Portfolio Project: API Spec

CS 493: Cloud Application Development
Fall 2022
Oregon State University
Alexander Kim, kima4
Last Update: 12/4/2022, 2:40 pm Eastern

Application URL

This URL is where the application is deployed and it also contains a link for account creation or log in.
https://kima4-cs493-final.ue.r.appspot.com/

Change Log

Version	Change	Date
1.0	Initial Version	Dec 4, 2022

Data Model

The app stores three types of entities in the database: Users, Pets, and Schools

Users

Property	Data Type	Required?	Valid Values / Notes
id	Integer	Yes	Valid values are integers that have been automatically generated by Datastore. Automatically generated by Datastore. Cannot be viewed normally to avoid confusion with owner_id and cannot be modified.
owner_id	String	Yes	Valid values are strings with ‘Owner’ followed by the ‘sub’ property of the JWT. User’s unique ID derived from the JWT for authentication and authorization. Cannot be modified.
pets	Array	Yes	Valid values are arrays containing id values of the user’s pets. Array containing pet id values corresponding to pets the user owns. Cannot be directly modified.

Pets

Property	Data Type	Required?	Valid Values / Notes
id	Integer	Yes	Valid values are integers that have been automatically generated by Datastore. The ID of the pet. Cannot be modified or manually entered.
name	String	Yes	Valid values are strings between 1 and 50 characters, inclusive, that contain only letters, spaces, periods, dashes, and apostrophes. The name of the pet.
breed	String	Yes	Valid values are strings between 1 and 30 characters, inclusive, that contain only letters, spaces, periods, dashes, and apostrophes. The breed/species of the pet.
age	Integer	Yes	Valid values are integers between 0 and 150, inclusive. The age of the pet.
owner	String	Yes	Valid values are strings with 'Owner' followed by the 'sub' property of the owner's JWT. The unique ID of the pet's owner. Cannot be changed or manually set.
school	JSON Object	No	Valid values are JSON objects with two properties: id and name. id is the automatically generated integer ID number of the school and name is the name of the school. If the pet is not attending a school, the value of school is instead null. Information about the school that the pet attends. Cannot be directly modified.

Schools

Property	**Data Type**	**Required?**	**Valid Values / Notes**
id	Integer	Yes	Valid values are integers that have been automatically generated by Datastore. The ID of the school. Cannot be modified or manually entered.
name	String	Yes	Valid values are strings between 1 and 50 characters, inclusive, that contain only alphanumeric characters, spaces, periods, commas, dashes, and apostrophes. The name of the school.
location	String	Yes	Valid values are strings between 1 and 80 characters, inclusive, that contain only alphanumeric characters, spaces, periods, commas, dashes, and apostrophes. The location of the school.
headmaster	Integer	Yes	Valid values are strings between 1 and 50 characters, inclusive, that contain only letters, spaces, periods, dashes, and apostrophes. The name of the headmaster of the school
students	Array	Yes	Valid values are arrays containing id values of the pets enrolled at the school. Array containing pet id values corresponding to pets that attend the school. Cannot be directly modified.

Relationship between Non-User Entities

In this app, Pets have a many-to-one relationship with Schools. Each pet can only be enrolled at most one school at any given time, whereas each school can enroll multiple pets. To model this, the pet.school property can take a single JSON object that represents the school that it attends. On the school side, the school.students property is an array that can contain the pet.id values of the pets that are enrolled in it.

Modeling the User Entity

In this app, identification is done using the JWT. The unique identifier for each user is the word ‘Owner’ followed by the sub property of the JWT (i.e. Owner{{JWT}}). Therefore, requests that need a user identifier should have an authorization header with a Bearer Token that contains the JWT. This unique identifier is stored in Datastore as the owner_id property and is used in lieu of the automatically generated id value for most purposes. This means that the user identity is mapped to a supplied JWT by checking the sub value of the JWT and determining to which user entity that sub value belongs.

The relationship between Pets and Users is another many-to-one relationship. Each pet can only belong to one owner, whereas each user can own multiple pets. To model this, the pet.owner property takes a string, the unique identifier for each user. The user entity uses the same model as the school entity, in which user.pets is an array that can contain the pet.id values of the pets that the user owns.

API Specifications - Users

Attempting to POST, PATCH, PUT, or DELETE to /users will result in a 405 status code.

