**🧩 功能目標：Check Login REST API**

* 帳號密碼儲存在 D1（SQLite）
* 提供 API：
  + POST /login，接收 JSON { username, password }
  + 回傳是否正確，例如 { success: true }

**✅ 系統設計概述**

**1. Cloudflare 元件**

| **元件** | **說明** |
| --- | --- |
| D1 | 儲存使用者帳密（建議加密密碼） |
| Workers | 提供 REST API 給前端 |
| GitHub Pages | 放前端頁面（可用 fetch 呼叫 API） |

**2. D1 資料表設計範例**

sql

複製編輯

CREATE TABLE users (

id INTEGER PRIMARY KEY AUTOINCREMENT,

username TEXT UNIQUE NOT NULL,

password\_hash TEXT NOT NULL

);

**🚀 Step-by-step 實作教學（登入驗證 API）**

**🔧 假設你已經有：**

* 一個 Cloudflare 帳號
* 一個已建立的 D1 資料庫（名為例如 usersdb）

**📝 Workers 程式碼（login API）**

ts

複製編輯

// worker.js

export default {

async fetch(request, env, ctx) {

const url = new URL(request.url);

if (url.pathname === "/login" && request.method === "POST") {

const { username, password } = await request.json();

// 查詢使用者密碼 hash

const stmt = env.DB.prepare("SELECT password\_hash FROM users WHERE username = ?");

const result = await stmt.bind(username).first();

if (!result) {

return Response.json({ success: false, message: "User not found" }, { status: 404 });

}

// 這裡簡化為明碼比對，建議使用 bcrypt

if (result.password\_hash === password) {

return Response.json({ success: true });

} else {

return Response.json({ success: false, message: "Wrong password" }, { status: 401 });

}

}

return new Response("Not Found", { status: 404 });

}

};

**🧪 建立 D1 資料庫與測試資料**

用 Wrangler CLI：

bash

複製編輯

# 建立資料庫

wrangler d1 create usersdb

# 連線管理

wrangler d1 execute usersdb --command "

CREATE TABLE users (id INTEGER PRIMARY KEY, username TEXT UNIQUE, password\_hash TEXT);

INSERT INTO users (username, password\_hash) VALUES ('alice', 'secret123');

"

⚠️ 正式應用中你應使用 bcrypt 等方式儲存加密後的密碼。

**🛜 wrangler.toml 設定**

toml

複製編輯

name = "login-api"

compatibility\_date = "2024-06-01"

[[d1\_databases]]

binding = "DB"

database\_name = "usersdb"

database\_id = "xxxx-xxxx" # 用 `wrangler d1 list` 查到的

**🧪 前端 JS 呼叫範例**

js

複製編輯

async function checkLogin(username, password) {

const res = await fetch("https://your-worker-url.workers.dev/login", {

method: "POST",

headers: { "Content-Type": "application/json" },

body: JSON.stringify({ username, password }),

});

const result = await res.json();

if (result.success) {

alert("Login OK");

} else {

alert("Login Failed: " + (result.message || "Unknown"));

}

}