

Desarrollo de API RESTful para Manejo de Cuentas Bancarias

Objetivo: Desarrollar una API RESTful en Laravel para la gestión de cuentas bancarias, incluyendo la creación de cuentas, consulta de saldos, depósitos, retiros y consulta de historial de transacciones.

Funcionalidades Principales:

- 1. Endpoint para Crear una Nueva Cuenta Bancaria:**
 - `POST /api/accounts`
 - Crear una nueva cuenta bancaria con número de cuenta único y saldo opcional.
 - Respuesta: JSON con los detalles de la cuenta recién creada.
- 2. Endpoint para Consultar el Saldo de una Cuenta:**
 - `GET /api/accounts/{id}/balance`
 - Consultar el saldo actual de una cuenta específica.
 - Respuesta: JSON con el saldo actual de la cuenta.
- 3. Endpoint para Realizar un Depósito:**
 - `POST /api/transactions/deposit`
 - Realizar un depósito en una cuenta existente.
 - Parámetros: `account_id` (ID de la cuenta), `amount` (monto del depósito).
 - Respuesta: JSON con detalles de la transacción.
- 4. Endpoint para Realizar un Retiro:**
 - `POST /api/transactions/withdrawal`
 - Realizar un retiro de una cuenta existente.
 - Parámetros: `account_id` (ID de la cuenta), `amount` (monto del retiro).
 - Respuesta: JSON con detalles de la transacción.
- 5. Endpoint para Consultar el Historial de Transacciones:**
 - `GET /api/accounts/{id}/transactions`
 - Consultar el historial de transacciones de una cuenta específica.
 - Respuesta: JSON con un array de transacciones asociadas a la cuenta.

Puntos Extra (Opcionales):

- 1. Autenticación:**
 - Implementar autenticación para proteger las rutas sensibles.
 - Permitir el acceso solo a usuarios autenticados para realizar operaciones como depósitos, retiros y consultas de transacciones.
- 2. Manejo de Errores:**
 - Implementar manejo de errores adecuado, con códigos de estado HTTP y mensajes descriptivos.
 - Considerar casos como fondos insuficientes, solicitudes inválidas, etc.
- 3. Documentación:**
 - Documentar la API de manera clara y concisa, especificando los recursos disponibles y sus formatos de solicitud y respuesta.
 - Utilizar herramientas como Swagger o Laravel API Doc Generator para generar documentación interactiva.
- 4. Validación de Datos:**
 - Validar y sanitizar los datos de entrada para garantizar la integridad de los datos en la base de datos.

Notas Adicionales:

- Utilizar Eloquent para interactuar con la base de datos.
- Las respuestas de la API deben estar en formato JSON.
- Implementar control de errores y códigos de estado HTTP adecuados.
- Se espera seguir las convenciones RESTful en el diseño de las rutas.
- Se valora el uso de principios SOLID y buenas prácticas de desarrollo.

¡Éxito en el desarrollo de la API!