The APIs available for this application are designed in a RESTful way. To interact with the API, you must provide the requested information with the correct HTTP method.

The directory that houses the APIs is https://simplecoop.swollenhippo.com/

Endpoints:

coop.php

This endpoint (POST) should be called at the beginning of your project to get a CoopID that is activated and registered. This is to simulate a customer having a CoopID printed on a label on the side of their coop. You will use this newly created CoopID for all calls to create a new user.

Method: POST
Required Parameters:
Expected Returns:JSON Object with a key of CoopID

Method: PUT
Required Parameters: SessionID, Street1, Street2, City, State, ZIP
Expected Returns: JSON Object with a key of Outcome

Method: DELETE
Required Parameters: SessionID
Expected Returns: JSON Object with a key of Outcome

Method: GET
Required Parameters: SessionID
Expected Returns: An array of JSON objects with all Coop details

eggs.php

This endpoint can be used to keep a running count of the harvested eggs. For example, you could have a button on your UI allowing you to enter the eggs harvested for the day. You could then show an egg counter to the user or even graph the eggs using a library such as chartsjs

Method: POST
Required Parameters: SessionID, observationDateTime, eggs
Expected Returns: A JSON Object with a key of Outcome or LogID

Method: PUT
Required Parameters: SessionID, logID
Expected Returns:

Method: DELETE
Required Parameters: SessionID
Expected Returns: A JSON Object with a key of Outcome

Method: GET

Required Parameters: SessionID, days
Expected Returns: An array of JSON objects with all Egg record details

environment.php
This endpoint can be used to create, delete, and view temperature and humidity observations. This is useful if you wish to create a graph that shows the observations or if you want to add an element to your UI that displays the last observed temperature and humidity

Method: POST
Required Parameters: SessionID, observationDateTime, temperature, humidity
Expected Returns: A JSON Object with a key of Outcome

Method: DELETE
Required Parameters: SessionID, logID
Expected Returns: A JSON Object with a key of Outcome

Method: GET
Required Parameters: SessionID, days
Expected Returns: An array of JSON objects with all environment record details

sessions.php
This endpoint is used to create a session that can then be used to make requests to all other endpoints that require a sessionID. The put will allow you to update the last date/time the session is used. The get call would allow you to validate that a particular sessionID is still valid.

Method: POST
Required Parameters: Email, Password
Expected Returns: A JSON Object with a key of Outcome or SessionID

Method: PUT
Required Parameters: SessionID
Expected Returns: A JSON Object with a key of Outcome

Method: DELETE
Required Parameters: SessionID
Expected Returns: A JSON Object with a key of Outcome

Method: GET
Required Parameters: SessionID
Expected Returns: A JSON Object containing all Session Details

settings.php

This endpoint should be used to create records of various settings. The setting and value are stored in the database as a varchar allowing you the opportunity to choose what those should be stored as.

Method: POST
Required Parameters: SessionID, setting, value
Expected Returns: A JSON Object with a key of Outcome

Method: PUT
Required Parameters: SessionID, setting, value
Expected Returns: A JSON Object with a key of Outcome

Method: DELETE
Required Parameters: SessionID, setting
Expected Returns: A JSON Object with a key of Outcome

Method: GET
Required Parameters: SessionID, setting
Expected Returns: An array of JSON objects with all setting record details

useraddress.php

You may use this endpoint to record a user's address. This address could be used with the openweatherapi to show the current conditions at the location of the coop OR even provide a graph of the past weather conditions compared to the environment observation records

Method: POST
Required Parameters: Email, Street1, Street2, City, State, ZIP
Expected Returns: A JSON Object with a key of Outcome

Method: PUT
Required Parameters: Email, Street1, Street2, City, State, ZIP
Expected Returns: A JSON Object with a key of Outcome

Method: DELETE
Required Parameters: Email
Expected Returns: A JSON Object with a key of Outcome

Method: GET
Required Parameters: Email
Expected Returns: An array of JSON objects with all user address record details

users.php
This endpoint should be used to create, update, delete, or verify a user account.

Method: POST
Required Parameters: Email, Password, FirstName, LastName, CoopID
Expected Returns: A JSON Object with a key of Outcome

Method: PUT
Required Parameters: Email, Password, FirstName, LastName
Expected Returns: A JSON Object with a key of Outcome

Method: DELETE
Required Parameters: Email
Expected Returns: A JSON Object with a key of Outcome

Method: GET
Required Parameters: Email
Expected Returns: A JSON Object with all user details

Example Calls

coop.php
An example on how you can create a new coopID to use in your project using the jquery .post

```
$.post('https://simplecoop.swollenhippo.com/coop.php',function(result){
console.log(result);
})
```

users.php
An example on how you can create a new user using the jquery .post

```
$.post(https://simplecoop.swollenhippo.com/users.php'{Email:'bburchfield2@
tntech.edu',Password:'Mickey2022!',FirstName:'Bennn',LastName:'BURCHFIELD'
,CoopID:'65'},function(result){,function(result){
console.log(result);
})
```

sessions.php
An example on how you can create a new user using the jquery .post

```
$.post(https://simplecoop.swollenhippo.com/sessions.php'{Email:'bburchfiel
d2@tntech.edu',Password:'Mickey2022!'},function(result){,function(result){
console.log(result);
})
```

If you plan to use the environment endpoint in your project, below is a code snippet you can run to generate records associated with your coop.

```
function getRandomObservationDatetime() {
  // Get the current date
   var currentDate = new Date();
  // Generate a random number of days between 0 and 100
   var randomDays = Math.floor(Math.random() * 101);
  // Calculate the random date within the specified range
   var randomDate = new Date(currentDate);
   randomDate.setDate(currentDate.getDate() - randomDays);
  // Format the date in ISO string format
   var observationDatetime = randomDate.toISOString();
   return observationDatetime;
}
function getRandomTemperature() {
  // Generate a random temperature between 42 and 90
   return Math.random() * (90 - 42) + 42;
}
function getRandomHumidity() {
  // Generate a random humidity between 30 and 98
   return Math.random() * (98 - 30) + 30;
}
// Loop to execute the code block 100 times
for (var i = 0; i < 100; i++) {
  // The SessionID and additional data with random temperature, humidity,
and observationdatetime
   var requestData = {
     SessionID: 'PUTYOURSESSIONIDHERE',
     temperature: getRandomTemperature(),
     observationDateTime: getRandomObservationDatetime(),
     humidity: getRandomHumidity()
   };
  // Making the AJAX request using POST
   $.post('https://simplecoop.swollenhippo.com/environment.php',
requestData,  function(result) {
      console.log(result);
   });
}
```