

# POST Endpoint with Input Validation



```
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel, Field


app = FastAPI()

class User(BaseModel):
    username: str = Field(..., min_length=3)
    email: str

@app.post("/users/")
def create_user(user: User):
    if "@" not in user.email:
        raise HTTPException(status_code=400, detail="Invalid email address")
    return {"message": "User created successfully", "user": user}
```



# PUT Endpoint for Updating Resources:



```
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel

app = FastAPI()

class Item(BaseModel):
    name: str
    description: str

items_db = {}

@app.put("/items/{item_id}")
def update_item(item_id: int, item: Item):
    if item_id not in items_db:
        raise HTTPException(status_code=404, detail="Item not found")
    items_db[item_id] = item
    return {"message": "Item updated successfully", "item": item}
```