

Assignment 9: API Spec

CS 493: Cloud Application Development

Spring 2021

Oregon State University

Alyssa Lada

Deployed at: <https://hw9-ladaa.ue.r.appspot.com/>

Login: <https://hw9-ladaa.ue.r.appspot.com/>

[Change log](#)

[Data Model](#)

[List all Users](#)

[List all Dogs](#)

[List all Services](#)

[Create a Service](#)

[Get an Service](#)

[Edit a Service -- PATCH](#)

[Edit all Services -- PATCH](#)

[Edit a Service -- PUT](#)

[Edit all Services -- PUT](#)

[Delete all Services](#)

[Delete a Service](#)

[Create a Dog](#)

[Get a Dog](#)

[Edit a Dog -- PATCH](#)

[Edit all Dogs -- PATCH](#)

[Edit a Dog -- PUT](#)

[Edit all Dogs-- PUT](#)

[Delete all Dogs](#)

[Delete a Dog](#)

[Assign an Service to a Dog](#)

[Remove an Service from a Dog](#)

Change log

Version	Change	Date
1.1	Updated relationship routes	June 2 2021
1.0	Initial version.	May 21 2021

Data Model

User

Description

The user entity is automatically created by logging in to the login page. If the user already exists, each log in refreshes the datastore with the current user information. The user entity cannot be posted, patched, put, or deleted.

Unique Identifier

The sub value of the generated JWT is used as the id attribute of the datastore key for the user entity.

Making Requests with the User Identifier

Protected routes require a valid user JWT passed as a bearer token in the authentication header. The bearer token is parsed in the route to retrieve the sub value of the jwt, which is then used as the datastore key to retrieve and modify the datastore user entity. No routes in this API require the direct use of the user ID value.

User / Non-User Relationships

The user entity has a zero to many relationship with the dog entity.

Properties

Property Name	Type	Required	Valid Values	Notes
id	String	Yes	Automatically generated by logging in to the application home page.	Sub value of the JWT generated at log in.
nickname	String	Yes	Automatically generated by logging in to the application home page.	The username portion of the email address used to log in to the app.
email	String	Yes	Automatically generated by logging in to the application home page.	The email address used to log in to the app.
dogs	Array	No	Automatically generated in POST/dogs route.	Contains dog ids. If there are no dogs assigned to the user, then dogs is undefined.

Dog

Description

A dog. The dog entity stores the name, breed, weight, owner, and assigned services for dog ID. For example, a 45.3 pound Collie named Lassie owned by Steve.

Relationships

The dog entity has a zero to many relationship with the service entity. The dog entity must be related to one and only one user entity.

Properties

Property Name	Type	Required	Valid Values	Notes
id	String	Yes	Automatically generated by datastore.	The id of the dog.
name	String	Yes	Any non-empty string.	Name of the dog.
breed	String	Yes	Any non-empty string.	The dog's breed, i.e. Golden Retriever, Collie, etc.
weight	Number (Integer or Float)	Yes	A number greater than zero.	The weight of the dog in pounds.
owner	String	No	User id. Automatically generated in POST/dogs route.	The id of the user that the dog is assigned to.
services	Array	No	Automatically generated in PUT/dogs/dogId/services/serviceId/ route.	Contains the ids of the services assigned to the dog. If the dog has no assigned services, then services is undefined.

Service

Description

A service for a dog's care. The service entity stores the type, frequency, technician, and dog IDs for dogs assigned the service. For example, a daily walk with the technician Laura might be assigned to the dog IDs for Lassie and Laddie.

Relationships

The service entity has a zero to many relationship with the dog entity.

Properties

Property Name	Type	Required	Valid Values	Notes
id	String	Yes	Automatically generated by datastore.	The id of the service.
type	String	Yes	Any non-empty string.	The type of service.
frequency	String	Yes	Any non-empty string.	The frequency at which the service occurs.
technician	String	Yes	Any non-empty string.	The name of the assigned technician.
dogs	Array	No	Automatically generated in PUT/dogs/dogId/services/serviceId/ route.	Contains the ids of the dogs receiving the service. If the service is not assigned to at least one dog, then dogs is undefined.

