Truora_Colombia

1. Supón que un usuario reportó que al transferir dinero no se le descontó de su cuenta, pero si le llegó el dinero a otra persona. ¿Cómo pudo pasar eso?

En este issue reportado el tema radicaría en posibles issues a descartar:

Issue 1

Error de actualización en la BD es decir error al actualizar el saldo de la cuenta usuario que realiza la transferencia el dinero podría deducirse de la cuenta del remitente, pero no llegarse a reflejar en la cuenta del destinatario.

Issue 2

Se envió la petición con valor X en amount y no se resta funds que es donde se tiene los fondos totales del user

Es decir lo que ocurre en la siguiente línea from.Money -= amount

Es decir, se indica que se está restando cierta cantidad de dinero (que seria en la operación el amount), del saldo de la cuenta del usuario “From”, en este contexto from hace parte de la estructura y money es un campo que almacena el saldo de la cuenta del user.

Entonces ejm, si el saldo actual del usuario es from.money: 5, y amount en la operación fue 2 se ejecutaría la línea y arrojara el nuevo saldo. En caso contrario es que no existiera en ese momento exacto la resta de from.money, sobre el amount.

Entonces en palabras más claras. no existía la operación de resta en el código al momento de la ejecución de la transfer, por este motivo realiza la transferencia pero no resta de los fondos del usuario que la ejecuto.

Posible solución

sería necesario revisar el código para identificar posibles errores de lógica o condiciones de carrera (que una condición carrera es básicamente cuando se ejecutan dos transacciones del mismo usuario al mismo tiempo), también como tema preventivo seria implementar control de concurrencia para garantizar que las transferencias de dinero se realicen de manera segura y confiable y tener control sobre solo 1 sesión iniciada por 1 usuario(dispositivo).

2. Otro usuario reportó que pudo transferir más dinero de lo que tenía en la cuenta. ¿Puede ser posible? cómo?

Esto ocurriría según la documentación por la línea (from.Money < 0) y tendríamos diferentes escenarios a validar

Primer caso ejemplo

si dos solicitudes para transferir dinero de la misma cuenta se ejecutan simultáneamente, ambas podrían pasar la verificación de saldo disponible y luego deducir más dinero del que está realmente disponible en la cuenta.

Segundo caso ejemplo

No se esta validando como debería la variable amount, es decir, la condición no se cumple cuando intenta validar el amount entre los fondos del emisor a el monto a enviar.

Posible solución:

Como posibles soluciones a este bug seria, implementar algún tipo de bloqueo o sincronización para garantizar que solo una solicitud pueda modificar el saldo de la cuenta a la vez. Y esto podría lograrse dejando un único user activo por dispositivo para que no se ejecute dos transacciones al mismo tiempo

3. Posible Error en el reporte de usuario

la ruta solicitada no se encuentra definida en el enrutador la ruta /users/{id}.{format:(html|tsv)} parece estar correctamente definida en el enrutador a.Router, sin embargo se debería de tener en cuenta que añadiéndole el format generamos mas issues de los que solucionamos seria mejor en un json y convertirlo de ser necesario

también habría un posible error en findby FindBy del objeto a.Users esté correctamente implementada y sea capaz de manejar la búsqueda de usuarios por su ID.

Posible solución,

pasar de user id a user name dejand claro que seria una llave principal sin opción a repetir username, claro que esto se descarta con el token de inicio de sesión

Verifica que los métodos respond.HTML y respond.Respond estén correctamente implementados y sean capaces de enviar la respuesta adecuada al cliente.

4. Posible error en transferencia

En error transferencia al hacerlo a un Nick el cual puede llevar caracteres extraños o repetidos lo mejor seria hacerla directamente al ID de user al que se crea, como llave principal.

Esto solventaría transfers que se ejecuten a un user que no es por error en capa 8 (Error_humano).

Si el sistema maneja múltiples solicitudes de transferencia concurrentes, podría haber un problema de concurrencia que cause que los fondos se envíen a un destinatario incorrecto

Si no se implementan controles de acceso adecuados, un usuario malintencionado podría manipular la solicitud y enviar dinero a un destinatario incorrecto.