Get All Users [Unprotected]

Gets a list of all users. Shows the unique ID for each user, as well as the number of pets each user owns. The number of pets property is in lieu of listing the pet ID numbers.

```
GET /users
```

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns an array of JSON objects, each representing a user.
Failure	406 Not Acceptable	The server is unable to send the response in the requested type. Only application/json is acceptable.

Response Examples

Success

```
Status: 200 OK

[
  {
    ‘owner_id’: ‘Owner12345’,
    ‘num_pets’: 6
  },
  {
    ‘owner_id’: “Owner67890”,
    ‘num_pets’: 2
  },
  {
    ‘owner_id’: ‘Owner24680’,
    ‘num_pets’: 0
  }
]
```

Failure

```
Status: 406 Not Acceptable

{
  ‘Error’: ‘The requested response type must be application/json’
}
```

Get a User [Protected]

Gets information for the requested user if the appropriate JWT is supplied.

```
GET /users/:owner_id
```

Request

Path Parameters

Name	Description
owner_id	The unique identifier for each user, derived from the JWT.

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns a JSON object, representing a user and their pets.
Failure	401 Unauthorized	The JWT is either missing or invalid.
Failure	403 Forbidden	The JWT does not correspond to the user page that is being accessed.
Failure	406 Not Acceptable	The server is unable to send the response in the requested type. Only application/json is acceptable.

Response Examples

Success

```
Status: 200 OK

{
  ‘owner_id’: ‘Owner67890’,
  ‘pets’: [
    {
      ‘id’: 21212,
      ‘name’: ‘George’,
      ‘breed’: ‘Husky’,
      ‘age’: 3,
      ‘school’: {
        ‘id’: 11111,
        ‘name’: ‘Super School’
      },
      ‘self’:
        ‘https://kima4-cs493-final.ue.r.appspot.com/pets/21212’
    },
    {
      ‘id’: 32323,
      ‘name’: ‘Sam’,
      ‘breed’: ‘Rat’,
      ‘age’: 1,
      ‘school’: null,
      ‘self’:
        ‘https://kima4-cs493-final.ue.r.appspot.com/pets/21212’
    }
  ]
}
```

Failure

```
Status: 401 Unauthorized

{
  ‘Error’: ‘The JWT is missing or invalid’
}
```

```
Status: 403 Forbidden

{
  ‘Error’: ‘You cannot view another person’s information page’
}
```

```
Status: 406 Not Acceptable

{
  ‘Error’: ‘The requested response type must be application/json’
}
```

API Specifications - Pets

Attempting to PATCH, PUT, or DELETE to /pets will result in a 405 status code.

Create a Pet [Protected]

Creates a new pet. Attributes other than name, breed, and age are ignored. The school value is set to null by default.

```
POST /pets
```

Request

Path Parameters
None

Request Body
Required

Request Body Format
JSON

Request JSON Attributes

Name	Description	Required?
name	Name of the pet. Can only contain letters, spaces, periods, dashes, and apostrophes and must be between 1 and 50 characters, inclusive.	Yes
breed	Breed or species of the pet. Can only contain letters, spaces, periods, dashes, and apostrophes and must be between 1 and 30 characters, inclusive.	Yes
age	Age of the pet in years. Must be between 0 and 150, inclusive.	Yes

Request Body Example

```
{
  ‘name’: ‘Mr. Squishy’,
  ‘breed’: ‘Sea Cucumber’,
  ‘age’: 2
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	Returns the JSON object that represents the pet.
Failure	400 Bad Request	The request body is missing one or more required attributes or one or more given attributes have invalid values.
Failure	401 Unauthorized	The JWT is either missing or invalid.
Failure	406 Not Acceptable	The server is unable to send the response in the requested type. Only application/json is acceptable.

Response Examples

Success

```
Status: 201 Created

{
  ‘id’: 45454,
  ‘name’: ‘Mr. Squishy’,
  ‘breed’: ‘Sea Cucumber’,
  ‘age’: 2,
  ‘self’: ‘https://kima4-cs493-final.ue.r.appspot.com/pets/45454’,
  ‘school’: null,
  ‘owner’: ‘Owner12345’
}
```

Failure

```
Status: 400 Bad Request

{
  ‘Error’: ‘The request object is missing at least one of the
required attributes or at least one of the attributes has an
invalid value’
}
```

```
Status: 401 Unauthorized

{
  ‘Error’: ‘The JWT is missing or invalid’
}
```

```
Status: 406 Not Acceptable

{
  ‘Error’: ‘The requested response type must be application/json’
}
```

Get All Pets [Protected]

Gets a list of pets that belong to the user. Response is paginated with a maximum of five pets per page.