List all Users

List all the users currently registered in the app. Unprotected endpoint.

GET /users

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	Returned if the content-type header in the request is not set to application/json.

Response Examples

Success

```
Status: 200 OK
{
  "count":2,
  "items":[
    {
      "email":"dog@place.com",
      "nickname":"dog",
      "id":"auth0|60ba8f6e1420880sfhzlhfuwaf4"
    },
    {
      "email":"rachel@petcare.com",
      "id":"auth0|60bbe9sdfbxdvb9479675",
      "nickname":"rachel"
      "dogs": [
        "arflasvn214",
        "afalifdabml"
      ]
    }
  ]
}
```

Failure

Status: 406 Not Acceptable

```
{  
  "Error": "Request header must accept application/json"  
}
```

List all Dogs

List all the dogs registered to the currently logged in user, 5 dogs per page, with a link to the next page of results unless there are no more pages of results. Protected endpoint.

GET /dogs

Request

Authorization Header

Bearer Token

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Must send with valid JWT as Bearer Token
Failure	406 Not Acceptable	Returned if the content-type header in the request is not set to application/json.

Response Examples

Success

```
Status: 200 OK
{
  "count": 6,
  "items": [
    {
      "id": "1a234z56",
      "name": "Fido",
      "breed": "German Shepherd",
      "weight": 54.6,
      "owner": "98756dagr5",
      "services": [
        "Jfhgr",
        "753289yth"
      ],
      "self": "https://<your-app>/dogs/1a234z56"
    },
    {
      "id": "bhjbhj45",
      "name": "Lassie",
```



```

    "breed": "Rough Coated Collie",
    "weight": 65,
    "owner": "98756dagr5",
    "self": "https://<your-app>/dogs/bhjbhj45"
  },
  {
    "id": "4grre7",
    "name": "Gertrude",
    "breed": "German Shepherd",
    "weight": 75,
    "owner": "98756dagr5",
    "self": "https://<your-app>/dogs/4grre7"
  },
  {
    "id": "bhjbhj45",
    "name": "Lassie",
    "breed": "Rough Coated Collie",
    "weight": 65,
    "owner": "98756dagr5",
    "self": "https://<your-app>/dogs/bhjbhj45"
  },
  {
    "id": "1a234z56",
    "name": "Snoopy",
    "breed": "Beagle",
    "weight": 27,
    "owner": "98756dagr5",
    "self": "https://<your-app>/dogs/1a234z56"
  }
],
"next": "https://<your-app>/dogs?cursor=<cursor-key>"
}

```

Failure

Status: 401 Unauthorized
<pre> { "Error": "Unauthorized -- <error message>" } </pre>
Status: 406 Not Acceptable
<pre> { "Error": "Request header must accept application/json" } </pre>

List all Services

List all the services, 5 services per page, with a link to the next page of results unless there are no more pages of results. Unprotected endpoint.

GET /services

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	Returned if the content-type header in the request is not set to application/json.

Response Examples

Success

```
Status: 200 OK
{
  "count": 8,
  "items": [
    {
      "id": "jfhgr",
      "technician": "Steve Kowsinski",
      "type": "30 minute walk",
      "frequency": "Daily",
      "dog": "bhjbhj45",
      "self": "https://<your-app>/services/jfhgr"
    },
    {
      "id": "753289yth",
      "technician": "Ciri Kowsinski",
      "type": "Groom",
      "frequency": "Monthly",
      "dog": "sdfdsttt4",
      "self": "https://<your-app>/services/753289yth"
    },
    {
      "id": "32132d",
      "technician": "Steve Kowsinski",
```

```
"type": "Nail Trim",
"frequency": "Monthly",
"dog": "kjhj",
"self": "https://<your-app>/services/32132d"
},
{
  "id": "09876v",
  "technician": "Ciri Kowsinski",
  "type": "15 minute walk",
  "frequency": "Daily",
  "dog": "bhjbhj45",
  "self": "https://<your-app>/services/09876v"
},
{
  "id": "g87fyjhf",
  "technician": "Steve Kowsinski",
  "type": "Brush",
  "frequency": "Daily",
  "dog": "bhjbhj45",
  "self": "https://<your-app>/services/g87fyjhf"
}
],
"next": "https://<your-app>/services?cursor=<cursor-key>"
}
```

Failure

Status: 406 Not Acceptable

```
{
  "Error": "Request header must accept application/json"
}
```

Create a Service

Allows you to create a new service. Unprotected route.