5. Posibles errores en la base de datos

En todas estas rutas, se realizan consultas a la base de datos para recuperar o modificar información de usuario. Si ocurre un error al interactuar con la base de datos, el controlador responde con un código de estado 500 "Internal Server Error". Sería útil manejar estos errores de manera más específica y devolver respuestas adecuadas al cliente.

Posible Solución

A medida que se vayan presentando issues se van documentando y clasificando para una mayor claridad al momento de consultar la base de datos

6. **Posible error en validación de datos de entrada**

POST /users: Aunque se realizan algunas comprobaciones básicas, como verificar que el nombre de usuario y la contraseña no estén vacíos, la validación de los datos de entrada podría ser más rigurosa.

Posible solución

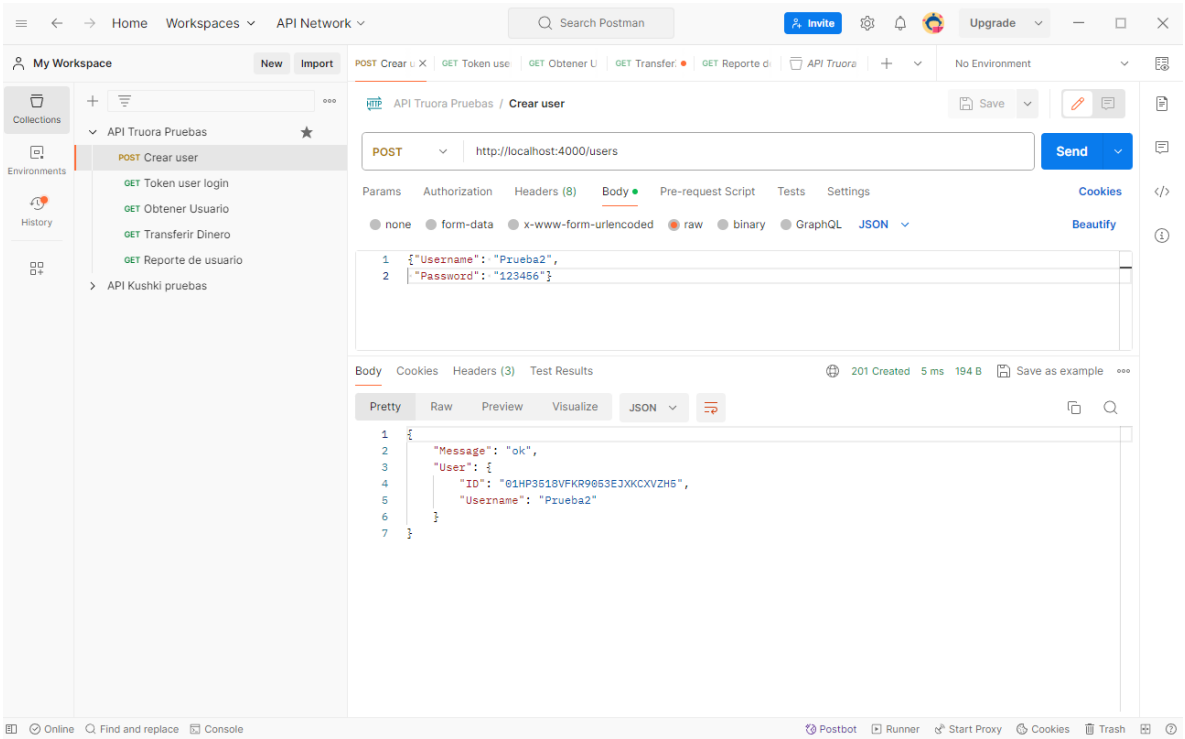
se podría realizar una validación más exhaustiva de la longitud de los campos y la complejidad de la contraseña.

Y en pro de solución de todos los erroes seria mejorar todos los detail_message:Error es decir dejarlos mas detallados y no tan general ayuda a dev, QA y soporte.

