```
from fastapi import FastAPI,
HTTPException
from pydantic import BaseModel
from typing import List
import motor.motor_asyncio as aiomotor

app = FastAPI()

# Connect to MongoDB
client =
aiomotor.AsyncIOMotorClient("mongodb://
localhost:27017")
db = client.my_database
users_collection = db.users

class User(BaseModel):
    name: str
    age: int

@app.post("/users/",
response_model=User)
async def create_user(user: User):
    user_dict = user.dict()
    result = await
users_collection.insert_one(user_dict)
    user_dict["id"] = str(result.inserted_id)
    return user_dict

@app.get("/users/",
response_model=List[User])
async def read_users():
    cursor = users_collection.find()
    users = []
    async for user in cursor:
        user["id"] = str(user["_id"])
        del user["_id"]
        users.append(user)
    return users

@app.get("/users/{user_id}",
response_model=User)
async def read_user(user_id: str):
    user = await
users_collection.find_one({"_id": user_id})
    if user:
        user["id"] = user_id
        del user["_id"]
        return user
    else:
        raise
HTTPException(status_code=404,
detail="User not found")

@app.put("/users/{user_id}",
response_model=User)
async def update_user(user_id: str, user:
User):
    result = await
users_collection.replace_one({"_id":
user_id}, user.dict())
    if result.matched_count == 1:
        return user
    else:
        raise
HTTPException(status_code=404,
detail="User not found")

@app.delete("/users/{user_id}")
```

```
async def delete_user(user_id: str):
    result = await users_collection.delete_one({"_id": user_id})
    if result.deleted_count == 1:
        return {"detail": "User deleted"}
    else:
        raise HTTPException(status_code=404, detail="User not found")
```