

Accounts

Model

PetUser

- username: The username selected by a seeker or shelter.
- email: The email of the user who is signing up.
- password: The password chosen for the account.
- first_name: The first name of the user (only for pet seekers).
- last_name: The last name of the user (only for pet seekers).
- shelter_name: The name of the shelter (only for pet shelters).
- phone_number: The phone number of the user.
- location: The location of the user.
- preferences: The preferences of the user (only for pet seekers).
- mission_statement: The mission statement of the shelter (only for pet shelters).
- profile_picture: The profile picture of the user.
- is_shelter: The function to determine if a PetUser object is a shelter.
- is_preference: The function to determine if a preference is in the PetUser's preferences

API Endpoints

accounts/token/

Description:

- Login a user and receive a JWT authentication token.

Methods:

- POST

Payloads:

- username
- password

accounts/token/refresh/

Description:

- Refresh a JWT and receive a new one

Methods:

- POST

Payloads:

- refresh: Takes a refresh type JWT

accounts/signup/shelter/

Description:

- Create a pet shelter user.

Methods:

- POST

Payloads:

- username: The username selected by a seeker or shelter.
- email: The email of the user who is signing up.
- password: The password chosen for the account.
- shelter_name: The name of the shelter.
- phone_number: The phone number of the user.
- location: The location of the user.
- mission_statement: The mission statement of the shelter.
- profile_picture: The profile picture of the user.

accounts/signup/seeker/

Description:

- Create a pet seeker user.

Methods:

- POST

Payloads:

- username: The username selected by a seeker or shelter.
- email: The email of the user who is signing up.
- password: The password chosen for the account.
- first_name: The first name of the user.
- last_name: The last name of the user.
- phone_number: The phone number of the user.
- location: The location of the user.
- preferences: The preferences of the user.
- profile_picture: The profile picture of the user.

accounts/update/shelter/

Description:

- Update a pet shelter user profile.
- Retrieves the proper profile through request.user

Methods:

- PATCH/PUT

Payloads:

- email: The email of the user who is signing up.
- password: The password chosen for the account.
- shelter_name: The name of the shelter.
- phone_number: The phone number of the user.
- location: The location of the user.
- mission_statement: The mission statement of the shelter.
- profile_picture: The profile picture of the user.

accounts/update/seeker/

Description:

- Update a pet seeker user profile
- Retrieves the proper profile through request.user

Methods:

- PATCH/PUT

Payloads:

- email: The email of the user who is signing up.
- password: The password chosen for the account.
- first_name: The first name of the user.
- last_name: The last name of the user.
- phone_number: The phone number of the user.
- location: The location of the user.
- preferences: The preferences of the user.
- profile_picture: The profile picture of the user.

accounts/profile/shelter/<int:pk>

Description:

- View the profile of a shelter
- Must be an authenticated user
- Gets which shelter through pk in url

Methods:

- GET

Payloads:

- None

accounts/profile/seeker/<int:pk>

Description:

- View the profile of a seeker
- Must be an authenticated user
- If it is a shelter user, it can only see seekers that applied to a pet listing by the shelter

Methods:

- GET

Payloads:

- None

accounts/list/shelter/

Description:

- View a list of shelters
- Must be an authenticated user

Methods:

- GET

Payloads:

- None

accounts/delete/shelter/

Description:

- Delete a shelter user
- Will receive which shelter to delete through request.user

Methods:

- DELETE

Payloads:

- None

accounts/delete/seeker/

Description:

- Delete a seeker user
- Will receive which seeker to delete through request.user

Methods:

- DELETE

Payloads:

- None

Pet Listings

Model(s)

PetListing

- shelter: The owner of the PetListing.
- name: The name of the pet.
- breed: The breed of the pet.
- age: The age of the pet.
- size: The size of the pet (small, medium, large).
- gender: The gender of the pet (female, male, unknown).
- status: The status of the PetListing (available, pending, withdrawn, adopted).
- colour: The colour of the pet.
- location: The location of the pet.
- description: The description of the pet.
- behaviour: The behaviour of the pet.
- requirements: The requirements for the pet.
- medical_history: The medical history of the pet.
- publication_date: The date in which the listing was made

PetImage

- image: The image file(s) for the pet.
- pet_listing: Pet listings that the images belong to.

