",
   "size": "Small",
   "gender": "Male",
   "status": "Adopted”
}
```

- **Authentication: ANY**

**Pet Listing (All)**

- **Endpoint: /pets/all>**
- **Method: GET**
- **Description: Get all the pets**
- **Authentication: ANY**

**Pet Listing (Search)**

- **Endpoint: /pets/search/>**
- **Method: GET**
- **Params: age=2, status="Adopted”, … other filtering options, sort_by = name, age, size**
- **Description: Get filtered and sorted pets**
- **Authentication: ANY**

## Comments:

**Models:**

**Comment Model**
**Fields:**

- **username (string): The unique username for the user.**
- **text (string): The comment written by the user.**
- **created_at (DateTime): The time at which the comment was created (auto populated)**
- **user (int): user ID referenced from the CustomUser model**
- **shelter (int): shelter ID referenced from the Shelter model**
- **application (int) = application ID referenced from the Application model**

**Data in the Comment Model is sorted in descending order by the ‘created_at’ field.**

**Comment Creation and Listing for Shelter Comments**

- **Endpoint: /comments/shelters/<int:shelter_id>/comments/**
- **Method: POST**
- **Description: Write a comment for a shelter**
- **Payload:**
  **{ "text": "shelter comment" }**
- **Response:**
  **{"id": 1, "text": "shelter comment", "created_at": "2023-11-15T19:30:10.396356Z", "user": 1, "shelter": 1, "application": null}**
- **Authentication: Logged in user only (JWT required)**

- **Endpoint: /comments/shelters/<int:shelter_id>/comments/**
- **Method: GET**

- **Description: View all shelter comments**
- **Response:**
  **{"count": 1, "next": null, "previous": null, "results": [ {"id": 2, "text": "application comment", "created_at": "2023-11-15T19:30:10.396356Z", "user": 1, "shelter": 1, "application": null }]}**
- **Authentication: Logged in user only (JWT required)**

**<u>Comment Creation and Listing for Application Comments</u>**

- **Endpoint: /comments/applications/<int:application_id>/comments/**
- **Method: POST**
- **Description: Write a comment for an application**
- **Payload:**
  **{ "text": "application comment"}**
- **Response:**
  **{"id": 2, "text": "application comment", "created_at": "2023-11-15T19:30:10.396356Z", "user": 1, "shelter": 1, "application": 1}**
- **Authentication: Respective shelter and seeker only (JWT required)**

- **Endpoint: /comments/applications/<int:application_id>/comments/**
- **Method: GET**
- **Description: View all application comments**
- **Response:**
- **{"count": 1, "next": null, "previous": null, "results": [ {"id": 1, "text": "application comment", "created_at": "2023-11-15T19:30:10.396356Z", "user": 1, "shelter": 1, "application": 1 }]}**
- **Authentication: Respective shelter and seeker only (JWT required)**

**<u>Detailed view of a specific comment</u>**

- **Endpoint: comments/shelters/<int:shelter_id>/comments/<int:comment_id>/**
- **Method: GET**
- **Description: View a specific comment for a shelter**
- **Response:**
  **{"id": 1, "text": "shelter comment", "created_at": "2023-11-15T19:30:10.396356Z", "user": 1, "shelter": 1, "application": null}**
- **Authentication: Logged in user only (JWT required)**

- **Endpoint: comments/applications/<int:application_id>/comments/<int:comment_id>/**
- **Method: GET**
- **Description: View a specific comment for an application**
- **Response:**
  **{"id": 2, "text": "application comment", "created_at": "2023-11-15T19:30:10.396356Z", "user": 1, "shelter": 1, "application": 1}**
- **Authentication: Respective shelter and seeker only (JWT required)**

# Applications:

**Application Model**
Fields:

- seeker_user (foreign key to CustomUser): Refers to the user applying as a pet seeker.
- shelter_user (foreign key to CustomUser): Refers to the user representing a shelter in the application.
- app_status (string, max length 20, default='pending'): Status of the application.
- pet (foreign key to Pets): Refers to the pet related to this application.
- name (string, max length 100): Name of the applicant.
- email (email): Email of the applicant.
- address (text): Address of the applicant.
- pettype (string, max length 100): Type of pet the applicant is interested in.
- petage (integer): Age of the pet the applicant is interested in.
- ownedbefore (string, max length 300): Information about whether the applicant has owned pets before.
- plantoprovide (string, max length 300): Plan to provide care for the pet.
- reason (string, max length 300): Reason for the application.
- emergencycontact (string, max length 12): Emergency contact of the applicant.
- emergencyemail (email): Emergency email of the applicant.
- acknowledge (boolean): Indicates if the applicant acknowledges something.
- created_at (DateTime): Date and time of application creation.
- last_update_time (DateTime): Date and time of the last update to the application.

**Application List and Create View**

- **Endpoint**: applications/
- **Method**: POST
- **Description**: Create a new application
- **Payload**:

```
{ "name": "user",

 "email": "use12@gmail.cpom",

 "address": "333 spadina",

 "pettype": "dog",

 "petage": 10,

 "ownedbefore": "yes",

 "plantoprovide": "yes",

 "reason": "yes",

 "emergencycontact": "123456",
```

"emergencyemail": "rt@gmail.com",

"acknowledge": false,}

**Response**:

{“name": "user",

"email": "use12@gmail.cpom",

"address": "333 spadina",

"pettype": "dog",

"petage": 10,

"ownedbefore": "yes",

"plantoprovide": "yes",

"reason": "yes",

"emergencycontact": "123456",

"emergencyemail": "rt@gmail.com",

"acknowledge": false,

"seeker_user": 1,

"shelter_user": 2,

"pet": 1}

- **Authentication**: AllowAny

**Application Update**

- **Endpoint**:applications /<int:application_id>
- **Method**: PATCH
- **Description**: Update the status of a specific application
- **Payload**: { "app_status": "accepted" } or { "app_status": "denied" } for shelters, { "app_status": "withdrawn" } for seekers
- **Response**: Updated application details
- **Authentication**: JWTAuthentication (Shelter or Seeker)

**Filtered Application List View**

- **Endpoint**: applications/list
- **Method**: GET
- **Description**: Get a list of applications based on user role and status filter
- **Query Parameters**: app_status (accepted/pending/denied/withdrawn), sort_by (created_at/last_update_time)
- **Authentication**: JWTAuthentication (Shelter or Seeker)

**Get Specific Application**

- **Endpoint**: applications`/<int:application_id>/detail`
- **Method**: GET
- **Description**: Get a specific application based on user role and status filter
- **Query Parameters**: app_status (accepted/pending/denied/withdrawn), sort_by (created_at/last_update_time)
- **Authentication**: JWTAuthentication (Shelter or Seeker)

## Notifications:

**Models:**

**Notification Model**

**Fields:**

- **title (string):** The title of the notification.
- **body (text):** The main content or message of the notification.
- **user (foreign key to CustomUser):** The user to whom the notification is addressed. Each notification is associated with a unique user through a foreign key relationship with the CustomUser model.
- **content_type (foreign key to ContentType):** The content type of the related object for the notification. This is used in conjunction with object_id and content_object to link to the specific content being referenced.
- **object_id (positive integer):** The ID of the related object for the notification.
- **content_object (generic foreign key, linking content_type and object_id):** A generic foreign key linking to the specific content being referenced. This allows for a flexible association with various content types.
- **created_at (datetime):** The date and time when the notification was created.
- **is_read (choice field, default 'unread', choices: 'unread', 'read')**: Indicates whether the notification has been read or is still unread.
- **model_url (string):** A URL representing the link to the related model. This can be used to navigate to the specific content mentioned in the notification.

**<u>Notification Detail (List)</u>**

- **Endpoint: /notifications/<int:pk>**
- **Method: GET**
- **Description: Get the notification with the given notification id**
- **Payload:**
  **{ int: 2}**
- **Response:**

```
 {
    "id": 2,
    "title": "New comment on post 9",
    "body": "New comment on your post 9",
    "object_id": 9,
    "created_at": "2023-11-15T18:03:23.678340Z",
    "is_read": "unread",
    "model_url": "/comments/shelters/1/comments/9",
    "user": 1,
    "content_type": 12
    },
```

- **Authentication: Shelter or Seeker (JWT Authentication)**

**<u>Notification Detail (Delete)</u>**

- **Endpoint: /notifications/<int:pk>**
- **Method: DELETE**
- **Description: Delete the notification with the given notification id**
- **Payload:**
  **{ int: 2}**
- **Response: {'detail': Notification deleted successfully'}**
- **Authentication: Shelter or Seeker (JWT Authentication)**

**<u>Notification (All)</u>**

- **Endpoint: '/notifications'**
- **Method: GET**
- **Params: status = read/unread**
- **Description: Get all the notification of current logged in user**
- **Authentication: Shelter or Seeker (JWT Authentication)**