DOCUMENTATION TO THE VOTING APP AND WEB SCRAPER BACKEND API ENDPOINTS

This document provides a simplified guide for frontend developers to connect with the available API endpoints in the backend.

Url to API: https://poll-app-b0bs.onrender.com, this service is on the free tier for the time being so boot up time of 50s is expected.

Link to GitHub repo: https://github.com/OG-Mendez/poll_app, please star the repository This

simple documentation can also be found in the repository.

Below are the endpoints, their expected input and output, and the HTTP methods used.

1. Sign Up

Endpoint: /api/signup/
Method: POST

Description: Registers a new user and returns an authentication token.

Request Body:

```
{
  "username": "new_user",
  "email": "new_user@example.com",
  "password": "securepassword"
}
```

Response:

Success 201:

```
{
  "message": "User created successfully",
  "token": "abc123def456ghi789"
}
```

Error 400:

```
{
  "error": "Username already taken",
  "error": "Email already taken"
}
```

2. Login

Endpoint: /api/login/ **Method**:
POST

Description: Authenticates a user and provides an authentication token.

Request Body:

```
{
  "username": "new_user",
  "password": "securepassword"
}
```

Response:

Success 200:

```
{
  "token": "abc123def456ghi789"
}
```

Error 400:

```
{
  "error": "Invalid credentials, please check to make sure the email and/or password is correct"
}
```

3. Logout

Endpoint: /api/logout/
Method: POST

Description: Logs out the user by deleting their authentication token.

Headers:

Authorization: Token abc123def456ghi789

Response:

Success 200:

```
{
  "message": "Logged out successfully" }
```

Error 400:

```
{
  "error": "Token not found"
}
```

4. Create Poll

Endpoint: /api/create_poll/
Method: POST

Description: Creates a new poll with multiple choices.

Request Body:

```
{
  "tag": "poll123",
  "question": "What's your favorite color?",
  "end_time": "2025-01-15T12:00:00Z",
  "choices": [
    {"A": "Red"},
    {"B": "Blue"},
    {"C": "Green"}
  ]
}
```

Response:

Success 201:

```
{
  "message": "Poll created successfully, write down the TAG: poll123 and the code: 123456"
}
```

Error 400:

```
{
  "error": "All fields are required"
}
```

5. Get Questions

Endpoint: /api/get_questions/
Method: GET

Description: Fetches the questions and choices for a given poll.

Response:

Success 200:

```
{
  "poll_id": "123456",
  "question": "What's your favorite color?",
  "choices": {
    "A": "Red",
```

```
        "B": "Blue",
        "C": "Green"
    }
}
```


Error 400:


```
{
    "error": "Poll not found"
}
```


6. Vote

Endpoint: /api/vote/

Method: POST

Description: Submits a vote for a given choice in a poll.

Parameters: tag and code

Request Body:


```
{
    "poll_id": "4",
    "choice": "A"
}
```


Response:

Success 200:


```
{
    "message": "Vote submitted successfully"
}
```


Error 404:


```
{
    "error": "Invalid tag or code",
    "error": "Invalid choice or question"
}
```

7. Get Results

Endpoint: /api/results/
Method: GET

Description: Fetches the results for a completed poll.

Response:

Success 200:


```
{
  "poll_id": "4",
  "question": "What's your favorite color?",
  "choices": {
    "A": 25,
    "B": 15,
    "C": 10
  }
}
```


Error 404:


```
{
  "error": "Poll not found or voting has not ended"}
```


WEB SCRAPER BACKEND API

Endpoint: /api/scraper/

Method: POST

Description: Scrapes the content from the provided URL based on the specified tags and returns a downloadable CSV file.

Response:

Success 200:


```
{
  "url": "https://quotes.toscrape.com/",
  "tags": "span.text, small.author",
}
```

Error 400:

```
{
   "error": "Bad Request"}
```

Note

- All requests that require authentication should include the Authorization header: Authorization: Token <your-token>
- Ensure the input fields match the specified keys exactly (e.g., username, password).
- Dates and times should be provided in ISO 8601 format (e.g., YYYY-MM-DDTHH:MM:SSZ).