```
GET /pets?page=:page
```

Request

Path Parameters

None

Query Parameters

Name	**Description**	**Required?**
page	Page number for pagination. If a page is not given, defaults to the first page, i.e. the first five results.	No

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	**Status Code**	**Notes**
Success	200 OK	Returns a JSON object with the property ‘pets’ which contains a list of the information of up to five pets. There is also a property ‘total_pets’ that displays the total number of pets that the user owns. If there are more pets that have not been retrieved, there will also be a ‘next’ key that contains the URL for the next set of pets to be retrieved.
Failure	401 Unauthorized	The JWT is either missing or invalid.
Failure	406 Not Acceptable	The server is unable to send the response in the requested type. Only application/json is acceptable.

Response Examples

Success

```
Status: 200 OK

{
  ‘pets’: [
    {
      ‘id’: 12121,
      ‘name’: ‘Jon’,
      ‘breed’: ‘Squirrel’,
      ‘age’: 2,
      ‘owner’: ‘Owner12345’,
      ‘school’: null
      ‘self’:
        ‘https://kima4-cs493-final.ue.r.appspot.com/pets/12121’
    },
    {
      ‘id’: 23232,
      ‘name’: ‘Sam’,
      ‘breed’: ‘Rabbit’,
      ‘age’: 3,
      ‘owner’: ‘Owner12345’,
      ‘school’: null
      ‘self’:
        ‘https://kima4-cs493-final.ue.r.appspot.com/pets/23232’
    },
    {
      ‘id’: 34343,
      ‘name’: ‘Tyler’,
      ‘breed’: ‘Guinea Pig’,
      ‘age’: 1,
      ‘owner’: ‘Owner12345’,
      ‘school’: null
      ‘self’:
        ‘https://kima4-cs493-final.ue.r.appspot.com/pets/34343’
    },
    {
      ‘id’: 45454,
      ‘name’: ‘Mr. Squishy’,
      ‘breed’: ‘Sea Cucumber’,
      ‘age’: 2,
      ‘owner’: ‘Owner12345’,
      ‘school’: null
      ‘self’:
        ‘https://kima4-cs493-final.ue.r.appspot.com/pets/45454’
    },
    {
      ‘id’: 56565,
      ‘name’: ‘Megan’,
      ‘breed’: ‘Golden Eagle’,
      ‘age’: 6,
      ‘owner’: ‘Owner12345’,
      ‘school’: null
      ‘self’:
        ‘https://kima4-cs493-final.ue.r.appspot.com/pets/56565’
    }
  ],
  ‘total_pets’: 6,
  ‘next’: ‘https://kima4-cs493-final.ue.r.appspot.com/pets?page=2’
}
```

Failure

```
Status: 401 Unauthorized

{
  ‘Error’: ‘The JWT is missing or invalid’
}
```

```
Status: 406 Not Acceptable

{
  ‘Error’: ‘The requested response type must be application/json’
}
```

Get a Pet [Protected]

Gets an existing pet.

```
GET /pets/:pet_id
```

Request

Path Parameters

Name	Description
pet_id	ID of the pet.

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns a JSON object that represents the pet.
Failure	401 Unauthorized	The JWT is either missing or invalid.
Failure	403 Forbidden	The pet belongs to a different user.
Failure	404 Not Found	A pet with the supplied ID was not found.
Failure	406 Not Acceptable	The server is unable to send the response in the requested type. Only application/json is acceptable.

Response Examples

Success

```
Status: 200 OK

{
  ‘id’: 12121,
  ‘name’: ‘Jon’,
  ‘breed’: ‘Squirrel’,
  ‘age’: 2,
  ‘owner’: ‘Owner12345’,
  ‘school’: null
  ‘self’: ‘https://kima4-cs493-final.ue.r.appspot.com/pets/12121’
}
```

Failure

```
Status: 401 Unauthorized

{
  ‘Error’: ‘The JWT is missing or invalid’
}
```

```
Status: 404 Forbidden

{
  ‘Error’: ‘The pet belongs to someone else’
}
```

```
Status: 404 Not Found

{
  ‘Error’: ‘No pet with this pet_id exists’
}
```

```
Status: 406 Not Acceptable

{
  ‘Error’: ‘The requested response type must be application/json’
}
```

Patch a Pet [Protected]

Patches an existing pet. Only name, breed, and age can be modified. Other properties are ignored or will return error codes.

