/\*\*

\* @fileoverview ملف البيئة (Environment & Configuration) - v16.2

\* @module GAssistant.Config

\* @description

\* يحتوي على جميع الإعدادات العامة والثابتة للمساعد والمشروع،

\* منظمة ضمن مساحة اسم GAssistant.Config.

\* هذا الملف هو المرجع المركزي للإعدادات المشتركة في ai\_core.gs و Sidebar وغيرها.

\* تم التوسيع والتعليق الكامل لجميع الثوابت وتطبيق معايير Clear Namespacing,

\* Dynamic Access Control, و Secret Store Integration.

\*/

// تعريف الكائن العام GAssistant إذا لم يكن موجودًا بالفعل

var GAssistant = GAssistant || {};

// تعريف وحدة الإعدادات والتكوين ضمن GAssistant

GAssistant.Config = (function() {

'use strict';

// ⏱️ مدة صلاحية الكاش بالثواني (30 دقيقة)

const CACHE\_DURATION\_SECONDS = 1800;

// 📦 خدمة الكاش لـ Apps Script

const CACHE = CacheService.getDocumentCache();

/\*\*

\* @const {object} \_DEFAULT\_SETTINGS

\* @description

\* كائن يحتوي على جميع الإعدادات الافتراضية والثابتة للنظام.

\* يتم قراءة بعض هذه الإعدادات ديناميكيًا من ورقة 'التعليمات الرئيسية'.

\* تم تسميته بـ `\_DEFAULT\_SETTINGS` للإشارة إلى أنه خاص بالوحدة.

\*/

const \_DEFAULT\_SETTINGS = (function () {

return {

ENVIRONMENT: "production", // يمكن تغييره إلى 'staging' أو 'development' عند الاختبار

DEBUG\_MODE: true , // ✅ إذا true: يُسجّل التنبيهات والمفاتيح الناقصة

SYSTEM\_LANGUAGE: "ar", // اللغة الافتراضية للنظام

MASTER\_INSTRUCTIONS\_SHEET\_NAME: "التعليمات الرئيسية", // اسم ورقة الإعدادات الرئيسية

OPERATION\_LOG\_SHEET\_NAME: "سجل العمليات", // اسم ورقة سجل العمليات

MAX\_LOG\_ENTRIES: 1000, // الحد الأقصى لعدد إدخالات السجل

MAX\_LOG\_ENTRIES\_FOR\_CONTEXT: 5, // عدد إدخالات السجل التي يتم تضمينها في سياق النموذج

DEFAULT\_MODEL: "gemini-1.5-pro", // النموذج الافتراضي لـ Gemini

MODEL\_FOR\_SIMPLE\_TASKS: "gemini-1.5-flash", // النموذج السريع للمهام البسيطة

ALLOWED\_MODELS: ["gemini-1.5-pro", "gemini-1.5-flash", "gemini-1.0-pro"], // النماذج المسموح بها

API\_ENDPOINT: "https://generativelanguage.googleapis.com/v1beta/models/", // نقطة نهاية Gemini API

API\_KEYS\_NAMES: { // أسماء مفاتيح API في Script Properties

GEMINI: "GEMINI\_API\_KEY",

OPENAI: "OPENAI\_API\_KEY"

},

GENERATION\_CONFIG: { // إعدادات توليد النصوص لـ Gemini

temperature: 0.7,

topP: 0.95,

topK: 60,

maxOutputTokens: 2048,

responseMimeType: "text/plain",

},

SAFETY\_SETTINGS: [ // إعدادات الأمان لـ Gemini

{ category: "HARM\_CATEGORY\_HARASSMENT", threshold: "BLOCK\_NONE" },

{ category: "HARM\_CATEGORY\_HATE\_SPEECH", threshold: "BLOCK\_NONE" },

{ category: "HARM\_CATEGORY\_SEXUALLY\_EXPLICIT", threshold: "BLOCK\_NONE" },

{ category: "HARM\_CATEGORY\_DANGEROUS\_CONTENT", threshold: "BLOCK\_NONE" },

],

TOOLS\_ENABLED: true, // هل تمكين استخدام الأدوات (Function Calling)

TOOL\_CONFIG: { // إعدادات الأدوات (مثل thinkingConfig)

"functionCallingConfig": {

"mode": "AUTO"