POST /services

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Strings cannot be empty.

Name	Type	Description	Required?
type	String	The type of service.	Yes
frequency	String	The frequency of the service.	Yes
technician	String	The name of the technician assigned to service.	Yes

Request Body Example

```
{
  "type": "groom",
  "frequency": "monthly",
  "technician": "Rachel"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	Returned if the request is missing any of the 3 required attributes, or if any of the passed attributes are invalid.
Failure	406 Not Acceptable	Returned if the content-type header in the request is not set to application/json.
Failure	406 Not Acceptable	Returned if the request body is not application/json

Response Examples

Success

Status: 201 Created

```
{
  "id": "aghbc123",
  "type": "groom",
  "frequency": "monthly",
  "technician": "Rachel"
  "self": "https://<your-app>/services/aghbc123"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "Invalid request parameters"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Request header must accept application/json"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Server only accepts application/json data"
}
```

Get a Service

Allows you to get an existing service. Unprotected endpoint.

GET /services/:service_id

Request

Path Parameters

Name	Type	Description
service_id	String	ID of the service

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No service with this id exists
Failure	406 Not Acceptable	Returned if the content-type header in the request is not set to application/json.

Response Examples

Success

Status: 200 OK
<pre>{ "id": "abc123", "type": "groom", "frequency": "monthly", "technician": "Rachel" "self": "https://<your-app>/services/abc123" }</pre>

Failure

Status: 404 Not Found
<pre>{ "Error": "No service with this id exists" }</pre>
Status: 406 Not Acceptable
<pre>{ "Error": "Request header must accept application/json" }</pre>

Edit a Service -- PATCH

Allows you to edit a service. Unprotected endpoint.

PATCH /services/:service_id

Request

Path Parameters

Name	Type	Description
service_id	String	ID of the service

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Only the attributes passed in the request are changed. Original attributes that are not updated remain unchanged. Strings cannot be empty.

Name	Type	Description	Required?
type	String	The type of the service.	No
frequency	String	The frequency at which the service occurs.	No
technician	String	The name of the assigned technician.	No

Request Body Example

```
{
  "type": "groom",
  "frequency": "monthly",
  "technician": "Rachel"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	The request parameters included id or dogs, or the passed values were invalid.
Failure	404 Not Found	No service with this id exists.
Failure	406 Not Acceptable	Returned if the content-type header in the request is not set to application/json.
Failure	406 Not Acceptable	Returned if the request body is not application/json

Response Examples

Success

Status: 200 OK

```
{
  "id": "abc123",
  "type": "groom",
  "frequency": "monthly",
  "technician": "Rachel"
  "self": "https://<your-app>/services/abc123"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "Invalid request parameters"
}
```

Status: 404 Not Found

```
{
  "Error": "No service with this id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Request header must accept application/json"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Server only accepts application/json data"
}
```


Edit all Services -- PATCH

Not allowed. Unprotected endpoint.

PATCH /services/

Request

None

Response

No body

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Failure	405 Method Not Allowed	Response header "Allow" is "GET, POST"

Response Examples

Failure

Status: 405 Method Not Allowed

Edit a Service -- PUT

Allows you to reset all attributes for a service. Unprotected endpoint.

PUT /services/:service_id

Request

Path Parameters

Name	Type	Description
service_id	String	ID of the service

Request Body

Request Body Format

JSON

Request JSON Attributes

Any attributes not passed in the request are set to “default”. Attributes may not be empty strings.

