

PHASE 2

Models

User:

- Inherit username, password, email, first name, last_name from base user class
- Avatar profile picture
- Location (string for now)
- Phone number (string for now, limited to 10 chars, validated on create/update)
- isShelter (boolean) // just to distinguish, all properties are the same

Pet Listing :

- Name
- Breed
- Age
- Size
- Color
- Sex
- Photos
- Description/behaviour
- Medical history
- Listing status (available, adopted, pending, or withdrawn)
- Publication date
- Location
- Special needs/ other notes
- Owner (foreignkey to user, many-to-one, required, not null)

Comment:

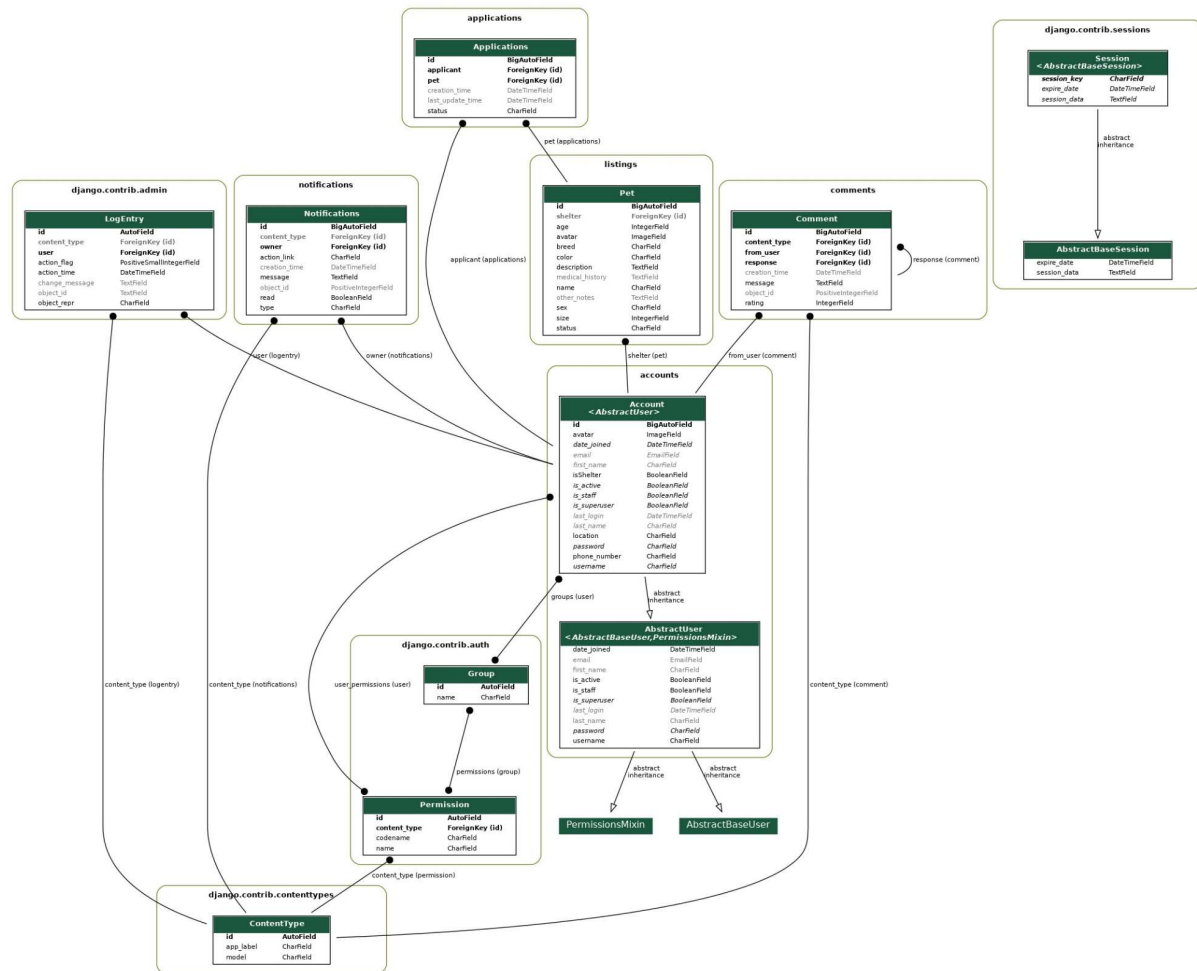
- message
- from_user (foreignkey to user, required, not null, one-to-many)
- content_object (generic foreign key to either application or user, required, not null)
 - Serves as the "to" field
 - **Docs linked on the P2 instructions**
- rating (max_value = 5, nullable, blank = false)
- response (foreignkey to another comment, nullable, blank = false)
- creation_time

