

تمام. هذه **الخطوات النهائية النظيفة** (بدون خلط قديم/جديد). نقذها حرفياً في مستودع صديقك أو أي مستودع جديد.

---

## الهدف النهائي

الملفات كلها داخل مجلد . documents/. الملفات كلها داخل مجلد . documents/index.json يولد ملف فهرسة تلقائياً GitHub Actions . documents/index.html تقرأ files/index.json و تعرض الملفات (PDF/ صفحة . HTML/JSON...).

---

## الخطوة 1: جهز هيكل المستودع

داخل المستودع اجعل هذا الشكل :

```
/index.html  
/favicon.ico    (اختياري)  
/documents/  
      pdf. (مثال) تعليمات  
/.github/  
  /workflows/  
    generate-documents-index.yml
```

وتقدر تسوى مجلدات داخله documents/ مهم: كل الملفات التي تريد عرضها تكون داخل : مثل

documents/ أول ابتدائي/ ملف1 pdf

documents/ ثاني ابتدائي/ ملف2 pdf

documents/ ملفات عامة/ ...

## تلقائيا GitHub Action الخطوة 2: أضف ملف index.json توليد )

أنشئ هذا الملف بالمسار التالي بالضبط :

.github/workflows/generate-documents-index.yml

loop) هذه النسخة "النهائية" وتمنع الـ ضع بداخله هذا المحتوى:

```
name: Generate documents index.json on: push: branches: ["main"]
paths: - "documents/**" - ".github/workflows/generate-documents-
index.yml" paths-ignore: - "documents/index.json" workflow_dispatch:
permissions: contents: write jobs: build-index: runs-on: ubuntu-
latest steps: - name: Checkout uses: actions/checkout@v4 - name:
Generate documents/index.json run: | python3 - <<'PY' import json
import pathlib root = pathlib.Path("documents") files = [] if
root.exists(): for p in root.rglob("*"): if p.is_file() and not
p.name.startswith(".") and p.name != "index.json": rel =
p.relative_to(root).as_posix() parts = rel.split("/") folder =
parts[0] if len(parts) > 1 else "filفات عامة" files.append({ "name":
p.name, "path": rel, "folder": folder }) files.sort(key=lambda x:
(x["folder"], x["name"])) out = root / "index.json"
out.write_text(json.dumps(files, ensure_ascii=False, indent=2),
encoding="utf-8") print(f"Wrote {out} with {len(files)} files") PY -
name: Commit index.json uses: stefanzweifel/git-auto-commit-
action@v5 with: commit_message: "Auto-generate documents/index.json"
file_pattern: "documents/index.json"
```

---

## 3: ارفع أي ملف داخل documents لتشغيل الـ Action

مثلاً ارفع documents/تعليمات.pdf

إلى main . Push ثم اعمل

بعدها:

ادخل GitHub → تبويب Actions

ستجد Workflow شغال

بعد نجاحه ستري ملف جديد ظهر تلقائياً documents/index.json

---

## 4: ضع صفحة العرض (index.html)

في جذر المستودع index.html انسخ نسختك النهائية التي تعمل عندك وضعها كملف.

أن تكون هذه القيم موجودة HTML المهم داخل الـ:

## A) الفهرس

```
const INDEX_URL = "documents/index.json";
```

## B) زر "فتح في تبويب جديد" للـ HTML

لازم يكون displayFile داخل:

```
if (["html", "htm"].includes(ext)) { openTabBtn.href = fileUrl;
openTabBtn.style.display = "inline-flex"; } else {
openTabBtn.style.display = "none"; }
```

## C) عرض JSON داخل

لازم يكون displayFile :

```
if (ext === "json") { displayJson(fileUrl); return; }
```

وموجود CSS:

```
#viewer { overflow: auto; align-items: stretch; justify-content: stretch; }
.json-viewer { white-space: pre; overflow: auto; direction: ltr; text-align: left; }
```

## D) ترميز المسار (ضروري للأسماء العربية والمسارات)

```
function toUrlPath(filepath){ return
String(filepath).split("/").map(encodeURIComponent).join("/"); }
```

واستخدامه:

```
const fileUrl = `documents/${toUrlPath(filepath)}`;
```

## خطوة 5: فَعْل GitHub Pages

من إعدادات المستودع:

Settings → Pages

Source: Deploy from a branch

Branch: main

Folder: / (root)

ثم ادخل الرابط الذي يعطيه لك GitHub Pages.

---

## تشغيل سريع للتأكد

فتح الموقع.

يجب ظهر → PDF اختر ملف.

يجب ظهر + زر فتح تبويب جديد يظهر → HTML اختر ملف.

يجب ظهر كنص منسق مع سكريول → JSON اختر ملف.

---

## ماذا تقول لصديقك باختصار؟

”فقط documents وضع ملفاتك داخل“

”سيولد Action بيدك، الا index.json لا تكتب“

”سيحدث الفهرس تلقائياً documents / أي إضافة/حذف داخل“

”في الجذر index.html صفحة العرض هي“

---

ثم يغير Template إذا تريدها أسهل على صديقك: قل له ينسخ مستودعك ك فقط documents/ محتوى.