API endpoints

pet_listings/

Description:

- Create a pet listing. Must be a request from a shelter.

Methods:

- POST

Payloads:

- Name
- Breed
- Age
- Size
- Gender
- Status
- Colour
- Location
- Description
- Behaviour
- Requirements
- Medical_history
- Images

pet_listings/list/

Description:

- Get all pet listings. Must be a request from a seeker.
- Get all pet listings belonging to a shelter. Must be a request from the shelter.

Methods:

- GET

Payloads:

- None

pet_listings/<int:pk>/

Description:

- Retrieve the pet listing with id = pk. Must be a request from a seeker.
- If it is a request from a shelter, the listing must belong to the shelter.
- Update the pet listing with id = pk. Must be a request from the shelter the listing belongs to.
- Delete the pet listing with id = pk. Must be a request from the shelter the listing belongs to.

Methods:

- GET
- PUT/PATCH
- DELETE

Payloads:

- Name
- Breed
- Age
- Size
- Gender
- Status
- Colour
- Location
- Description
- Behaviour
- Requirements
- Medical_history
- Images

pet_listings/search/

Description:

- Search by filter and/or sort.
- Retrieve all listings that satisfy the filter on the specified sorting order.

Methods:

- GET

Payloads:

- Shelter_name
- Breed
- Status
- Size
- Min_age
- Max_age
- Colour
- Gender
- Sort_by_age
- Sort_by_name
- Sort_by_size
- Sort_in_desc

Applications

Model

Applications

- `pet_listing`: ForeignKey to `PetListing`, representing the pet listing the application is for.
- `pet_name`: CharField storing the name of the pet.
- `pet_seeker`: ForeignKey to `PetUser` for the user applying for the pet.
- `pet_shelter`: ForeignKey to `PetUser` for the shelter offering the pet.
- `all_agree`, `considerate`, `medical`: IntegerFields storing binary choices (yes or no).
- `full_name`, `email`, `phone_number`: CharFields for applicant's contact information.
- `age_range`: CharField with choices for the applicant's age range.
- `living_arrangement`: CharField with choices for the applicant's living situation.
- `have_applied`: IntegerField for if the applicant has previously applied.
- `street_address`, `city`, `province`, `postal_code`: CharFields for the applicant's address.
- `status`: CharField with choices for the application's status (pending, accepted, denied, withdrawn).
- `created_at`, `updated_at`: DateTimeFields for creation and last update timestamps.

API Endpoints

`applications/create/<int:pk>/`

Description:

- Allows pet seekers to create a new application for a pet listing. It captures various details about the applicant and their preferences. A notification is generated for the shelter and the seeker upon successful creation.

Methods:

- POST

Payloads:

- All fields excluding `pet_listing`, `pet_name`, `status`, `pet_seeker`, and `pet_shelter`. This includes the applicant's personal details, living arrangements, preferences, etc.

applications/update/<int:_pk>/seeker/

Description:

- Enables pet seekers to update the status of their application, specifically to mark it as 'withdrawn'. It notifies the shelter about the status change.

Methods:

- PUT/PATCH

Payloads:

- `status`: Only `pending` and `withdrawn` are allowed as choices.

applications/path/<int:pk>/shelter/

Description:

- Used by shelters to update the status of an application, such as marking it as 'accepted' or 'denied'. The related pet seeker is notified about the status change.

Methods:

- PUT/PATCH

Payloads:

- `status`: Choices include `pending`, `accepted`, and `denied`.

applications/list/

Description:

- Lists all applications associated with the authenticated user (pet seeker or shelter). Provides options to filter by status and sort by creation or last update time.