```
PATCH /pets/:pet_id
```

Request

Path Parameters

Name	Description
pet_id	ID of the pet.

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Name of the pet. Can only contain letters, spaces, periods, dashes, and apostrophes and must be between 1 and 50 characters, inclusive.	No
breed	Breed or species of the pet. Can only contain letters, spaces, periods, dashes, and apostrophes and must be between 1 and 30 characters, inclusive.	No
age	Age of the pet in years. Must be between 0 and 150, inclusive.	No

Request Body Example

```
{
  ‘name’: ‘Jordan’
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns a JSON object that represents the pet.
Failure	400 Bad Request	One or more attributes in the request body have invalid values.
Failure	400 Bad Request	The request body tries to change the school property.
Failure	400 Bad Request	The request body tries to change the owner property.
Failure	401 Unauthorized	The JWT is either missing or invalid.
Failure	403 Forbidden	The pet belongs to a different user.
Failure	404 Not Found	A pet with the supplied ID was not found.
Failure	406 Not Acceptable	The server is unable to send the response in the requested type. Only application/json is acceptable.

Response Examples

Success

```
Status: 200 OK

{
  ‘id’: 34343,
  ‘name’: ‘Jordan’,
  ‘breed’: ‘Guinea Pig’,
  ‘age’: 1,
  ‘owner’: ‘Owner12345’,
  ‘school’: null
  ‘self’: ‘https://kima4-cs493-final.ue.r.appspot.com/pets/34343’
}
```

Failure

```
Status: 400 Bad Request

{
  ‘Error’: ‘At least one of the attributes in the request object
has an invalid value’
}
```

```
Status: 400 Bad Request

{
  ‘Error’: ‘PATCH to /pets/:pet_id cannot be used to update the
school - use PUT or DELETE to /pets/:pet_id/school/:school_id to
modify relationships between pets and schools’
}
```

```
Status: 400 Bad Request

{
  ‘Error’: ‘The owner of the pet cannot be changed’
}
```

```
Status: 401 Unauthorized

{
  ‘Error’: ‘The JWT is missing or invalid’
}
```

```
Status: 403 Forbidden

{
  ‘Error’: ‘The pet belongs to someone else’
}
```

```
Status: 404 Not Found

{
  ‘Error’: ‘No pet with this pet_id exists’
}
```

```
Status: 406 Not Acceptable

{
  ‘Error’: ‘The requested response type must be application/json'
}
```

Put a Pet [Protected]

Puts an existing pet. Only name, breed, and age can be modified. Other properties are ignored or will return error codes.

```
PUT /pets/:pet_id
```

Request

Path Parameters

Name	Description
pet_id	ID of the pet.

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Name of the pet. Can only contain letters, spaces, periods, dashes, and apostrophes and must be between 1 and 50 characters, inclusive.	Yes
breed	Breed or species of the pet. Can only contain letters, spaces, periods, dashes, and apostrophes and must be between 1 and 30 characters, inclusive.	Yes
age	Age of the pet in years. Must be between 0 and 150, inclusive.	Yes

Request Body Example

```
{
  ‘name’: ‘John’,
  ‘breed’: ‘Chipmunk’,
  ‘age’: 3
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	Puts the URL of the updated pet into the Location header of the response.
Failure	400 Bad Request	One or more attributes in the request body are missing or have invalid values.
Failure	400 Bad Request	The request body tries to change the school property.
Failure	400 Bad Request	The request body tries to change the owner property.
Failure	401 Unauthorized	The JWT is either missing or invalid.
Failure	403 Forbidden	The pet belongs to a different user.
Failure	404 Not Found	A pet with the supplied ID was not found.
Failure	406 Not Acceptable	The server is unable to send the response in the requested type. Only application/json is acceptable.

Response Examples

Success

```
Status: 303 See Other

Location header contains URL of updated pet.
```

Failure

```
Status: 400 Bad Request

{
  'Error': 'The request object is missing at least one of the
required attributes or at least one of the attributes has an
invalid value'
}
```

```
Status: 400 Bad Request

{
  'Error': 'PUT to /pets/:pet_id cannot be used to update the
school - use PUT or DELETE to /pets/:pet_id/school/:school_id to
modify relationships between pets and schools'
}
```

