

1. List All Pets

- **Method:** GET
- **Path:** /pets
- **Description:** Retrieves a list of all available pets.
- **Input:** None
- **Output:**
 - **Status Code:** 200 OK
 - **Response Body:** List of pet objects
 - id (integer): Pet ID
 - name (string): Name of the pet
 - type (string): Type of pet (e.g., Dog, Cat)
 - age (integer): Age of the pet

2. Get Pet by ID

- **Method:** GET
- **Path:** /pets/<pet_id>
- **Description:** Retrieves details of a specific pet by ID.
- **Input:**
 - **Path Parameter:** pet_id (integer): ID of the pet
- **Output:**
 - **Status Code:** 200 OK or 404 Not Found
 - **Response Body** (if 200 OK):
 - id (integer): Pet ID
 - name (string): Name of the pet
 - type (string): Type of pet
 - age (integer): Age of the pet
 - **Response Body** (if 404 Not Found): Error message

3. Add a New Pet

- **Method:** POST
- **Path:** /pets
- **Description:** Adds a new pet to the list.
- **Input:**
 - **Request Body:**
 - name (string): Name of the pet
 - type (string): Type of pet
 - age (integer): Age of the pet
- **Output:**
 - **Status Code:** 201 Created
 - **Response Body:**
 - id (integer): New Pet ID
 - name (string): Name of the pet

- **type** (string): Type of pet
- **age** (integer): Age of the pet

4. Update Pet Information

- **Method:** PUT
- **Path:** /pets/<pet_id>
- **Description:** Updates information for a specific pet.
- **Input:**
 - **Path Parameter:** pet_id (integer): ID of the pet
 - **Request Body** (optional fields):
 - **name** (string): Updated name of the pet
 - **type** (string): Updated type of pet
 - **age** (integer): Updated age of the pet
- **Output:**
 - **Status Code:** 200 OK or 404 Not Found
 - **Response Body** (if 200 OK):
 - Updated pet object with id, name, type, and age
 - **Response Body** (if 404 Not Found): Error message

5. Delete a Pet by ID

- **Method:** DELETE
- **Path:** /pets/<pet_id>
- **Description:** Deletes a specific pet by ID.
- **Input:**
 - **Path Parameter:** pet_id (integer): ID of the pet
- **Output:**
 - **Status Code:** 200 OK or 404 Not Found
 - **Response Body** (if 200 OK): Success message
 - **Response Body** (if 404 Not Found): Error message

6. Get User Information

- **Method:** GET
- **Path:** /users/<user_id>
- **Description:** Retrieves information about a specific user.
- **Input:**
 - **Path Parameter:** user_id (integer): ID of the user
- **Output:**
 - **Status Code:** 200 OK or 404 Not Found
 - **Response Body** (if 200 OK):
 - **id** (integer): User ID
 - **name** (string): Name of the user
 - **email** (string): Email of the user

- **adoption_progress** (list): List of adoption progress records, each with:
 - **pet_id** (integer): Pet ID
 - **status** (string): Status of the adoption (e.g., Pending, Approved, Denied)
- **Response Body** (if **404 Not Found**): Error message

7. View User's Adoption Progress

- **Method:** GET
- **Path:** /users/<user_id>/adoption-progress
- **Description:** Retrieves adoption progress details for a specific user.
- **Input:**
 - **Path Parameter:** **user_id** (integer): ID of the user
- **Output:**
 - **Status Code:** 200 OK or 404 Not Found
 - **Response Body** (if 200 OK): List of adoption progress entries
 - Each entry includes:
 - **pet_id** (integer): Pet ID
 - **status** (string): Adoption status
 - **Response Body** (if 404 Not Found): Error message

8. Manager Access to User Information

- **Method:** GET
- **Path:** /manager/users/<user_id>
- **Description:** Allows the manager to retrieve detailed information about a user.
- **Input:**
 - **Path Parameter:** **user_id** (integer): ID of the user
- **Output:**
 - **Status Code:** 200 OK or 404 Not Found
 - **Response Body** (if 200 OK): User object with all fields as described above
 - **Response Body** (if 404 Not Found): Error message