Methods:

- GET

Payloads:

- Not applicable (lists applications based on the user and optional filters like status).

applications/get/<int:pk>/

Description:

- Retrieves detailed information about a specific application, including associated messages.

Methods:

- GET

Payloads:

- Not applicable

applications/<int:pk>/messages/

Description:

- Lists all messages associated with a particular application, providing a communication history between the pet seeker and the shelter.

Methods:

- GET

Payloads:

- Not applicable

applications/<int:pk>/send-message/

Description:

- Allows sending a message related to a specific application. This is used for communication between the pet seeker and the shelter regarding the application.

Methods:

- POST

Payloads:

- content: The text content of the message.

Comments

Model

Comment Model (Abstract Base Class)

- `text`: A text field to store the comment's content.
- `user`: A foreign key linking to the `PetUser` model, indicating the user who posted the comment. It's set to delete the comment if the associated user is deleted.
- `create_time`: A date-time field that automatically records the creation time of the comment.

ShelterComment Model (Inherits from Comment)

- `rating`: An integer field to store a rating, optional and can be left blank.
- `shelter`: A foreign key linking to the `PetUser` model, specifically filtering for users with `shelter_name` not null, indicating the shelter the comment is related to.
- `user`: Redefines the `user` foreign key from the base class, with a related name `shelterComments`.

Reply Model (Inherits from Comment)

- `thread`: A foreign key linking to the `ShelterComment` model, indicating the comment thread this reply belongs to.
- `Meta`: Contains a verbose name plural 'replies'.

API Endpoints

`comments/<int:shelter_pk>/`

Description:

- Lists all comments for a specific shelter (`shelter_pk`). Requires authentication.
- Creates a new comment for a shelter. Requires authentication.

Methods:

- GET (list comments), POST (create a new comment)

Payloads:

- text: The content of the comment.
- user: Automatically assigned from the authenticated user.
- shelter: Automatically assigned based on `shelter_pk` from the URL.
- rating (Optional): An integer rating value.

comments/<int:shelter_pk>/<int:pk>/

Description:

- Updates the `is_read` status of a specific notification for the authenticated user to indicate a message is read.

Methods:

- PUT/PATCH

Payloads:

- `is_read`: Updated read status of the notification.

comments/<int:shelter_pk>/<int:pk>/reply/

Description:

- Creates a reply to a specific shelter comment (`pk`) within a particular shelter (`shelter_pk`). Requires authentication.

Methods:

- POST

Payloads:

- text: The content of the reply.
- user: Automatically assigned from the authenticated user.
- thread: Automatically assigned based on `pk` of the comment being replied to.

Notification

Notification

- `user_id`: A foreign key linking to the `PetUser` model, representing the user associated with the notification. It's set to delete the notification if the associated user is deleted.
- `message`: A character field storing the message of the notification, with a maximum length of 128 characters.
- `is_read`: A boolean field indicating whether the notification has been read. Defaults to `False`.
- `created_at`: A date-time field that records when the notification was created. It's automatically set when a notification is created (`auto_now_add=True`).
- `related_link`: A character field storing a related link, if any, with a maximum length of 255 characters. It's optional and can be left blank or null.

API endpoints

Notifications/

Description:

- Retrieves a paginated list of notifications for the authenticated user.

Methods:

- GET

Payload:

- Filtering by `is_read` status via query parameters.
- Pagination parameters (page number, page size).

Notifications/<int:pk>/

Description:

- Updates the `is_read` status of a specific notification for the authenticated user.

Methods:

- PUT/PATCH

Payload:

- `is_read`: Updated read status of the notification.

Notifications/<int:pk>/delete/

Description:

- Deletes a specific notification for the authenticated user.

Methods:

- DELETE

Payload:

- None required (uses the notification's primary key from the URL).

Final notes:

During the early stages of development, we accidentally committed a few migration files, which might cause potential problems. We have referenced multiple sources and booked office hours, which all resulted in not-so-ideal solutions, meaning we would not be able to get rid of the migration files. Thus, we decided to put it as a side note here.