Name	Type	Description	Required?
type	String	The type of the service.	No
frequency	String	The frequency at which the service occurs.	No
technician	String	The name of the assigned technician.	No

Request Body Example

```
{
  "type": "groom",
  "frequency": "monthly",
  "technician": "Rachel"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	Cannot update id or dogs, or passed attributes were invalid.
Failure	404 Not Found	No service with this id exists
Failure	406 Not Acceptable	Returned if the content-type header in the request is not set to application/json.
Failure	406 Not Acceptable	Returned if the request body is not application/json

Response Examples

Success

Status: 200 OK

```
{
  "id": "abc123",
  "type": "groom",
  "frequency": "monthly",
  "technician": "Rachel"
  "self": "https://<your-app>/services/abc123"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "Invalid request parameters"
}
```

Status: 404 Not Found

```
{
  "Error": "No service with this id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Request header must accept application/json"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Server only accepts application/json data"
}
```

Edit all Services -- PUT

Not allowed. Unprotected endpoint.

PUT /services/

Request

None

Response

No body

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Failure	405 Method Not Allowed	Response header "Allow" is "GET, POST"

Response Examples

Failure

Status: 405 Method Not Allowed

Delete all Services

Not allowed. Unprotected endpoint.

DELETE /services/

Request

None

Response

No body

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Failure	405 Method Not Allowed	Response header "Allow" is "GET, POST"

Response Examples

Failure

Status: 405 Method Not Allowed

Delete a Service

Allows you to delete a service. If the service is assigned to a dog, deleting the service removes the service from the dog. Unprotected endpoint.

DELETE /services/:service_id

Request

Path Parameters

Name	Type	Description
service_id	String	ID of the service

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No service with this id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

```
{  
  "Error": "No service with this id exists"  
}
```

Create a Dog

Adds a new dog to the database. Protected endpoint.

POST /dogs

Request

Authorization Header

Bearer Token

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Strings cannot be empty, and weight must be greater than zero.

Name	Type	Description	Required?
name	String	The name of the dog.	Yes
breed	String	The breed of the dog.	Yes
weight	Number	The weight of the dog.	Yes

Body Example

```
{
  "name": "Lassie",
  "breed": "Rough Coated Collie",
  "weight": 45
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	URL of created resource included in Location header
Failure	400 Bad Request	Returned if the request is missing any of the 3 required attributes, or if the attributes are invalid.
Failure	401 Unauthorized	Must send with valid JWT as Bearer Token
Failure	406 Not Acceptable	Returned if the content-type header in the request is not set to application/json.
Failure	406 Not Acceptable	Returned if the request body is not application/json

Response Examples

Success

Status: 201 Created

```
{
  "id": "675fjc",
  "owner": "798tgh9",
  "name": "Lassie",
  "breed": "Rough Coated Collie",
  "weight": 45
  "self": "https://<your-app>/dogs/675fjc"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "Invalid request parameters"
}
```

Status: 401 Unauthorized

```
{
  "Error": "Unauthorized -- <error message>"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Request header must accept application/json"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Server only accepts application/json data"
}
```


Get a Dog

Allows you to get an existing dog. Protected endpoint.

GET /dogs/:dog_id

Request

Authorization Header

Bearer Token

Path Parameters

Name	Type	Description
dog_id	String	ID of the dog

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Must send with valid JWT as Bearer Token
Failure	404 Not Found	No dog with this id exists for the logged in user
Failure	406 Not Acceptable	Returned if the content-type header in the request is not set to application/json.

Response Examples

Success

Status: 200 OK
<pre>{ "id": "675fjc", "owner": "798tgh9", "name": "Lassie", "breed": "Rough Coated Collie", "weight": 45, "self": "https://<your-app>/dogs/675fjc" }</pre>

Failure

Status: 401 Unauthorized
<pre>{ "Error": "Unauthorized -- <error message>" }</pre>
Status: 404 Not Found

<pre>{ "Error": "No dog with this id exists for the logged in user" }</pre>
<p>Status: 406 Not Acceptable</p> <pre>{ "Error": "Request header must accept application/json" }</pre>

Edit a Dog -- PATCH

Allows you to edit a dog. Protected endpoint.

