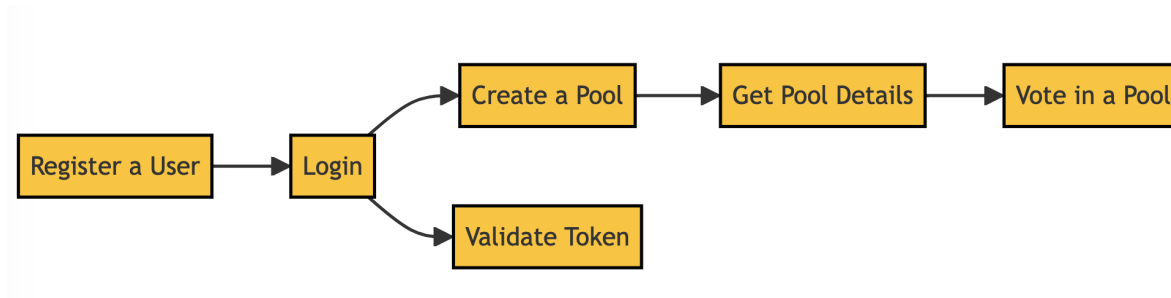


api.md

API Documentation

Introduction



This API documentation provides details about the endpoints, request/response formats, authentication/authorization mechanisms, and guidelines for the Realtime Pool Voting App.

Base URL

The base URL for all API endpoints is `https://your-api-url.com`.

Authentication

The Realtime Pool Voting App uses Firebase for authentication. To access protected endpoints, you need to include an `Authorization` header in your requests with a valid Firebase ID token.

Endpoints

Realtime Pool Voting App API 1.0.0 OAS 3.0

Servers

<https://your-api-url.com> ▾

default



POST

</auth/register> Register a User

POST

</auth/login> Login

POST

</auth/validate> Validate Token

POST

</pools> Create a Pool

GET

/pools/{pool_id} Get Pool Details

POST

/pools/{pool_id}/vote Vote in a Pool

1. Create a Pool

- URL: `/pools`
- Method: POST
- Description: Creates a new pool for voting.
- Request Body:

```
{
  "name": "Pool Name",
  "options": ["Option 1", "Option 2", "Option 3"]
}
```

- Response:

- Status: 201 Created

- Body:

```
{
  "id": "pool_id",
  "name": "Pool Name",
  "options": ["Option 1", "Option 2", "Option 3"],
  "votes": {
    "Option 1": 0,
    "Option 2": 0,
    "Option 3": 0
  }
}
```

2. Get Pool Details

- URL: /pools/{pool_id}
- Method: GET
- Description: Retrieves the details of a specific pool.
- Response:
 - Status: 200 OK
 - Body:

```
{
  "id": "pool_id",
  "name": "Pool Name",
  "options": ["Option 1", "Option 2", "Option 3"],
  "votes": {
    "Option 1": 10,
    "Option 2": 5,
    "Option 3": 3
  }
}
```

```
}  
}
```

3. Vote in a Pool

- URL: /pools/{pool_id}/vote
- Method: POST
- Description: Casts a vote for an option in a pool.
- Request Body:

```
{  
  "option": "Option 1"  
}
```

- Response:
 - Status: 200 OK
 - Body:

```
{  
  "id": "pool_id",  
  "name": "Pool Name",  
  "options": ["Option 1", "Option 2", "Option 3"],  
  "votes": {  
    "Option 1": 11,  
    "Option 2": 5,  
    "Option 3": 3  
  }  
}
```

4. Register a User

- URL: /auth/register
- Method: POST
- Description: Registers a new user.
- Request Body:

```
{  
  "email": "user@example.com",  
  "password": "password123"  
}
```

- Response:
 - Status: 201 Created
 - Body:

```
{  
  "id": "user_id",  
  "email": "user@example.com"  
}
```

5. Login

- URL: /auth/login
- Method: POST
- Description: Authenticates a user and returns an access token.
- Request Body:

```
{  
  "email": "user@example.com",  
  "password": "password123"  
}
```

- Response:
 - Status: 200 OK
 - Body:

```
{  
  "access_token": "your_access_token"  
}
```

6. Validate Token

- URL: /auth/validate
- Method: POST
- Description: Validates an access token.
- Request Body:

```
{  
  "access_token": "your_access_token"  
}
```

- Response:
 - Status: 200 OK
 - Body:

```
{  
  "valid": true  
}
```

Error Handling

In case of errors, the API will respond with appropriate status codes and error messages in the response body. Here are some common error scenarios:

- 400 Bad Request : Invalid request format or missing required fields.
- 401 Unauthorized : Missing or invalid Firebase ID token.
- 404 Not Found : Pool not found with the specified ID.

Conclusion

This API documentation provides an overview of the available endpoints and their functionalities for the Realtime Pool Voting App. Please ensure you follow the guidelines and include the necessary authentication headers when making requests to the API.