Documentación de la API

Descripción

Esta API permite gestionar un sistema multiusuario que incluye usuarios, grupos y acciones. Las principales operaciones incluyen la creación de usuarios, grupos y acciones, así como la asignación de grupos a usuarios y acciones a grupos. También es posible obtener las relaciones entre usuarios y grupos. Explico la creación y el manejo de errores en cada caso.

Endpoints

Crear Usuario

Descripción: Crea un nuevo usuario.
URL: /usuarios
Método: POST
Request Body:

```
{
    "nombre": "string",
    "email": "string",
    "password": "string"
}
```

Respuesta de Éxito:

Código: 200

```
{
    "message": "Nuevo usuario creado!"
}
```

Respuesta de Error:

Código: 500

```
{
    "error": "Mensaje de error"
}
```

Crear Grupo

Descripción: Crea un nuevo grupo.
URL: /grupos
Método: POST

```
{
    "nombre": "string"
}
```

Respuesta de Éxito:

Código: 200

```
{
    "message": "Nuevo grupo creado!"
}
```

Respuesta de Error:

Código: 500

```
{
    "error": "Mensaje de error"
}
```

Crear Acción

Descripción: Crea una nueva acción.
URL: /acciones
Método: POST

```
{
    "nombre": "string"
}
```

Respuesta de Éxito:

Código: 200

```
{
    "message": "Nueva acción creada!"
}
```

Respuesta de Error:

Código: 500

```
{
    "error": "Mensaje de error"
}
```

CORS

Para permitir solicitudes CORS, los encabezados necesarios se agregan a las respuestas:

```
void add_cors_headers(Response& res) {
    res.set_header("Access-Control-Allow-Origin", "*");
    res.set_header("Access-Control-Allow-Methods", "POST, GET, OPTIONS");
    res.set_header("Access-Control-Allow-Headers", "Content-Type");
}
```

Manejo de Errores

La API maneja errores y responde con un mensaje JSON apropiado y un código de estado HTTP 500 en caso de excepciones:

```
catch (const sql::SQLException& e) {
    res.status = 500;
    json response = {{"error", e.what()}};
    res.set_content(response.dump(), "application/json");
    std::cerr << "SQL Exception: " << e.what() << std::endl;
} catch (const std::exception& e) {
    res.status = 500;
    json response = {{"error", e.what()}};
    res.set_content(response.dump(), "application/json");
    std::cerr << "Exception: " << e.what() << std::endl;
```

```
}
```