

1. Introducción

Este documento explica detalladamente el funcionamiento del archivo `main.py`, el cual define una pequeña API con FastAPI y SQLAlchemy.

2. Importaciones Principales

```
from fastapi import FastAPI, Depends
from sqlalchemy.orm import Session
from src import models
from src.database import SessionLocal
from sqlalchemy import select
```

Explicación:

- `FastAPI` : framework para crear APIs.
 - `Depends` : sistema de inyección de dependencias.
 - `Session` : clase de SQLAlchemy para manejar sesiones de base de datos.
 - `models` : archivo que contiene clases como `Movimiento`.
 - `SessionLocal` : función que nos da una sesión local de base de datos.
 - `select` : utilidad para realizar consultas SQL (no se usa en el archivo, pero puede ser parte de código futuro).
-

3. Creación de la Aplicación

```
app = FastAPI()
```

Explicación:

Se crea una instancia de FastAPI para comenzar a definir rutas de la API.

4. Dependencia para la Base de Datos

```
def get_db():
    db = SessionLocal()
    try:
        yield db
```

```
    finally:  
        db.close()
```

Explicación:

- `get_db()` es una función que genera una sesión a la base de datos.
- `yield` : permite usar esta función como dependencia en FastAPI.
- `db.close()` : garantiza que la conexión se cierra tras el uso.

5. Ruta para Obtener Movimientos

```
@app.get("/movimientos")  
def obtener_movimientos(db: Session = Depends(get_db)):  
    resultados = db.query(models.Movimiento).all()  
    return [  
        {  
            "id": m.id,  
            "tipo": m.tipo,  
            "categoria": m.categoría,  
            "monto": float(m.monto),  
            "fecha": m.fecha.isoformat()  
        }  
        for m in resultados  
    ]
```

Explicación Detallada:

5.1 Decorador `@app.get("/movimientos")`

- Define una ruta GET de la API.
- Se accede mediante el navegador o herramientas como Postman.

5.2 Función `obtener_movimientos(...)`

- Recibe una sesión `db` gracias a `Depends(get_db)`.

5.3 `db.query(models.Movimiento).all()`

- `db` : objeto sesión que permite consultar la base de datos.
- `query(...)` : crea una consulta sobre el modelo Movimiento.
- `all()` : devuelve una lista de objetos Movimiento.

5.4 Lista de Comprensión

```
[  
    { ... } # diccionario  
    for m in resultados  
]
```

- Se usa para transformar cada objeto Movimiento en un diccionario.
- Esto permite serializar fácilmente a formato JSON.

6. Serialización JSON

¿Por qué convertir a diccionario?

- FastAPI devuelve respuestas JSON.
- Los objetos SQLAlchemy no pueden ser convertidos directamente.
- Al usar `dict` se garantiza compatibilidad.

`monto: float(m.monto)`

- Convierte Decimal o tipos especiales a `float` (formato JSON compatible).

`fecha: m.fecha.isoformat()`

- Convierte `datetime` a texto ISO 8601 (`YYYY-MM-DDTHH:MM:SS`).

7. Lista de Comprensión

Definición:

Forma compacta de construir una lista a partir de otra estructura iterable.

Ejemplo:

```
cuadrados = [x**2 for x in range(5)]
```

En nuestro código:

```
[{  
    ...  
} for m in resultados]
```

- Forma clara y breve de transformar cada `m` (Movimiento) en un diccionario.

8. Alternativa sin Lista de Comprensión

Código equivalente:

```
lista = []  
for m in resultados:  
    lista.append({  
        "id": m.id,  
        "tipo": m.tipo,  
        "categoria": m.categoría,  
        "monto": float(m.monto),  
        "fecha": m.fecha.isoformat()  
    })  
return lista
```

Comparación:

- **Lista de comprensión:** más compacta, legible para programadores con experiencia.
- **For tradicional:** más explícito, puede ser más claro para principiantes.

9. json.dumps(...)

- Convierte un diccionario o lista en un texto JSON.
- No se usa en este archivo porque FastAPI lo hace internamente.

10. Acceso a Propiedades del Modelo

```
m.id, m.tipo, m.categoría, ...
```

- `m` es una instancia de `Movimiento`.
- Accedemos a sus propiedades como a cualquier objeto Python.

11. Conclusión

- El archivo `main.py` define una ruta GET sencilla.
 - Utiliza SQLAlchemy para acceder a la base de datos.
 - Convierte los datos a JSON-compatible con una lista de comprensión.
 - Muy útil como punto de partida para una API REST moderna.
-

Puedes convertir este documento a PDF u ODP para usarlo como guía visual.