

Nature Reviews

Created by Tim Dirks
April 26 2021

Overview:

This project is a website for people to sign up and post reviews on nature parks. Only users that have signed up may post comments on webpages, however guests may view the park pages and previous comments. There are 2 park pages, 1 login page, 1 sign up page, 1 home page and 1 admin page. All of the pages except the admin page can be accessed through the home page.

Homepage: <https://synopede.pw/nature-reviews/home.html>

GitHub: <https://github.com/timdir/nature-reviews>

All of the examples provided in the last section can be found at this GitHub link. Simply import this JSON file into Postman:

nature-reviews_API.postman_collection.json

API Documentation:

The API was developed using Swagger and exported as an html2 file. The documentation can be found at the following address:

<https://synopede.pw/nature-reviews/API/v1/documentation.html>

This API contains many more details on the HTTP status codes received from the server upon requests and also provides more examples of what to send as parameters in the body or in the path. Nevertheless, some extra examples are included in the last section.

Summary of API:

host:	synopede.pw	
basePath:	/nature-reviews/API/v1	
POST	/users/signup	Create a new user
POST	/users/login	Login to the site
POST	/users/admin	Admin login for admin page
GET	/parks/{park_id}	Get park by ID
POST	/comments	Add a new comment
PUT	/comments	Edit a comment
DELETE	/comments	Delete a comment
GET	/comments/{park_id}	Retrieve all comments on a park page by park ID
POST	/sessions	Verify the identity of the user
DELETE	/sessions	Delete a session key

Admin page:

The admin requires the following credentials in JSON format in the body of a POST request:

```
{  
  "username": "admin",  
  "password": "ty^D{jQXT7I93d6gbkym,ki4g.7u?>[r5|G0!!Qv6"  
}
```

Sometimes it is needed to press the Login button twice upon entering the credentials on the admin.html webpage. The admin cannot access any pages other than the Admin page. Requests can be sent in the body as a JSON to this link:

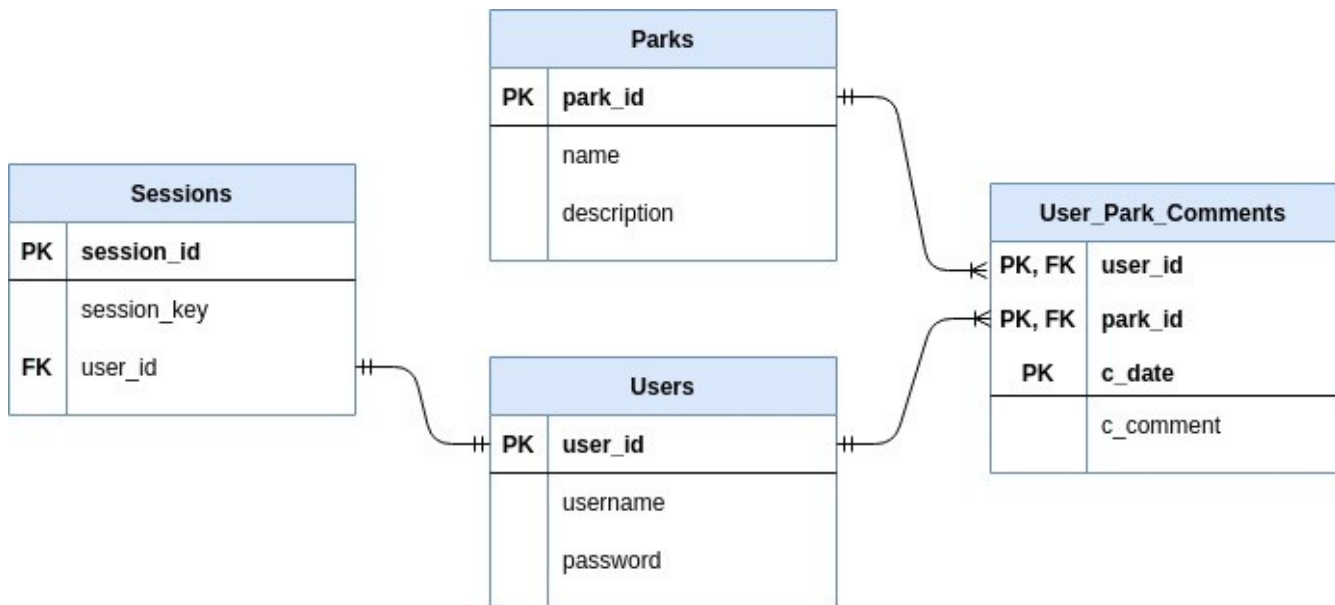
<https://synopede.pw/nature-reviews/API/v1/users/admin>

Or entered in the text fields at this html page:

<https://synopede.pw/nature-reviews/admin.html>

ERD:

This ERD shows the relationships between the relations and their attributes. Instead of API tokens, I chose to use a session key which can be renewed upon login and the previous token is destroyed. I did this for some slightly better security.