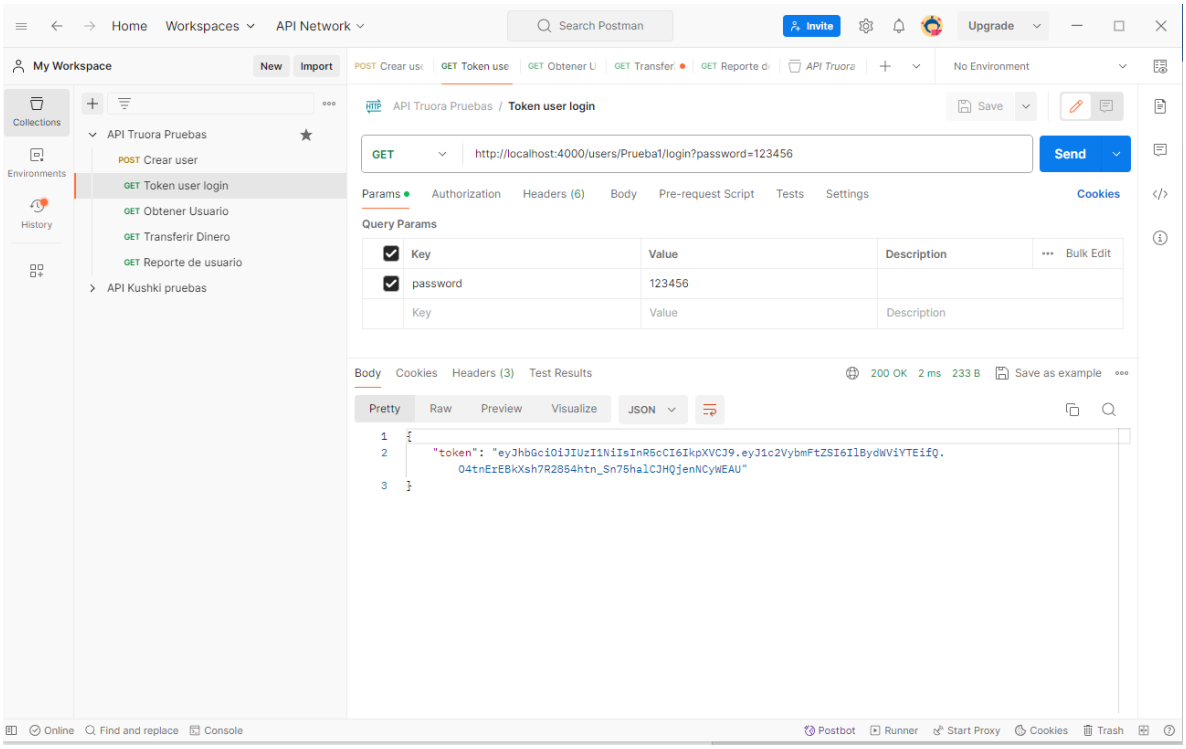
Como recomendación para los problemas reportados, existen servicios en la nube que proporcionan herramientas de integración que facilitan mas el proceso de consumo de API, SOAP y REST, es decir, mejor controladas, centralizadas y mejor auditoria para transaccionalidad, deriva en mayor facilidad de detección de bugs. Un servicio ejemplo para estas solvencias de bugs será implementar Amazon api Gateway.

