

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}
```

Sharing Demo Code on CodeSandbox

- Organize your project: Structure your code so it's easy to understand and modify. Use a modular approach with separate files for models, routes, and services.
- Document your code: Add comments and docstrings explaining each function, class, and endpoint.
- Using CodeSandbox:
 - Create a new sandbox for your FastAPI project in CodeSandbox.
 - Add your backend files in the server directory.
 - Run and test the API directly from CodeSandbox using the integrated terminal.