/\*\*

\* =============================================================================

\* @file 00\_initializer.gs

\* @module GAssistant.Initializer

\* @description

\* هذا الملف هو نقطة الانطلاق وحجر الأساس للمشروع بأكمله.

\*

\* الوظائف الرئيسية:

\* 1. \*\*إنشاء مساحة الاسم:\*\* يقوم بتعريف الكائن العام `GAssistant` مرة واحدة فقط،

\* لتجنب تلوث النطاق العام (Global Scope) وضمان تنظيم الكود.

\* 2. \*\*تهيئة الوحدات الفرعية:\*\* يقوم بإنشاء الكائنات الفارغة للوحدات الرئيسية

\* (مثل `AI`, `UI`, `Tools`) لضمان وجودها قبل أن تحاول الملفات الأخرى

\* إضافة خصائص إليها، مما يمنع أخطاء "cannot set property of undefined".

\* 3. \*\*نقطة الدخول عند الفتح:\*\* يحتوي على دالة `onOpen` التي يتم استدعاؤها

\* تلقائيًا بواسطة Google Sheets، والتي بدورها تشغل الفحوصات الأولية

\* ثم تهيئ واجهة المستخدم.

\*

\* @architectural\_note يجب أن يكون هذا الملف هو أول ملف يتم تحميله في

\* مشروع Google Apps Script لضمان تهيئة البيئة بشكل صحيح.

\* =============================================================================

\*/

// تعريف الكائن العام للمشروع. هذا هو المدخل الوحيد في النطاق العام.

var GAssistant = typeof GAssistant !== "undefined" ? GAssistant : {};

// تهيئة مساحات الأسماء الأساسية لضمان وجودها.

GAssistant.AI = GAssistant.AI || {};

GAssistant.UI = GAssistant.UI || {};

GAssistant.Tools = GAssistant.Tools || {};

GAssistant.Utils = GAssistant.Utils || {};

GAssistant.Security = GAssistant.Security || {};

GAssistant.Analytics = GAssistant.Analytics || {};

GAssistant.Tests = GAssistant.Tests || {};

GAssistant.Agents = GAssistant.Agents || {};

GAssistant.API = GAssistant.API || {};

GAssistant.Types = GAssistant.Types || {}; // لإدارة تعريفات الأنواع

/\*\*

\* دالة Apps Script القياسية التي تُنفذ عند فتح المستند.

\* تم تعديلها لتشمل فحص سلامة النظام قبل إنشاء القوائم.

\* ملاحظة: تم نقل onOpen هنا من Code.gs لأنها نقطة الدخول الطبيعية للتطبيق.

\*/

function onOpen() {

// تشغيل الفحوصات أولاً لضمان سلامة البيئة.

// إذا فشلت الفحوصات، سيتوقف التنفيذ ولن يتم إنشاء القائمة.

if (!runPreflightChecks()) {

SpreadsheetApp.getUi().alert("فشل في تهيئة G-Assistant. يرجى مراجعة سجلات التنفيذ (Execution logs) لمزيد من التفاصيل.");

return;

}

// بعد التأكد من أن كل شيء سليم، يتم إنشاء واجهة المستخدم.

// يتم التحقق من وجود الوحدة لتجنب الأخطاء إذا فشل تحميلها.

const UI = GAssistant.UI;

if (UI && typeof UI.onOpen === 'function') {

UI.onOpen();

} else {

console.error("CRITICAL: GAssistant.UI.onOpen is not available. The UI cannot be initialized.");

// عرض رسالة خطأ بسيطة للمستخدم في حالة الفشل الكارثي.

SpreadsheetApp.getUi().alert("خطأ فادح في تهيئة واجهة المستخدم للمساعد.");

}

}

/\*\*

\* يقوم بتشغيل سلسلة من الفحوصات التشخيصية لضمان سلامة بيئة المشروع.

\* هذه الدالة هي "حارس البوابة" للمشروع.

\* @returns {boolean} - true إذا نجحت جميع الفحوصات، وإلا false.

\*/

function runPreflightChecks() {

// لا يمكننا الاعتماد على GAssistant.Utils.log هنا لأنه قد لا يكون قد تم تحميله بعد

// لذا نستخدم console.log مباشرة.

console.log("Initializer: Running Preflight Checks...");

// 1. التحقق من وجود الوحدات الأساسية

const coreModules = ['Config', 'Utils'];

for (const mod of coreModules) {

if (typeof GAssistant[mod] === 'undefined') {

console.error(`PREFLIGHT CHECK FAILED: Core module "GAssistant.${mod}" is not loaded. Halting.`);

return false; // إيقاف الفحص إذا كانت الوحدات الأساسية مفقودة

}

}

console.log("Preflight Check PASSED: Core modules (Config, Utils) are loaded.");

// الآن يمكننا استخدام Utils بأمان

const { error, log } = GAssistant.Utils;

// 2. التحقق من صحة ملف التكوين (مفاتيح API الأساسية)

const Config = GAssistant.Config;

const config = Config.getAll();

const requiredKeys = ['API\_KEY', 'API\_ENDPOINT', 'GEMINI\_PRO\_MODEL'];

for (const key of requiredKeys) {

if (!config[key] || (typeof config[key] === 'string' && config[key].trim() === '')) {

error(`Preflight Check FAILED: Configuration key "${key}" is missing or empty in Script Properties.`);

return false;

}

}

log("Preflight Check PASSED: Essential configuration keys are present.");

// 3. التحقق من وجود بقية الوحدات الرئيسية (اختياري لكن جيد)

const otherModules = ['UI', 'AI.Core', 'Tools.Catalog'];

for (const mod of otherModules) {

const path = mod.split('.');

let current = GAssistant;

let found = true;

for (const p of path) {

if (typeof current[p] === 'undefined') {

found = false;

break;

}

current = current[p];

}

if(!found) {

warn(`Preflight Check WARNING: Module "GAssistant.${mod}" is not fully loaded. Some functionalities may be unavailable.`);

}

}

log("Preflight Checks completed successfully.");

return true; // كل شيء على ما يرام

}