PATCH /dogs/:dog_id

Request

Authorization Header

Bearer Token

Path Parameters

Name	Type	Description
dog_id	String	ID of the dog

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Only the attributes passed in the request are changed. Attributes may not be empty strings, and weight must be a number greater than zero.

Name	Type	Description	Required?
name	String	The name of the dog.	No
breed	String	The breed of the dog.	No
weight	Number	The weight of the dog.	No

Request Body Example

```
{
  "name": "Laddie",
  "breed": "Smooth Coated Collie",
  "weight": 47
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	The request parameters included id or services parameters, or included invalid values for breed, name, or weight.
Failure	401 Unauthorized	Must send with valid JWT as Bearer Token
Failure	404 Not Found	No dog with this id exists for the logged in user
Failure	406 Not Acceptable	Returned if the content-type header in the request is not set to application/json.

Failure	406 Not Acceptable	Returned if the request body is not application/json
---------	--------------------	--

Response Examples

Success

<p>Status: 200 OK</p> <pre>{ "id": "675fjc", "owner": "798tgh9", "name": "Laddie", "breed": "Smooth Coated Collie", "weight": 47, "self": "https://<your-app>/dogs/675fjc" }</pre>
--

Failure

<p>Status: 400 Bad Request</p> <pre>{ "Error": "Invalid request parameters" }</pre>
<p>Status: 401 Unauthorized</p> <pre>{ "Error": "Unauthorized -- <error message>" }</pre>
<p>Status: 404 Not Found</p> <pre>{ "Error": "No dog with this id exists for the logged in user" }</pre>
<p>Status: 406 Not Acceptable</p> <pre>{ "Error": "Request header must accept application/json" }</pre>
<p>Status: 406 Not Acceptable</p> <pre>{ "Error": "Server only accepts application/json data" }</pre>

Edit all Dogs -- PATCH

Not allowed. Protected endpoint.

PATCH /dogs/

Request

Authorization Header

Bearer Token

Response

No body

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Failure	401 Unauthorized	Must send with valid JWT as Bearer Token
Failure	405 Method Not Allowed	Response header "Allow" is "GET, POST"

Response Examples

Failure

Status: 401 Unauthorized
<pre>{ "Error": "Unauthorized -- <error message>" }</pre>
Status: 405 Method Not Allowed

Edit a Dog -- PUT

Allows you to edit an entire dog and set all attributes. Protected endpoint.

PUT /dogs/:dog_id

Request

Authorization Header

Bearer Token

Path Parameters

Name	Type	Description
dog_id	String	ID of the dog

Request Body

Passing an empty body request to this URL will set all attributes of the dog to null except id.

Request Body Format

JSON

Request JSON Attributes

Any attributes not passed in the request are set to "default". If weight is not passed, it is reset to the default value of 1. Attributes may not be empty strings, and weight must be a number greater than zero.

Name	Type	Description	Required?
name	String	The name of the dog.	No
breed	String	The breed of the dog.	No
weight	Integer	The weight of the dog.	No

Request Body Example

```
{
  "name": "Lassie",
  "breed": "Rough Coated Collie",
  "weight": 45
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 Ok	
Failure	400 Bad Request	The request parameters included id or services parameters, or included invalid values for breed, name, or weight.
Failure	401 Unauthorized	Must send with valid JWT as Bearer Token
Failure	404 Not Found	No dog with this id exists for the logged in user

Failure	406 Not Acceptable	Returned if the content-type header in the request is not set to application/json.
Failure	406 Not Acceptable	Returned if the request body is not application/json

Response Examples

Success

<p>Status: 200 Ok</p> <pre>{ "id": "675fjc", "owner": "798tgh9", "name": "Lassie", "breed": "Rough Coated Collie", "weight": 45 "self": "https://<your-app>/dogs/675fjc" }</pre>
--

Failure

<p>Status: 400 Bad Request</p> <pre>{ "Error": "Invalid request parameters" }</pre>
<p>Status: 401 Unauthorized</p> <pre>{ "Error": "Unauthorized -- <error message>" }</pre>
<p>Status: 404 Not Found</p> <pre>{ "Error": "No dog with this id exists for the logged in user" }</pre>
<p>Status: 406 Not Acceptable</p> <pre>{ "Error": "Request header must accept application/json" }</pre>
<p>Status: 406 Not Acceptable</p> <pre>{ "Error": "Server only accepts application/json data" }</pre>

Edit all Dogs-- PUT

Not allowed. Protected endpoint.

