Project Documentation

Project Overview

This project is a task management system that uses Redis for queuing tasks and Express.js for the API. The backend is structured with a focus on scalability and performance, using a cluster setup to utilize multiple CPU cores.

Prerequisites

- Node.js and npm installed.
- Redis server running.
- Environment variables for Redis configuration.

Project Structure

- `app.js`: Main application file.
- `taskRouter.js`: Router for task-related routes.
- `taskController.js`: Controller handling task requests.
- `taskQueue.js`: Service for queuing tasks.
- `redisConfig.js`: Configuration for Redis connection.
- `logs/task.log`: Log file for task completion.

Running the Project

1. **Install dependencies**:

```

npm install

```

2. **Set up environment variables**:

   Create a `.env` file with the following variables:

```

REDIS_HOST=127.0.0.1

REDIS_PORT=6379

REDIS_USER=your_redis_username

REDIS_PASSWORD=your_redis_password

```

3. **Start the Redis server**:

   Ensure that the Redis server is running.

4. **Run the project**:

```

npm start

```

   This will start the server on port 5000.

API Endpoints

- **POST /api/v1/task**: Queue a task.
  - **Request Body**: `{ "user_id": "string" }`
  - **Response**: `{ "message": "Task queued successfully" }`

GitHub Repository

You can find the source code and contribute to the project on GitHub:

[GitHub Repository](https://github.com/ADI9325/RedisProject)

Live Deployment

The backend is live and deployed on Render.com:

[Live Backend on Render](https://redisproject.onrender.com)

Portfolio

You can view my portfolio here:

[Portfolio](https://itsaadi04.codeclout.in/)