

Python Challenge

Matías, nos alegra que hayas querido formar parte de este proceso de selección para el puesto de Desarrollador Backend - Python.

En esta oportunidad, para conocer un poco más acerca de tus habilidades, te compartimos un pequeño challenge, el cual será analizado por el líder técnico, Juan Da Graca, y en base al resultado del mismo, avanzaríamos con una entrevista.

Sabe que no vamos a exigir la identificación de una solución óptima, aunque, en caso que la encuentres, será muy reconocido!!

La fecha límite de entrega es el próximo Lunes 14 de Marzo de 2022 a las 15hs y en caso que tengas con alguna duda, podrás consultar a Juan a través del siguiente correo:

jdagraca@ad-cap.com.ar.

Tene en cuenta que no podremos contestarte más de un mail con un límite de 5 consultas como máximo.

Consigna

Crear una API REST que brinde la siguiente funcionalidad básica de un sistema de control de movimientos monetarios. El mismo debe permitir crear y editar ingresos y egresos de dinero, y mostrar un balance resultante de las operaciones registradas.

Requerimientos Técnicos:

- Deberás desarrollar una API REST utilizando FastAPI
- Los datos mostrados deben ser persistidos en una base de datos relacional
- La API deberá exponer endpoints que devuelvan datos en JSON

El sistema debe tener los modelos Cliente, Movimiento, MovimientoDetalle y Cuenta con los siguientes atributos:

<u>Cliente</u> <ul style="list-style-type: none">- id PK- nombre [string]	<u>Cuenta</u> <ul style="list-style-type: none">- cliente [Cliente]- saldo_disponible [float]
<u>Movimiento</u> <ul style="list-style-type: none">- id PK- fecha [datetime]- cliente [Cliente]	<u>MovimientoDetalle</u> <ul style="list-style-type: none">- movimiento [Movimiento]- tipo [string]- importe [float]

La misma debe proporcionar los siguientes endpoints:

Cientes

- Registrar/Editar un cliente
- Eliminar un cliente
- Consultar un cliente
- Listar todos los clientes

Movimientos

- Registrar un movimiento. Debe actualizar el saldo disponible de la cuenta correspondiente al cliente que realiza la transacción, dependiendo del tipo de movimiento el saldo puede ser aumentado o disminuido.
- Eliminar un movimiento. Restaura saldo disponible del cliente.
- Consultar un movimiento y su detalle.

Saldos

- Consultar saldo disponible en cuenta del cliente.

Por otro lado, deben existir los siguientes métodos:

- La clase Movimiento debe exponer un método `get_total` el cual calcule el importe total del movimiento y retorne ese valor en el serializer correspondiente.
- La clase Cuenta debe exponer el método `get_total_usd` que valore el saldo disponible en dólares utilizando la cotización expuesta en la API de <https://www.dolarsi.com/api/api.php?type=valoresprincipales>, "Dolar Bolsa" y retorne el valor en el serializer correspondiente.

Consideraciones finales:

- Tene en cuenta que existen dos tipos diferentes de movimientos. Ingresos (impacto positivo en saldo disponible) y Egresos (impacto negativo en saldo disponible).
- Al crear un movimiento de tipo extracción se debe validar que haya suficiente saldo disponible en la cuenta del cliente, en caso de no contar con saldo suficiente se debe retornar un error de validación.
- No debe ser posible modificar el tipo de operación (ingreso o egreso) una vez creada.
- El código fuente debe ser subido a un repositorio público, el cual debe ser proporcionado para su correcta examinación.
- Implementar test unitario para validar los endpoints.