PUT /dogs/

Request

Authorization Header

Bearer Token

Response

No body

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Failure	401 Unauthorized	Must send with valid JWT as Bearer Token
Failure	405 Method Not Allowed	Response header "Allow" is "GET, POST"

Response Examples

Failure

Status: 401 Unauthorized
<pre>{ "Error": "Unauthorized -- <error message>" }</pre>
Status: 405 Method Not Allowed

Delete all Dogs

Not allowed. Protected endpoint.

DELETE /dogs/

Request

Authorization Header

Bearer Token

Response

No body

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Failure	401 Unauthorized	Must send with valid JWT as Bearer Token
Failure	405 Method Not Allowed	Response header "Allow" is "GET, POST"

Response Examples

Failure

Status: 401 Unauthorized
<pre>{ "Error": "Unauthorized -- <error message>" }</pre>
Status: 405 Method Not Allowed

Delete a Dog

Allows you to delete a dog. Protected endpoint. If there are services assigned to the dog, the services are unassigned. Protected endpoint.

DELETE /dogs/:dog_id/

Request

Authorization Header

Bearer Token

Path Parameters

Name	Type	Description
dog_id	String	ID of the dog

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	Must send with valid JWT as Bearer Token
Failure	404 Not Found	No dog with this id exists for the logged in user

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized
<pre>{ "Error": "Unauthorized -- <error message>" }</pre>
Status: 404 Not Found
<pre>{ "Error": "No dog with this id exists for the logged in user" }</pre>

Assign an Service to a Dog

Add a service to a dog. Protected endpoint.

PUT /dogs/:dog_id/services/:service_id

Request

Authorization Header

Bearer Token

Path Parameters

Name	Type	Description
dog_id	String	ID of the dog
service_id	String	ID of the service

Request Body

None

Note: Set Content-Length to 0 in your request when calling out to this endpoint.

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a dog exists with this dog_id, and a service exists with this service_id, and the service is not already assigned to the dog
Failure	400 Bad Request	The service is already assigned to the dog.
Failure	401 Unauthorized	Must send with valid JWT as Bearer Token
Failure	403 Forbidden	The dog does not belong to this owner
Failure	404 Not Found	No dog with this dog_id exists, and/or no service with this service_id exists.

Response Examples

Success

Status: 204 No Content

Failure

Status: 400 Bad Request

```
{
  "Error": "This service is already assigned to this dog"
}
```

Status: 401 Unauthorized

```
{  
  "Error": "Unauthorized -- <error message>"  
}
```

Status: 403 Forbidden

```
{  
  "Error": "The specified dog does not belong to the logged in user"  
}
```

Status: 404 Not Found

```
{  
  "Error": "The specified dog and/or service does not exist"  
}
```

Remove a Service from a Dog

Remove a service from a dog. Protected endpoint.

DELETE /dogs/:dog_id/services/:service_id

Request

Authorization Header

Bearer Token

Path Parameters

Name	Type	Description
dog_id	String	ID of the dog
service_id	String	ID of the service

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a service exists with this service_id, a dog exists with this dog_id and this service is assigned to this dog.
Failure	401 Unauthorized	Must send with valid JWT as Bearer Token
Failure	403 Forbidden	The dog does not belong to this owner
Failure	404 Not Found	No service with this service_id is on the dog with this dog_id.

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized
<pre>{ "Error": "Unauthorized -- <error message>" }</pre>
Status: 403 Forbidden
<pre>{ "Error": "The specified dog does not belong to the logged in user" }</pre>

Status: 404 Not Found

```
{  
  "Error": "No service with this service_id is assigned to the dog with this dog_id"  
}
```