/\*\*

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

\* @file ai\_config.gs – v1.1

\* @module GAssistant.AI.Config

\* @description

\* يحتوي على كل الإعدادات والثوابت الخاصة بسلوك محرك الذكاء الاصطناعي.

\* يعتمد على الثوابت الأساسية من GAssistant.Config ويوفر إعدادات متقدمة للـ API.

\* يعتمد على: GAssistant.Config

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

\*/

var GAssistant = GAssistant || {};

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

// حقن التبعيات: يجب تمرير الوحدات المعتمدة هنا

GAssistant.AI.Config = (function(GA\_Config) {

'use strict';

/\*\*

\* @const {object} \_CONFIG

\* @description

\* كائن الإعدادات المركزي الخاص بوحدة الذكاء الاصطناعي.

\* يتم جلب بعض القيم من GAssistant.Config لضمان التوحيد.

\*/

const \_CONFIG = {

// ✨ تطوير: تم ربطها بالمتغيرات المركزية من GAssistant.Config

ALLOWED\_MODELS: GA\_Config.getAllowedModels(), // جلب النماذج المسموح بها من GAssistant.Config

DEFAULT\_MODEL: GA\_Config.getDefaultModel(), // جلب النموذج الافتراضي من GAssistant.Config

// ✨ تطوير: النموذج السريع للأوامر البسيطة

MODEL\_FOR\_SIMPLE\_TASKS: GA\_Config.getSetting('MODEL\_FOR\_SIMPLE\_TASKS') || "gemini-1.5-flash", // افتراضي

// 📏 إعداد جديد: الحد الأقصى لطول الـ prompt بالتوكنز (تقديريًا)

// هذا يعتمد على النموذج المستخدم، 32768 هو حد Gemini 1.5 Pro

MAX\_PROMPT\_LENGTH: GA\_Config.getSetting('MAX\_PROMPT\_LENGTH') || 30000, // ترك بعض المساحة للأمان

// 🧠 إعداد جديد: الحد الأقصى لطول تعليمات النظام قبل التقليص بواسطة ContextPacker

MAX\_SYSTEM\_INSTRUCTION\_LENGTH: GA\_Config.getSetting('MAX\_SYSTEM\_INSTRUCTION\_LENGTH') || 4000,

// إعدادات الـ API (يمكن جلبها من GAssistant.Config أو تعريفها هنا)

API\_ENDPOINT: GA\_Config.getSetting('API\_ENDPOINT') || "https://generativelanguage.googleapis.com/v1beta/models/",

GENERATION\_CONFIG: GA\_Config.getSetting('GENERATION\_CONFIG') || {

temperature: 0.7,

topP: 0.95,

topK: 40,

maxOutputTokens: 1024,

},

SAFETY\_SETTINGS: GA\_Config.getSetting('SAFETY\_SETTINGS') || [

{ category: "HARM\_CATEGORY\_HARASSMENT", threshold: "BLOCK\_MEDIUM\_AND\_ABOVE" },

{ category: "HARM\_CATEGORY\_HATE\_SPEECH", threshold: "BLOCK\_MEDIUM\_AND\_ABOVE" },

{ category: "HARM\_CATEGORY\_SEXUALLY\_EXPLICIT", threshold: "BLOCK\_MEDIUM\_AND\_ABOVE" },

{ category: "HARM\_CATEGORY\_DANGEROUS\_CONTENT", threshold: "BLOCK\_MEDIUM\_AND\_ABOVE" },

],

// أسماء مفاتيح الـ API في UserProperties/ScriptProperties

API\_KEYS\_NAMES: GA\_Config.getSetting('API\_KEYS\_NAMES') || {

GEMINI: 'GEMINI\_API\_KEY'

}

};

/\*\*

\* @function \_getApiKey

\* @description

\* تجلب مفتاح الـ API من UserProperties أو ScriptProperties.

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

\* @returns {string} - مفتاح الـ API.

\* @throws {Error} إذا لم يتم العثور على المفتاح.

\*/

function \_getApiKey(keyName) {

const userProps = PropertiesService.getUserProperties();

const scriptProps = PropertiesService.getScriptProperties();

let apiKey = userProps.getProperty(keyName) || scriptProps.getProperty(keyName);

if (!apiKey) {

// ✅ تحسين: استخدام GAssistant.Helpers.logOperation و GAssistant.AI.Dialogue.showError

// يجب أن تكون GAssistant.Helpers و GAssistant.AI.Dialogue محقونة هنا أيضًا إذا أردنا استخدامها مباشرةً.