```
Status: 400 Bad Request

{
  ‘Error’: ‘The owner of the pet cannot be changed’
}
```

```
Status: 401 Unauthorized

{
  ‘Error’: ‘The JWT is missing or invalid’
}
```

```
Status: 403 Forbidden

{
  ‘Error’: ‘The pet belongs to someone else’
}
```

```
Status: 404 Not Found

{
  ‘Error’: ‘No pet with this pet_id exists’
}
```

```
Status: 406 Not Acceptable

{
  ‘Error’: ‘The requested response type must be application/json'
}
```

Delete a Pet [Protected]

Deletes an existing pet. The pet id value is removed from the pets property of its owner. If a pet is currently enrolled in a school, the pet id is removed from the school's student property.

```
DELETE /pets/:pet_id
```

Request

Path Parameters

Name	Description
pet_id	ID of the pet.

Request Body

None

Response

Response Body Format

Success: None
Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	The JWT is either missing or invalid.
Failure	403 Forbidden	The pet belongs to a different user.
Failure	404 Not Found	A pet with the supplied ID was not found.
Failure	406 Not Acceptable	The server is unable to send the response in the requested type. Only application/json is acceptable.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 401 Unauthorized
```

```
{
  ‘Error’: ‘The JWT is missing or invalid’
}
```

```
Status: 403 Forbidden

{
  ‘Error’: ‘The pet belongs to someone else’
}
```

```
Status: 404 Not Found

{
  ‘Error’: ‘No pet with this pet_id exists’
}
```

```
Status: 406 Not Acceptable

{
  ‘Error’: ‘The requested response type must be application/json'
}
```

API Specifications - Schools

Attempting to PATCH, PUT, or DELETE to /schools will result in a 405 status code.

Create a School [Unprotected]

Creates a new school. Attributes other than name, location, and headmaster are ignored. The student value is set to an empty array by default.

```
POST /schools
```

Request

Path Parameters
None

Request Body
Required

Request Body Format
JSON

Request JSON Attributes

Name	Description	Required?
name	Name of the school. Can only contain letters, numbers, spaces, periods, dashes, commas, and apostrophes and must be between 1 and 50 characters, inclusive.	Yes
location	Location of the school. Can only contain letters, numbers, spaces, periods, dashes, commas, and apostrophes and must be between 1 and 80 characters, inclusive.	Yes
headmaster	Headmaster of the school. Can only contain letters, spaces, periods, dashes, and apostrophes and must be between 1 and 50 characters, inclusive.	Yes

Request Body Example

```
{
  ‘name’: ‘Canine Guitar Lessons’,
  ‘location’: ‘Chicago, Illinois’,
  ‘headmaster’: ‘Andrew Young’
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	Returns the JSON object that represents the school.
Failure	400 Bad Request	The request body is missing one or more required attributes or one or more given attributes have invalid values.
Failure	406 Not Acceptable	The server is unable to send the response in the requested type. Only application/json is acceptable.

Response Examples

Success

```
Status: 201 Created

{
  ‘id’: 22222,
  ‘name’: ‘Canine Guitar Lessons’,
  ‘location’: ‘Chicago, Illinois’,
  ‘headmaster’: ‘Andrew Young’,
  ‘students’: [],
  ‘self’:
    ‘https://kima4-cs493-final.ue.r.appspot.com/schools/22222’
}
```

Failure

```
Status: 400 Bad Request

{
  ‘Error’: ‘The request object is missing at least one of the
required attributes or at least one of the attributes has an
invalid value’
}
```

```
Status: 406 Not Acceptable

{
  ‘Error’: ‘The requested response type must be application/json’
}
```

Get All Schools [Unprotected]

Gets a list of schools. Response is paginated with a maximum of five schools per page.

```
GET /schools?page=:page
```

Request

Path Parameters

None

Query Parameters

Name	Description	Required?
page	Page number for pagination. If a page is not given, defaults to the first page, i.e. the first five results.	No

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns a JSON object with the property ‘schools’ which contains a list of the information of up to five schools. There is also a property ‘total_schools’ that displays the total number of schools in the system. If there are more schools that have not been retrieved, there will also be a ‘next’ key that contains the URL for the next set of schools to be retrieved.
Failure	406 Not Acceptable	The server is unable to send the response in the requested type. Only application/json is acceptable.

Response Examples

Success

