

```
# Secure Guest Upload API for Synology NAS (guest-container access)
```

```
from fastapi import FastAPI, File, UploadFile, HTTPException, Depends
```

```
from fastapi.security import HTTPBasic, HTTPBasicCredentials
```

```
from pathlib import Path
```

```
import shutil
```

```
import os
```

```
app = FastAPI()
```

```
security = HTTPBasic()
```

```
# Constants
```

```
UPLOAD_ROOT = Path("/volume1/guest-container") # Must match NAS mount path for guest
```

```
ALLOWED_USERS = {"guestvpn": "securepassword"} # Replace with secure credential store
```

```
# Ensure upload path exists
```

```
UPLOAD_ROOT.mkdir(parents=True, exist_ok=True)
```

```
# Authentication dependency
```

```
def authenticate(credentials: HTTPBasicCredentials = Depends(security)):
```

```
    correct_password = ALLOWED_USERS.get(credentials.username)
```

```
    if not correct_password or credentials.password != correct_password:
```

```
        raise HTTPException(status_code=401, detail="Invalid authentication")
```

```
    return credentials.username
```

```

# Upload endpoint
@app.post("/upload")
def upload_file(
    file: UploadFile = File(...),
    username: str = Depends(authenticate)
):
    target_path = UPLOAD_ROOT / file.filename
    with target_path.open("wb") as buffer:
        shutil.copyfileobj(file.file, buffer)
    return {"message": f"File '{file.filename}' uploaded successfully."}

```

Optional: Restrict filenames / extensions

```

@app.post("/secure-upload")
def secure_upload(
    file: UploadFile = File(...),
    username: str = Depends(authenticate)
):
    if not file.filename.endswith((".pdf", ".docx", ".xlsx")):
        raise HTTPException(status_code=400, detail="File type not allowed.")

    safe_name = file.filename.replace("../", "").replace("/", "_")
    target_path = UPLOAD_ROOT / safe_name

    with target_path.open("wb") as buffer:
        shutil.copyfileobj(file.file, buffer)

```

```
return {"status": "success", "filename": safe_name}
```

```
# Deployment hint (e.g. via uvicorn)
```

```
# uvicorn main:app --host 0.0.0.0 --port 8080 --reload
```

```
# Notes:
```

```
# 1. Place this script on your NAS in a Docker container or virtualized Python environment.
```

```
# 2. Ensure proper firewall and port forwarding (TCP 8080) only for trusted IPs.
```

```
# 3. Optionally log all access and uploads for compliance.
```