},

"thinkingConfig": {

"enableThinking": true, // تفعيل وضع التفكير

"thinkingModel": "gemini-1.5-flash", // نموذج لوضع التفكير

"thinkingTemperature": 0.2,

"thinkingMaxOutputTokens": 512

}

},

// أسماء أوراق العمل المستخدمة في المشروع

SALES\_SHEET: "المبيعات",

EXPENSES\_SHEET: "المصروفات",

ACCOUNTS\_SHEET: "الحسابات",

CLIENTS\_SHEET: "العملاء",

VENDORS\_SHEET: "الموردين",

DAILY\_SUMMARY\_SHEET: "الملخص اليومي",

DATE\_FORMAT: "yyyy-MM-dd", // تنسيق التاريخ الافتراضي

DEFAULT\_TAX\_RATE: 0.15, // معدل الضريبة الافتراضي

PROTECTED\_RANGES: ["A1:Z1"], // نطاقات محمية افتراضياً (مثل صف الرؤوس)

MAX\_ENTRIES\_PER\_DAY: 100, // الحد الأقصى للإدخالات اليومية

ALLOWED\_USERS: [], // قائمة المستخدمين المسموح لهم بالوصول (فارغة تعني الكل)

DEFAULT\_CURRENCY: "SAR", // رمز العملة الافتراضي

// إعدادات الواجهة الأمامية (Sidebar)

SIDEBAR\_FILE: "Sidebar", // اسم ملف HTML للشريط الجانبي الرئيسي

UNIFIED\_SIDEBAR\_FILE: "BuiltinDocsSidebar", // اسم ملف HTML للشريط الجانبي الموحد (توثيق ومراجعة)

// ألوان التنسيق الافتراضية

PRIMARY\_HEADER\_COLOR: "#d9ead3", // لون رؤوس الجداول

HIGHLIGHT\_COLOR: "#fce8e6", // لون التمييز في الجداول

// قوالب أوراق العمل (لإنشاء أوراق جديدة برؤوس محددة)

TEMPLATES: {

SALES: {

SHEET\_NAME: "المبيعات",

HEADERS: ["التاريخ", "المنتج", "الكمية", "سعر الوحدة", "الإجمالي", "العميل", "الوصف"]

},

EXPENSES: {

SHEET\_NAME: "المصروفات",

HEADERS: ["التاريخ", "الوصف", "المبلغ", "الفئة", "المورد"]

},

// يمكن إضافة المزيد من القوالب هنا

},

// إعدادات ورشة العمل (WORKSHOP\_TOOLS)

STAGING\_SHEET\_NAME: 'Developer Workshop', // اسم ورقة ورشة عمل المطور

TOOL\_FILE\_SUGGESTION\_PATTERNS: [ // أنماط لاقتراح الملفات في ورشة العمل

{ keywords: ['getactivespreadsheet', 'spreadsheetapp', 'createmenu', 'onopen', 'onsidebar'], file: 'ui.gs', weight: 10 },

{ keywords: ['getrange', 'setvalue', 'appendrow', 'sheet', 'range', 'getvalues', 'setvalues'], file: 'tools\_sheets.gs', weight: 9 },

{ keywords: ['revenue', 'expense', 'profit', 'currency', 'income', 'cost', 'ledger'], file: 'tools\_accounting.gs', weight: 8 },

{ keywords: ['logoperation', 'error', 'message', 'dialogue', 'cache'], file: 'helpers.gs', weight: 7 },

{ keywords: ['model', 'gemini', 'api', 'prompt', 'ask'], file: 'ai\_core.gs', weight: 7 },

{ keywords: ['context', 'session', 'remember', 'recall', 'memory'], file: 'ai\_memory.gs', weight: 6 },

{ keywords: ['message', 'show', 'alert', 'dialogue', 'prompt'], file: 'ai\_tools\_dialogue.gs', weight: 6 },

{ keywords: ['test', 'assert', 'expect', 'runtest'], file: 'tests.gs', weight: 10 },

{ keywords: ['reviewcode', 'patch', 'jsdoc', 'performance'], file: 'tools\_code\_review.gs', weight: 9 }, // جديد لمراجعة الكود

{ keywords: ['project', 'analyze', 'insights', 'files', 'scan'], file: 'tools\_project\_insights.gs', weight: 10 } // جديد لتحليلات المشروع

],

// تعريفات الأدوات (Function Declarations) لـ Gemini API

// هذه يتم جلبها ديناميكيًا من TOOLS\_CATALOG في GAssistant.AI.Config

TOOLS\_DEFINITIONS: []

};

})();

// كائن لتخزين الإعدادات النشطة بعد قراءتها من الخصائص أو الافتراضيات

let \_activeSettings = null;

/\*\*

\* @function \_readSettingsFromProperties

\* @description

\* تقرأ الإعدادات من ScriptProperties و UserProperties وتدمجها مع الإعدادات الافتراضية.

\* يتم تخزينها مؤقتًا في الكاش.

\* @returns {object} - كائن الإعدادات المدمجة.