```
Status: 200 OK

{
  ‘schools’: [
    {
      ‘id’: 11111,
      ‘name’: ‘Super School’,
      ‘location’: ‘Ottawa, Canada’,
      ‘headmaster’: ‘Samantha Kramer’,
      ‘students’: [21212],
      ‘self’:
        ‘https://kima4-cs493-final.ue.r.appspot.com/schools/11111’
    },
    {
      ‘id’: 22222,
      ‘name’: ‘Canine Guitar Lessons’,
      ‘location’: ‘Chicago, Illinois’,
      ‘headmaster’: ‘Andrew Young’,
      ‘students’: [],
      ‘self’:
        ‘https://kima4-cs493-final.ue.r.appspot.com/schools/22222’
    }
  ],
  ‘total_schools’: 2
}
```

Failure

```
Status: 406 Not Acceptable

{
  ‘Error’: ‘The requested response type must be application/json’
}
```

Get a School [Unprotected]

Gets an existing school.

```
GET /schools/:school_id
```

Request

Path Parameters

Name	Description
school_id	ID of the school.

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns a JSON object that represents the school.
Failure	404 Not Found	A school with the supplied ID was not found.
Failure	406 Not Acceptable	The server is unable to send the response in the requested type. Only application/json is acceptable.

Response Examples

Success

```
Status: 200 OK

{
  ‘id’: 11111,
  ‘name’: ‘Super School’,
  ‘location’: ‘Ottawa, Canada’,
  ‘headmaster’: ‘Samantha Kramer’,
  ‘students’: [21212],
  ‘self’:
    ‘https://kima4-cs493-final.ue.r.appspot.com/schools/11111’
}
```

Failure

```
Status: 404 Not Found

{
  ‘Error’: ‘No school with this school_id exists’
}
```

```
Status: 406 Not Acceptable

{
  ‘Error’: ‘The requested response type must be application/json’
}
```

Patch a School [Unprotected]

Patches an existing school. Only name, location, and headmaster can be modified. Other properties are ignored or will return error codes.

```
PATCH /schools/:school_id
```

Request

Path Parameters

Name	Description
school_id	ID of the school.

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Name of the school. Can only contain letters, numbers, spaces, periods, dashes, commas, and apostrophes and must be between 1 and 50 characters, inclusive.	No
location	Location of the school. Can only contain letters, numbers, spaces, periods, dashes, commas, and apostrophes and must be between 1 and 80 characters, inclusive.	No
headmaster	Headmaster of the school. Can only contain letters, spaces, periods, dashes, and apostrophes and must be between 1 and 50 characters, inclusive.	No

Request Body Example

```
{
  ‘headmaster’: ‘Valentina Surkov’
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns a JSON object that represents the school.
Failure	400 Bad Request	One or more attributes in the request body have invalid values.
Failure	400 Bad Request	The request body tries to change the students property.
Failure	404 Not Found	A school with the supplied ID was not found.
Failure	406 Not Acceptable	The server is unable to send the response in the requested type. Only application/json is acceptable.

Response Examples

Success

```
Status: 200 OK

{
  ‘id’: 22222,
  ‘name’: ‘Canine Guitar Lessons’,
  ‘location’: ‘Chicago, Illinois’,
  ‘headmaster’: ‘Valentina Surkov’,
  ‘students’: [],
  ‘self’:
    ‘https://kima4-cs493-final.ue.r.appspot.com/schools/22222’
}
```

Failure

```
Status: 400 Bad Request

{
  ‘Error’: ‘At least one of the attributes in the request object
has an invalid value’
}
```

```
Status: 400 Bad Request

{
  ‘Error’: ‘PATCH to /schools/:school_id cannot be used to update
the students - use PUT or DELETE to /pets/:pet_id/school/:school_id
to modify relationships between pets and schools’
}
```

```
Status: 404 Not Found

{
  ‘Error’: ‘No school with this school_id exists’
}
```

```
Status: 406 Not Acceptable

{
  ‘Error’: ‘The requested response type must be application/json'
}
```

Put a School [Unprotected]

Puts an existing school. Only name, location, and headmaster can be modified. Other properties are ignored or will return error codes.

```
PUT /schools/:school_id
```

Request

Path Parameters

Name	Description
school_id	ID of the school.