Screenshot API _ GET / POST

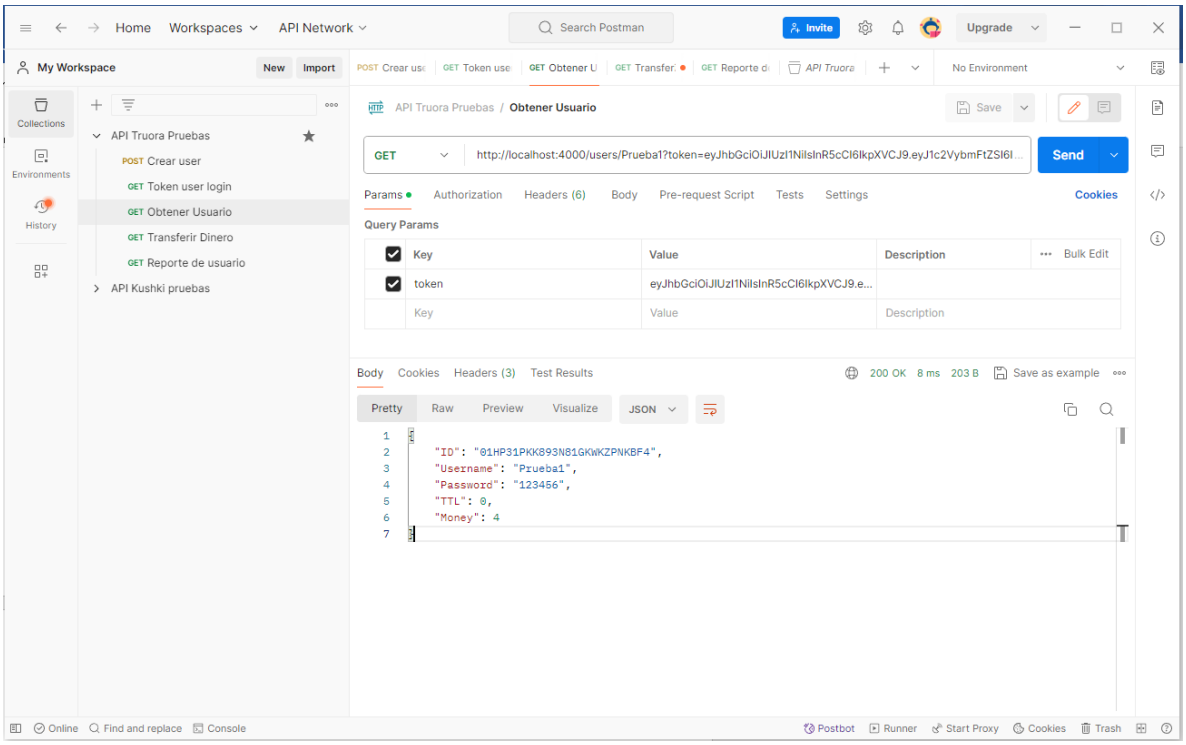
Crear Usuario



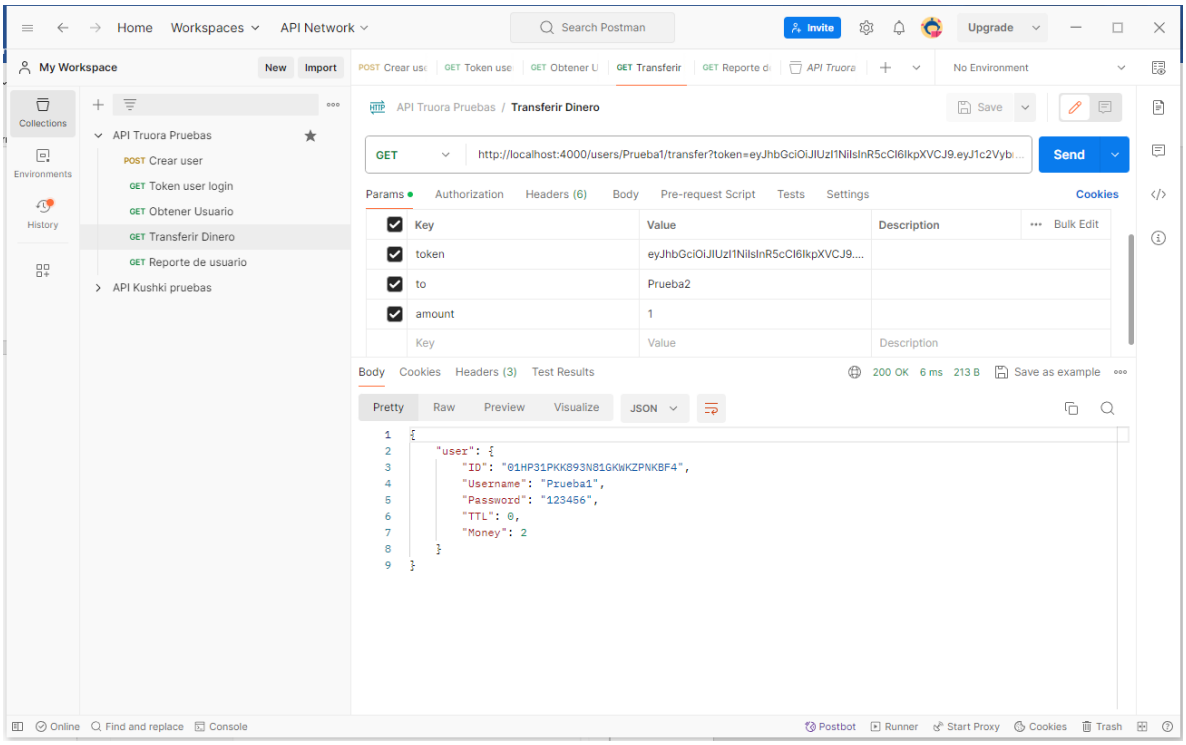
Inicio de sesión



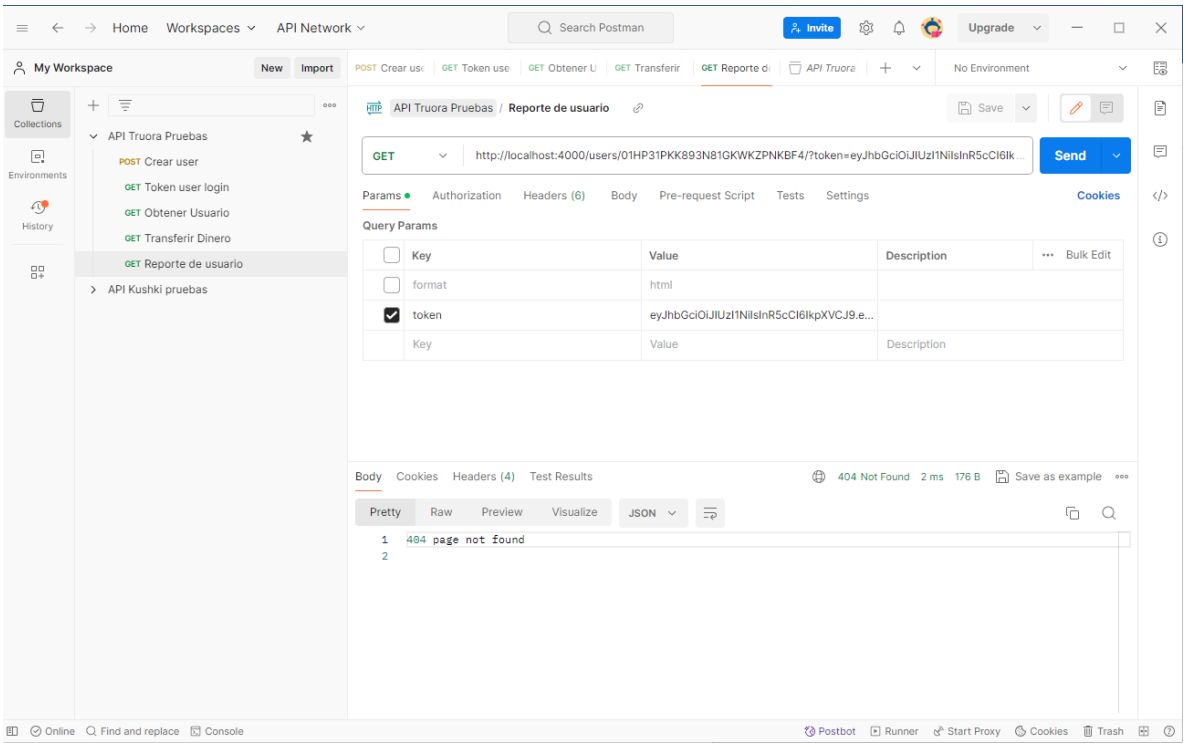
Obtener Usuario



Transferir Dinero



reporte de usuario



API Truora Pruebas.postman_collection.json

Conclusión, dejando documentados en dashboards en jira, redash, power bi, se identifica los issues/logs que el message:error no es claro y se trabajaría en documentar el proceso para escalar con dev y obtener mayor claridad del mismo, y así mismo de lado de tech se identificaría si es un issue que podrían detener desde L1 y tener a L2 en análisis de datos y código para solvencia de errores que mas nos afecten. Todo es análisis de datos el pro del GMV de cualquier compañía está en análisis de datos e issues.

Un plus es que desde L2 se podría trabajar en modificar directo en código los message error y que devs autoricen dicho cambio con eso liberamos tiempo de identificación y cambio de devs.

Seria un buen proyecto que a finalidad optimizaría tiempos.

¡Muchas gracias por la oportunidad!, muy interesante la prueba y llena de retos, pero aún mejor el resultado.