\*/

function \_readSettingsFromProperties() {

const scriptProps = PropertiesService.getScriptProperties().getProperties();

const userProps = PropertiesService.getUserProperties().getProperties();

const settings = JSON.parse(JSON.stringify(\_DEFAULT\_SETTINGS)); // نسخة عميقة من الافتراضيات

// دمج الخصائص من ScriptProperties (أولوية أعلى من الافتراضيات)

for (const key in scriptProps) {

if (settings.hasOwnProperty(key)) {

try {

settings[key] = JSON.parse(scriptProps[key]);

} catch (e) {

settings[key] = scriptProps[key];

}

}

}

// دمج الخصائص من UserProperties (أولوية أعلى من ScriptProperties)

for (const key in userProps) {

if (settings.hasOwnProperty(key)) {

try {

settings[key] = JSON.parse(userProps[key]);

} catch (e) {

settings[key] = userProps[key];

}

}

}

// جلب تعريفات الأدوات من TOOLS\_CATALOG وتعيينها

// يجب أن يكون TOOLS\_CATALOG متاحًا هنا (تم حقنه كوحدة عامة)

if (typeof TOOLS\_CATALOG !== 'undefined' && typeof TOOLS\_CATALOG.getFunctionList === 'function') {

settings.TOOLS\_DEFINITIONS = TOOLS\_CATALOG.getFunctionList(true); // جلب جميع الأدوات بما في ذلك الداخلية

} else {

console.warn("TOOLS\_CATALOG is not available or getFunctionList is not a function. Function calling might be limited.");

settings.TOOLS\_DEFINITIONS = [];

}

// تحديد دور المستخدم الحالي (يمكن أن يكون ديناميكيًا بناءً على البريد الإلكتروني أو غيره)

// حاليًا، يتم جلبها من AI\_CONTEXT، ولكن يمكن أن تكون جزءًا من هذا الملف إذا كانت ثابتة.

// settings.currentUserRole = GAssistant.AI.Context.getUserRole(); // مثال

return settings;

}

/\*\*

\* @function \_getAndCacheActiveSettings

\* @description

\* تجلب الإعدادات النشطة من الكاش أو تقرأها من الخصائص وتخزنها في الكاش.

\* @returns {object} - كائن الإعدادات النشطة.

\*/

function \_getAndCacheActiveSettings() {

if (\_activeSettings) {

return \_activeSettings; // إرجاع الإعدادات المخزنة في الذاكرة

}

let cachedSettings = CACHE.get('activeSettings');

if (cachedSettings) {

\_activeSettings = JSON.parse(cachedSettings);

return \_activeSettings;

}

const settings = \_readSettingsFromProperties();

\_activeSettings = settings; // تخزين في الذاكرة

CACHE.put('activeSettings', JSON.stringify(settings), CACHE\_DURATION\_SECONDS);

return settings;

}

/\*\*

\* @function refreshSettingsCache

\* @description

\* تُجبر النظام على إعادة قراءة الإعدادات من الخصائص وتحديث الكاش.

\*/

function refreshSettingsCache() {

\_activeSettings = null; // مسح الإعدادات من الذاكرة

CACHE.remove('activeSettings'); // مسح الإعدادات من الكاش

\_getAndCacheActiveSettings(); // إعادة قراءة وتخزين

console.log("Settings cache refreshed.");

}

/\*\*

\* @function getSetting

\* @description

\* تجلب قيمة إعداد معين.

\* @param {string} key - مفتاح الإعداد.

\* @returns {any} - قيمة الإعداد.

\*/

function getSetting(key) {

const settings = \_getAndCacheActiveSettings();

if (!settings.hasOwnProperty(key)) {

if (settings.DEBUG\_MODE) {

console.warn(`GAssistant.Config: Setting "${key}" not found. Check \_DEFAULT\_SETTINGS.`);

}

return undefined;

}

return settings[key];

}

/\*\*

\* @function getSecret

\* @description

\* تجلب مفتاح API سري من ScriptProperties.

\* @param {string} keyName - اسم المفتاح في ScriptProperties (مثال: "GEMINI\_API\_KEY").

\* @returns {string|null} - قيمة المفتاح أو null إذا لم يتم العثور عليه.

\*/

function getSecret(keyName) {

const secret = PropertiesService.getScriptProperties().getProperty(keyName);

if (!secret && \_getAndCacheActiveSettings().DEBUG\_MODE) {

console.warn(`GAssistant.Config: Secret "${keyName}" is missing in Script Properties.`);

}

return secret;

}

/\*\*

\* @function getAllowedModels

\* @description

\* تُرجع قائمة بالنماذج المسموح بها.

