

## People Objects

People objects are returned with the following properties.

Property	Type	Description
id	String	url safe identifier
self	Url String	link to self (not an intrinsic property of People objects; added when request is made)
fname	String	first name of person
lname	String	last name of person
age	Integer	age of person (years)
address	String	home address of person
pets	String[]	a list containing the ids of pets for which this person is caretaker for

## Pet Objects

Pet objects are returned with the following properties.

Property	Type	Description
id	String	url safe identifier
self	Url String	link to self (not an intrinsic property of Pet objects; added when request is made)
name	String	name of pet
species	String	species of pet (cat, dog, etc.)
age	Integer	age of pet (years)
weight	Integer	weight of pet
caretaker	String	id of the person who is caretaker for this pet, None if the pet does not currently have a caretaker

## Get all People and Pets

GET /

**Note:** Returns a json object containing "People" (a json list of all People in the database) and "Pets" (a json list of all Pets in the database).

## Get all People

GET /person

**Note:** Returns a json list of all People in the database.

## Get a single Person

GET /person/:id

**Note:** A request containing a valid id string which does not identify a Person in the database will return a 404 status code. A request containing an invalid id string will return a status code 500 server error.

## Get all Pets

GET /pet

**Note:** Returns a json list of all Pets in the database.

## Get single Pet

GET /pet/:id

**Note:** A request containing a valid id string which does not identify a Pet in the database will return a 404 status code. A request containing an invalid id string will return a status code 500 server error.

## Create a new Person

POST /person

**Input :** application/json

Name	Type	Description
fname	string	<b>Required.</b> First name of Person
lname	string	<b>Required.</b> Last name of Person
age	integer	<b>Required.</b> Age of Person (years)
address	integer	<b>Required.</b> Address of Person

**Note:** A request which contains invalid body data will return a status code 400 Bad Request.

## Create a new Pet

POST /pet

**Input :** application/json

Name	Type	Description
name	String	<b>Required.</b> Name of pet
species	String	<b>Required.</b> Species of Pet
age	integer	<b>Required.</b> Age of Pet (years)
weight	integer	<b>Required.</b> Weight of Pet (lbs)

**Note:** A request which contains invalid body data will return a status code 400 Bad Request.

## Modify a Person

PATCH /person/:id

**Input :** application/json

Name	Type	Description
age	integer	New age of Person (years)
address	string	New address of Person

**Note:** The id, fname, lname, and pets Person attributes cannot be modified through this request.

## Modify a Pet

PATCH /pet/:id

**Input :** application/json

Name	Type	Description
age	integer	New age of Pet (years)
weight	integer	New weight of Pet (lbs)

**Note:** The id, name, species and caretaker Pet attributes cannot be modified through this request.

## Delete a Person

DELETE /person/:id

Note: The caretaker attribute of all Pets for which this Person was caretaker for is set to **None** upon the deletion of this Person.

## Delete a Pet

DELETE /pet/:id

Note: This Pet's id string is removed from a Person's `pets[]` attribute upon deletion of this Pet, if said person was caretaker for this Pet.

## Add Care Relationship

PUT /pet/:id/caretaker

Input : application/json

Name	Type	Description
person_id	string	<b>Required.</b> Person id string of the Person who will be caretaker for this Pet.

Note: If the Pet identified by the `id` in the url already has a caretaker, the `id` in the url does not match a Pet in the database, or the `id` passed in the request body does not match a person in the database, a 400 error code is returned along with an appropriate error message. If the request is successful the Pet's **caretaker** is updated, along with the Person's **pets** list.

## Remove Care Relationship

PATCH /pet/:id/caretaker

Note: If the `id` passed in the url does not identify a Pet in the database, or the person `id` of this Pets **caretaker** does not identify a Person in the database, a 400 error is returned with the appropriate message. If the request is successful, the Pet's **caretaker** attribute is set to **None** and the Pet's `id` string is removed from the Person's **pets** list.