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Name of the school. Can only contain letters, numbers, spaces, periods, dashes, commas, and apostrophes and must be between 1 and 50 characters, inclusive.	Yes
location	Location of the school. Can only contain letters, numbers, spaces, periods, dashes, commas, and apostrophes and must be between 1 and 80 characters, inclusive.	Yes
headmaster	Headmaster of the school. Can only contain letters, spaces, periods, dashes, and apostrophes and must be between 1 and 50 characters, inclusive.	Yes

Request Body Example

```
{
  ‘name’: ‘Reptile Repository’,
  ‘location’: ‘Phoenix, Arizona’,
  ‘headmaster’: ‘Charles Smitty’
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	Puts the URL of the updated school into the Location header of the response.
Failure	400 Bad Request	One or more attributes in the request body are missing or have invalid values.
Failure	400 Bad Request	The request body tries to change the students property.
Failure	404 Not Found	A school with the supplied ID was not found.
Failure	406 Not Acceptable	The server is unable to send the response in the requested type. Only application/json is acceptable.

Response Examples

Success

```
Status: 303 See Other

Location header contains URL of updated school.
```

Failure

```
Status: 400 Bad Request

{
  ‘Error’: ‘The request object is missing at least one of the
required attributes or at least one of the attributes has an
invalid value’
}
```

```
Status: 400 Bad Request

{
  ‘Error’: ‘PUT to /schools/:school_id cannot be used to update the
students - use PUT or DELETE to /pets/:pet_id/school/:school_id to
modify relationships between pets and schools’
}
```

```
Status: 404 Not Found
```

```
{
  ‘Error’: ‘No school with this school_id exists’
}
```

```
Status: 406 Not Acceptable

{
  ‘Error’: ‘The requested response type must be application/json'
}
```

Delete a School [Unprotected]

Deletes an existing school. If a school has pets currently enrolled, the school property for those pets are set to null.

```
DELETE /schools/:school_id
```

Request

Path Parameters

Name	Description
school_id	ID of the school.

Request Body

None

Response

Response Body Format

Success: None
Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	A school with the supplied ID was not found.
Failure	406 Not Acceptable	The server is unable to send the response in the requested type. Only application/json is acceptable.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found

{
  'Error': 'No school with this school_id exists'
}
```

```
Status: 406 Not Acceptable

{
  ‘Error’: ‘The requested response type must be application/json'
}
```

API Specifications - Relationships

Enroll a Pet in a School [Protected]

Adds the pet to the school students property and adds the school to the pet school property. Can only be done if the pet is not already enrolled at a school.

```
PUT /pets/:pet_id/schools/:school_id
```

Request

Path Parameters

Name	Description
pet_id	ID of the pet.
school_id	ID of the school.

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	The JWT is either missing or invalid.
Failure	403 Forbidden	The pet belongs to a different user.
Failure	403 Forbidden	The pet is already enrolled at a school.
Failure	404 Not Found	The pet and/or school with the supplied IDs were not found.
Failure	406 Not Acceptable	The server is unable to send the response in the requested type. Only application/json is acceptable.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 401 Unauthorized

{
  ‘Error’: ‘The JWT is missing or invalid’
}
```

```
Status: 403 Forbidden

{
  ‘Error’: ‘The pet belongs to someone else’
}
```

```
Status: 403 Forbidden

{
  ‘Error’: ‘The pet is already enrolled at a school’
}
```

```
Status: 404 Not Found

{
  ‘Error’: ‘The specified pet and/or school does not exist’
}
```

```
Status: 406 Not Acceptable

{
  ‘Error’: ‘The requested response type must be application/json'
}
```

Unenroll a Pet from a School [Protected]

Removes the pet from the school students property and sets the pet school property to null.

```
DELETE /pets/:pet_id/schools/:school_id
```

Request

Path Parameters

Name	Description
pet_id	ID of the pet.
school_id	ID of the school.

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	The JWT is either missing or invalid.
Failure	403 Forbidden	The pet is already enrolled at a school.
Failure	404 Not Found	The pet and/or school with the supplied IDs were not found, or the pet is not enrolled at the specified school.
Failure	406 Not Acceptable	The server is unable to send the response in the requested type. Only application/json is acceptable.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 401 Unauthorized
```

```
{
  ‘Error’: ‘The JWT is missing or invalid’
}
```

```
Status: 403 Forbidden

{
  ‘Error’: ‘The pet belongs to someone else’
}
```

```
Status: 404 Not Found

{
  ‘Error’: ‘No pet with this pet_id is enrolled at a school with
this school_id’
}
```

```
Status: 406 Not Acceptable

{
  ‘Error’: ‘The requested response type must be application/json'
}
```