Application:

- Creation_time
- Last_update_time (When an application receives a new comment, its "last update time" should be changed)
- Listing (foreignkey to pet listing, many-to-one, required, not null)
- Applicant (foreignkey to user, required, not null)
- Status (pending, accepted, denied, withdrawn)

Notification:

- Owner (foreignkey to user, required, not null, many-to-one)
- Creation time
- Message
- read (boolean, default false)
- Action_link
- type(new comment, new application, status update)
- Content_object (generic foreign key to either comment or application, required, not null)



Endpoints

Endpoint: accounts/login

Methods: POST

Payload: username, password

Description: logs in a registered user and returns the JWT auth token

Example Response: {"token": "xxx", "refresh": "xxx"}

Endpoint: accounts/register

Methods: POST

Payload: username, password1, password2, first_name, last_name, email, isShelter

Description: Registers a new user based on isShelter, with the provided credentials

Example Response: {"username": "john1", "first_name": "john", "last_name": "doe"}

Endpoint: accounts/<int:pk>

Methods: GET

Payload: N/A

Description: returns the profile data of the user with id = pk. Any user can view a shelter, but shelters can only view profiles of users who have an application with them

Example Response: {"username": "john1", "first_name": "john", "last_name": "doe", "location": "toronto", "phone_number": "41612345678"}

Endpoint: accounts/<int:pk>

Methods: PATCH

Payload: {username, first_name, last_name, profile_picture, location, phone_number, password1, password2}

Description: updates the profile of the currently logged in user, returns the updated profile

Example Response:: {"username": "john1", "first_name": "john", "last_name": "doe", "location": "toronto", "phone_number": "41612345678"}

Endpoint: accounts/<int:pk>

Methods: DELETE

Payload: N/A

Description: deletes the user with the given id. If a shelter, then all the pet listings will be deleted. If a seeker, then all their applications will be deleted. All notifications will be deleted from the database regardless of the user.

Example Response: HTTP_204

Endpoint: accounts/all

Methods: GET

Payload: N/A

Description: returns a list of all shelters

Example Response: [{"first_name": "fname", "last_name": "", "email": "shelter@gmail.com", "location": "55 queen street"}]

Endpoint: listings/

Methods: POST

Payload: {name, breed, sex, age (months), size (lbs), color, photos, description, medical history, other_notes}

Description: creates a new listing under the logged in shelter with the given payload. Returns the new listing object

Example Response: {"id": 1, "name": "pup", "breed": "corgi", "sex": "male", "age": 5, "size": 5, "color": "beige", "description": "pup is a good boi", "medical history": "", "other_notes": ""}

Endpoint: listings/<int:pk>

Methods: PATCH

Payload: {id, name, breed, sex, age (months), size (lbs), color, photos, description, medical history, other_notes}

Description: updates the corresponding listing with the provided id. Returns the updated listing model

Example Response: {"id": 1, "name": "pup", "breed": "corgi", "sex": "male", "age": 5, "size": 5, "color": "beige", "description": "pup is a good boi", "medical history": "", "other_notes": ""}

Endpoint: listings/<int:pk>

Methods: DELETE

Payload: {listing_id}

Description: deletes the listing with id = listing_id. All associated applications and notifications will also be deleted. Returns the model that was deleted

Example Response: {"id": 1, "name": "pup", "breed": "corgi", "sex": "male", "age": 5, "size": 5, "color": "beige", "description": "pup is a good boi", "medical history": "", "other_notes": ""}

Endpoint: listings/<int:pk>

Methods: GET

Payload: N/A

Description: Returns the listing with id = listing_id

Example Response: {"id": 1, "name": "pup", "breed": "corgi", "sex": "male", "age": 5, "size": 5, "color": "beige", "description": "pup is a good boi", "medical history": "", "other_notes": ""}

Endpoint: listings/all/?page

Methods: GET

Payload: {shelter, status, age, gender, sort}

Description: Return a list of listings with the given filters applied

Example Response:[{"id": 1, "name": "pup", "breed": "corgi", "sex": "male", "age": 5, "size": 5, "color": "beige", "description": "pup is a good boi", "medical history": "", "other_notes": ""}]

Endpoint: comments

Methods: POST

Payload: {content_type('user' or 'applications'), object_id(int), message(str), rating(int:[1, 5])}

Description: Creates a new comment from the logged in user on the object specified using object_id. Returns the created comment.

