

Overlay API Documentation

Introduction

This document provides detailed information about the CRUD endpoints for the Overlay API.

The API is built using Flask and interacts with a MongoDB database to store overlays.

Base URL

http://<your-domain>:<port>

Endpoints

1. **Create Overlay**

- **URL**: `/overlay``
- **Method**: `POST``
- **Description**: Create a new overlay.
- **Request Body**:
 - `content``: Text content of the overlay. (String)
 - `position``: Position of the overlay. Valid values are "Top Left", "Top Right", "Bottom Left", "Bottom Right", "Center". (String)
 - `x``: X-coordinate of the overlay. (Integer)
 - `y``: Y-coordinate of the overlay. (Integer)
 - `width``: Width of the overlay. (Integer)
 - `height``: Height of the overlay. (Integer)
- **Response**:
 - Success: `{ success: true, message: "Overlay created successfully", id: <overlay_id> }`
 - Failure: `{ success: false, message: <error_message> }`

2. **Get Overlays**

- **URL**: `/overlay``
- **Method**: `GET``

Overlay API Documentation

- **Description**: Fetch all overlays.
- **Response**: Array of overlays.

3. **Update Overlay**

- **URL**: `/overlay/<overlay_id>`
- **Method**: `PUT`
- **Description**: Update a specific overlay.
- **Request Body**:
 - `content`: Text content of the overlay. (String)
 - `position`: Position of the overlay. (String)
 - `x`: X-coordinate of the overlay. (Integer)
 - `y`: Y-coordinate of the overlay. (Integer)
 - `width`: Width of the overlay. (Integer)
 - `height`: Height of the overlay. (Integer)
- **Response**:
 - Success: `{ message: 'Overlay updated successfully' }`

4. **Delete Overlay**

- **URL**: `/overlay/<overlay_id>`
- **Method**: `DELETE`
- **Description**: Delete a specific overlay.
- **Response**:
 - Success: `{ message: 'Overlay deleted successfully' }`

5. **API Status**

- **URL**: `/`

Overlay API Documentation

- **Method**: `GET`
- **Description**: Check if the API is running.
- **Response**: "Overlay API is running!"

Conclusion

Ensure to replace ``<your-domain>`` and ``<port>`` with your actual domain and port number where the Flask application is hosted. Always remember to use the appropriate headers, such as ``Content-Type: application/json``, when making requests to the API.