URLs and Parameters for Testing API Endpoints:

Get park data based on park id in the URL

Method: GET

URL: <https://synopede.pw/nature-reviews/API/v1/parks/1>

Requires token: No

Response from the server:

```
{
  "park_id": 1,
  "name": "Garibaldi Park",
  "description": "A beautiful park park with lots of hiking spots in British Columbia, Canada."
}
```

Get all comments on park with park id specified in the URL

Method: GET

URL: <https://synopede.pw/nature-reviews/API/v1/comments/1>

Requires token: No

Response from the server:

```
[
  {
    "username": "john",
    "c_date": "2018-02-14 19:39:00 UTC",
    "c_comment": "too cold"
  },
  {
    "username": "Chris.cross",
    "c_date": "2019-01-23 23:52:58 UTC",
    "c_comment": "this park is great!"
  },
  {
    "username": "caroline",
    "c_date": "2020-03-13 02:16:12 UTC",
    "c_comment": "did this with my best friend it was a lot of fun"
  }
]
```

Login to the site

Method: POST

URL: <https://synopede.pw/nature-reviews/API/v1/users/login>

Requires token: No

The site contains a default user that we can use for these examples. This data should be sent in the body in JSON format.

```
{
  "username": "default",
  "password": "1234"
}
```

The user will receive a session key as a response that will be stored in localStorage. If you are trying this example in Postman or another app, or on the website, you will receive a different session key and this one below will be destroyed.

Response from the server:

```
{"session_key": "osGxRhi4pIDf7wvEES2iDA=="}
```

Add a comment

Method: POST
URL: <https://synopede.pw/nature-reviews/API/v1/comments>
Requires token: Yes

This data should be sent in the body in JSON format. I am using the session key from the previous example, using session keys from these examples may not work for you. If you do a POST on /users/login first and receive a new session key, then you can replace the session key below and modify some of the data.

```
{
  "session_key":"osGxRhi4pIDf7wvEES2iDA==",
  "username":"default",
  "park_id":"1",
  "c_date":"2021-03-13 17:28:02 UTC",
  "c_comment":"I very much enjoyed the scenery at Garibaldi"
}
```

Response from the server:

Comment added.

Edit a comment

Method: PUT
URL: <https://synopede.pw/nature-reviews/API/v1/comments>
Requires token: Yes

This data should be sent in the body in JSON format. I am using the session key from the previous example, using session keys from these examples may not work for you.

```
{
  "session_key":"osGxRhi4pIDf7wvEES2iDA==",
  "username":"default",
  "park_id":"1",
  "c_date":"2021-03-13 18:49:53 UTC",
  "c_comment":"Also..... make sure to bring lots of water. We ran out!"
}
```

Response from the server:

Comment updated successfully!

Delete a comment

Method: Delete
URL: <https://synopede.pw/nature-reviews/API/v1/comments>
Requires token: Yes

This data should be sent in the body in JSON format. I am using the session key from the previous example, using session keys from these examples may not work for you.

```
{  
  "session_key":"osGxRhi4pIDf7wvEES2iDA==",  
  "username":"default",  
  "park_id":"1",  
  "c_date":"2021-03-13 18:49:53 UTC"  
}
```

Response from the server:

No response text is returned from the server, but a “204 No Content” HTTP code will be sent.

Sign up to the site

Method: POST
URL: <https://synopede.pw/nature-reviews/API/v1/users/signup>
Requires token: No

Here is an example of signing up to the site. This data should be sent in the body in JSON format.

```
{  
  "username":"clair",  
  "password":"4aj2&wPN,xkla"  
}
```

The user will receive a session key as a response that will be stored in localStorage. However, if you are trying this example in Postman, then it will not be stored in localStorage.

Response from the server:

```
{"session_key":"VlzaglyCCGcMVqtQgkp7iw=="}
```

Verify token is valid

Method: POST
URL: <https://synopede.pw/nature-reviews/API/v1/sessions>
Requires token: Yes

This token does not verify that it is the user, who owns the token, is currently using it. This request only verifies whether the token is valid. This data should be sent in the body in JSON format. I am using the session key from the previous example, using session keys from these examples may not work for you.

```
{  
  "session_key": "osGxRhi4pIDf7wvEES2iDA=="  
}
```

The server returns the user's username. This is used to change the greeting from Guest to the username and do other tasks such as enable the comment box, remove the Login and Sign up buttons and display the Edit and Delete buttons for comments the user has made.

Response from the server:

default

Logout a user / delete a session key

Method: DELETE
URL: <https://synopede.pw/nature-reviews/API/v1/sessions>
Requires token: Yes

This data should be sent in the body in JSON format. I am using the session key from the previous example, using session keys from these examples may not work for you.

```
{  
  "session_key": "osGxRhi4pIDf7wvEES2iDA=="  
}
```

The server returns the user's username. This is used to change the greeting from Guest to the username and do other tasks such as enable the comment box, remove the Login and Sign up buttons and display the Edit and Delete buttons for comments the user has made.

Response from the server:

No response text is returned from the server, but a "204 No Content" HTTP code will be sent.