Pet Store REST API

A complete Flask REST API implementation for managing a pet store inventory.

Project Structure

```
pet-store-api/
├── app.py                # Main Flask application
├── models.py             # Data models and in-memory storage
├── utils.py              # Utility functions (Ninja API, image handling)
├── routes/
│   ├── __init__.py
│   ├── pet_types.py      # /pet-types routes
│   ├── pets.py           # /pet-types/{id}/pets routes
│   └── pictures.py       # /pictures routes
├── pictures/             # Directory for storing pet images
└── requirements.txt      # Python dependencies
```

Setup Instructions

1. Install Dependencies

```bash
pip install -r requirements.txt
```

2. Configure API Key

Open `utils.py` and replace `YOUR_API_KEY_HERE` with your actual Ninja API key:

```python
NINJA_API_KEY = "your_actual_api_key_here"
```

You can get a free API key at: https://api-ninjas.com/

3. Create Directory Structure

Make sure to create the `routes` directory with an `__init__.py` file:

```bash
mkdir routes
touch routes/__init__.py
```

Then place the route files in the `routes` directory:

- `pet_types.py`
- `pets.py`
- `pictures.py`

4. Run the Application

```bash
python app.py
```

The API will be available at `http://localhost:5001`

API Endpoints

Pet Types

- **POST /pet-types** - Create a new pet type
  - Payload: `{"type": "Poodle"}`
- **GET /pet-types** - Get all pet types
  - Query params: `family`, `genus`, `type`, `id`, `lifespan`, `hasAttribute`
- **GET /pet-types/{id}** - Get specific pet type
- **DELETE /pet-types/{id}** - Delete pet type (only if no pets)

Pets

- **POST /pet-types/{id}/pets** - Add a pet
  - Payload: `{"name": "Buddy", "birthdate": "15-03-2020", "picture-url": "http://..."}`
- **GET /pet-types/{id}/pets** - Get all pets of a type
  - Query params: `birthdateGT`, `birthdateLT`
- **GET /pet-types/{id}/pets/{name}** - Get specific pet
- **PUT /pet-types/{id}/pets/{name}** - Update pet
- **DELETE /pet-types/{id}/pets/{name}** - Delete pet

Pictures

- **GET /pictures/{filename}** - Get pet picture

Testing Examples

Create a pet type:

```bash
curl -X POST http://localhost:5001/pet-types \
  -H "Content-Type: application/json" \
  -d '{"type": "Golden Retriever"}'
```

Get all pet types:

```bash
curl http://localhost:5001/pet-types
```

Add a pet:

```bash
curl -X POST http://localhost:5001/pet-types/1/pets \
  -H "Content-Type: application/json" \
  -d '{"name": "Buddy", "birthdate": "15-03-2020"}'
```

Query with filters:

```bash
curl "http://localhost:5001/pet-types?family=Canidae"
curl "http://localhost:5001/pet-types?hasAttribute=intelligent"
curl "http://localhost:5001/pet-types/1/pets?birthdateGT=01-01-2020"
```

Features Implemented

✅ All REST endpoints as specified
✅ Ninja Animals API integration
✅ Image downloading and storage
✅ Query string filtering
✅ Proper error handling with correct status codes
✅ Case-insensitive string comparisons
✅ Date parsing and comparison
✅ Unique ID generation (never reused)
✅ Unique pet names per type
✅ Blueprint-based routing
✅ In-memory data storage

Notes

- All data is stored in memory and will be lost when the server restarts

- Images are stored in the `pictures/` directory
- All string comparisons are case-insensitive
- Pet type IDs are never reused, even after deletion
- Pet names must be unique within a pet type