Example Response: {"id": 1, "message": "this shelter sucks", "rating": 1, "response_id": NULL}

Endpoint: comments/

Methods: GET

Params: {sort(int), page(int), object_id(int), for_shelter(int)}

Description: returns a list of comments on a shelter or an application

Example Response: [{"id": 1, "message": "this shelter sucks", "rating": 1, "response_id": NULL, "to": "toronto animal shelter", "from": "john doe"}]

Endpoint: applications

Methods: POST

Payload: {pet}

Description: Creates an application from the logged in pet seeker for the provided pet listing_id. Can only create applications for a pet listing that is "available". Returns the newly created application object

Example Response: {"id": 1, "created_time": timestamp, "last_updated": timestamp, "owner_id": 1, "listing_id": 5, "status": "pending"}

Endpoint: applications/<int:pk>

Methods: PATCH

Payload: {status}

Description: Updates the status of the application with id = pk. Shelter can only update the status of an application from pending to accepted or denied. Pet seeker can only update the status of an application from pending or accepted to withdrawn. Returns updated application model.

Example Response: {"id": 1, "created_time": timestamp, "last_updated": timestamp, "owner_id": 1, "listing_id": 5, "status": "pending"}

Endpoint: applications/<int:pk>

Methods: GET

Payload: N/A

Description: Return the application with id = pk. Only the owner or the recipient of the application can view the application.

Example Response: {"id": 1, "created_time" : *timestamp*, "last_updated":*timestamp*, "owner_id" : 1, "listing_id" : 5, "status" : "pending"}

Endpoint: applications/all/<str:status>/?page

Methods: GET

Payload: N/A

Description: Returns a list of applications that the current seeker made or that are made to the current shelter. Shelters can only view their own applications, not that of other shelters. Sorted by creation time then last updated time

Example Response:

```
{
  "Count": 5,
  "Next": "url",
  "Previous" : NULL,
  "results": [{"id": 1, "created_time" : timestamp, "last_updated":timestamp, "owner_id" : 1,
    "listing_id" : 5, "status" : "pending"}, {"id": 1, "created_time" : timestamp,
    "last_updated": timestamp, "owner_id" : 1, "listing_id" : 5, "status" : "pending"}]
}
```

Endpoint: notifications

Methods: POST

Payload: {owner(int), message, action_link, type('new_comment', 'new_applciation', 'status_update')}

Description: Creates a new unread notification for user_id of the specified type. Returns the newly created notification object

Example Response: {"id" : 1, "message" : "status updated", "owner_id" : 5, "read": false, "creation_time" : *timestamp*, "type" : "status update", "action_link" : "/application/view"}

Endpoint: notificatons/

Methods: GET

Params: {read(int), sort(int), page(int)}

Description: Returns a list of the current user's notifications based on the provided read status sorted by creation time.

Example Response:[{"id" : 1, "message" : "status updated", "owner_id" : 5, "read": false, "creation_time" : *timestamp*, "type" : "status update", "action_link" : "/application/view"}, {"id" : 1, "message" : "status updated", "owner_id" : 5, "read": false, "creation_time" : *timestamp*, "type" : "status update", "action_link" : "/application/view"}]

Endpoint: notifications/<int:pk>

Methods: DELETE

Payload: {notifcation_id}

Description: Deletes the current user's notification with the provided id. Returns the delete notification

Example Response: 204 For success deletion

Endpoint: notifications/<int:pk>

Methods: GET

Payload: {notification_id}

Description: Returns the current user's notification with the provided id

Example Response: {"id" : 1, "message" : "status updated", "owner_id" : 5, "read": false, "creation_time" : *timestamp*, "type" : "status update", "action_link" : "/application/view"}

Endpoint: notifications/<int:pk>

Methods: PATCH

Payload: {notification_id}

Description: Update the notification status from read:False to read:True

Example Response: {"id" : 1, "message" : "status updated", "owner_id" : 5, "read": true, "creation_time" : *timestamp*, "type" : "status update", "action_link" : "/application/view"}