\* @returns {string[]} - مصفوفة من أسماء النماذج.

\*/

function getAllowedModels() {

return \_getAndCacheActiveSettings().ALLOWED\_MODELS;

}

/\*\*

\* @function getDefaultModel

\* @description

\* تُرجع اسم النموذج الافتراضي.

\* @returns {string} - اسم النموذج الافتراضي.

\*/

function getDefaultModel() {

return \_getAndCacheActiveSettings().DEFAULT\_MODEL;

}

/\*\*

\* @function getModelForSimpleTasks

\* @description

\* تُرجع اسم النموذج المخصص للمهام البسيطة.

\* @returns {string} - اسم النموذج.

\*/

function getModelForSimpleTasks() {

return \_getAndCacheActiveSettings().MODEL\_FOR\_SIMPLE\_TASKS;

}

/\*\*

\* @function getApiEndpoint

\* @description

\* تُرجع نقطة نهاية Gemini API.

\* @returns {string} - نقطة نهاية API.

\*/

function getApiEndpoint() {

return \_getAndCacheActiveSettings().API\_ENDPOINT;

}

/\*\*

\* @function getGenerationConfig

\* @description

\* تُرجع إعدادات توليد النصوص.

\* @returns {object} - كائن إعدادات التوليد.

\*/

function getGenerationConfig() {

return \_getAndCacheActiveSettings().GENERATION\_CONFIG;

}

/\*\*

\* @function getSafetySettings

\* @description

\* تُرجع إعدادات الأمان.

\* @returns {Array<object>} - مصفوفة إعدادات الأمان.

\*/

function getSafetySettings() {

return \_getAndCacheActiveSettings().SAFETY\_SETTINGS;

}

/\*\*

\* @function getToolConfig

\* @description

\* تُرجع إعدادات الأدوات (مثل thinkingConfig).

\* @returns {object} - كائن إعدادات الأدوات.

\*/

function getToolConfig() {

return \_getAndCacheActiveSettings().TOOL\_CONFIG;

}

/\*\*

\* @function getToolsDefinitions

\* @description

\* تُرجع تعريفات الأدوات (Function Declarations) لـ Gemini API.

\* يتم جلبها من TOOLS\_CATALOG.

\* @returns {Array<object>} - مصفوفة تعريفات الأدوات.

\*/

function getToolsDefinitions() {

return \_getAndCacheActiveSettings().TOOLS\_DEFINITIONS;

}

/\*\*

\* @function getPrimaryHeaderColor

\* @description

\* تُرجع لون الرأس الأساسي للجداول.

\* @returns {string} - رمز اللون.

\*/

function getPrimaryHeaderColor() {

return \_getAndCacheActiveSettings().PRIMARY\_HEADER\_COLOR;

}

/\*\*

\* @function getCurrencyCode

\* @description

\* تُرجع رمز العملة الافتراضي.

\* @returns {string} - رمز العملة.

\*/

function getCurrencyCode() {

return \_getAndCacheActiveSettings().DEFAULT\_CURRENCY;

}

/\*\*

\* @function getStagingSheetName

\* @description

\* تُرجع اسم ورقة ورشة عمل المطور.

\* @returns {string} - اسم الورقة.

\*/

function getStagingSheetName() {

return \_getAndCacheActiveSettings().STAGING\_SHEET\_NAME;

}

/\*\*

\* @function getToolFileSuggestionPatterns

\* @description

\* تُرجع أنماط اقتراح الملفات لورشة العمل.

\* @returns {Array<object>} - مصفوفة الأنماط.

\*/

function getToolFileSuggestionPatterns() {

return \_getAndCacheActiveSettings().TOOL\_FILE\_SUGGESTION\_PATTERNS;

}

// --- تصدير الواجهة العامة للوحدة ---

return {

getSettings: \_getAndCacheActiveSettings, // تصدير الدالة الرئيسية لجلب جميع الإعدادات

refreshSettingsCache: refreshSettingsCache,

getSetting: getSetting,

getSecret: getSecret,

getAllowedModels: getAllowedModels,

getDefaultModel: getDefaultModel,

getModelForSimpleTasks: getModelForSimpleTasks,

getApiEndpoint: getApiEndpoint,

getGenerationConfig: getGenerationConfig,

getSafetySettings: getSafetySettings,

getToolConfig: getToolConfig,

getToolsDefinitions: getToolsDefinitions,

getPrimaryHeaderColor: getPrimaryHeaderColor,

getCurrencyCode: getCurrencyCode,

getStagingSheetName: getStagingSheetName,

getToolFileSuggestionPatterns: getToolFileSuggestionPatterns

};

})();