// لغرض هذا الملف، سنكتفي بالرمي بالخطأ.

throw new Error(`لم يتم العثور على مفتاح API لـ "${keyName}". يرجى إعداده في خصائص المشروع أو المستخدم.`);

}

return apiKey;

}

/\*\*

\* @function getConfig

\* @description

\* ترجع نسخة عميقة من كائن الإعدادات لمنع التعديل المباشر.

\* @returns {object} - كائن الإعدادات.

\*/

function getConfig() {

// استخدام JSON.stringify لإنشاء نسخة عميقة وآمنة

return JSON.parse(JSON.stringify(\_CONFIG));

}

/\*\*

\* @function getModelForTask

\* @description

\* يحدد النموذج المناسب للمهمة بناءً على نوع المهمة (بسيطة أو معقدة).

\* @param {boolean} isSimpleTask - صحيح إذا كانت المهمة بسيطة، خطأ إذا كانت معقدة.

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

\*/

function getModelForTask(isSimpleTask) {

return isSimpleTask ? \_CONFIG.MODEL\_FOR\_SIMPLE\_TASKS : \_CONFIG.DEFAULT\_MODEL;

}

/\*\*

\* @function getPreferredGeminiModel

\* @description

\* يحدد النموذج المفضل بناءً على نص الـ prompt ونوع المهمة.

\* يمكنه التبديل بين النماذج (مثل Gemini-1.5-pro-vision للصور، Gemini-1.5-pro للكود).

\* @param {string} promptText - نص الـ prompt من المستخدم.

\* @param {string} [modelOverride] - اسم نموذج محدد لتجاوز الاختيار التلقائي.

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

\*/

function getPreferredGeminiModel(promptText, modelOverride) {

if (modelOverride && \_CONFIG.ALLOWED\_MODELS.includes(modelOverride)) {

return modelOverride;

}

const lowerPrompt = promptText.toLowerCase();

// 👁️ نموذج للصور (إذا كان هناك اكتشاف لصور في الـ prompt)

// حاليًا، هذا يعتمد على وجود imageBlob في ask()، وليس على تحليل النص.

// يمكن إضافة منطق هنا لاحقًا لتحليل نص الـ prompt لاكتشاف نية بصرية.

// if (lowerPrompt.includes('صورة') || lowerPrompt.includes('تحليل بصري')) {

// return 'gemini-1.5-pro-vision';

// }

// 💻 نموذج للكود البرمجي

const isCode = /(كود|برمجة|سكريبت|دالة|جافاسكريبت|بايثون|html|css|javascript|python)/.test(lowerPrompt);

if (isCode) {

return 'gemini-1.5-pro'; // نموذج أقوى للكود

}

// 💬 نموذج للمحادثات العامة أو المهام البسيطة

const isChat = /(محادثة|سؤال|ماذا|كيف|لماذا|شرح|فكرة|مساعدة|تغذية راجعة)/.test(lowerPrompt);

if (isChat) {

return \_CONFIG.DEFAULT\_MODEL;

}

// 🧩 في حال لم يتم التعرف على النمط، العودة للنموذج الافتراضي للمهام البسيطة

return \_CONFIG.MODEL\_FOR\_SIMPLE\_TASKS;

}

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

return {

getConfig,

getApiKey: \_getApiKey, // تصدير دالة جلب المفتاح السري

getModelForTask,

getPreferredGeminiModel, // تصدير الدالة الجديدة

// يمكن تصدير إعدادات محددة مباشرة إذا لزم الأمر

getApiEndpoint: () => \_CONFIG.API\_ENDPOINT,

getGenerationConfig: () => \_CONFIG.GENERATION\_CONFIG,

getSafetySettings: () => \_CONFIG.SAFETY\_SETTINGS,

getMaxPromptLength: () => \_CONFIG.MAX\_PROMPT\_LENGTH, // ✅ تصدير إعداد جديد

getMaxSystemInstructionLength: () => \_CONFIG.MAX\_SYSTEM\_INSTRUCTION\_LENGTH // ✅ تصدير إعداد جديد

};

})(

GAssistant.Config // تم حقن GAssistant.Config هنا

);