/\*\*

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

\* @file ai\_core.gs – v1.2

\* @module GAssistant.AI.Core

\* @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, GAssistant.Helpers, GAssistant.AI.Dispatcher, GAssistant.AI.Memory, GAssistant.AI.Dialogue, GAssistant.AI.Context, GAssistant.AI.Ask, GAssistant.Tools.Catalog

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

\*/

var GAssistant = GAssistant || {};

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

// حقن التبعيات: يجب تمرير الوحدات المعتمدة هنا

GAssistant.AI.Core = (function(GA\_Config, GA\_Helpers, GA\_AI\_Dispatcher, GA\_AI\_Memory, GA\_AI\_Dialogue, GA\_AI\_Context, GA\_AI\_Ask, GA\_Tools\_Catalog) {

'use strict';

/\*\*

\* @function ask

\* @description

\* الدالة الرئيسية للتفاعل مع نموذج Gemini.

\* تقوم ببناء الطلب، استدعاء API، ومعالجة الرد.

\* @param {string} userPrompt - نص طلب المستخدم.

\* @param {string} [modelOverride] - اسم النموذج المراد استخدامه، يتجاوز الاختيار التلقائي.

\* @param {string} [imageBlob] - بيانات الصورة (Blob) إذا كان الطلب يتضمن صورة.

\* @returns {object} - كائن الرد النهائي من النموذج أو الأداة المنفذة.

\*/

function ask(userPrompt, modelOverride, imageBlob) {

GA\_Helpers.logOperation('GAssistant.AI.Core', 'ask:start', { prompt: userPrompt.substring(0, Math.min(userPrompt.length, 50)), modelOverride, hasImage: !!imageBlob });

let state = {

userPrompt: userPrompt,

model: null,

systemInstruction: "",

history: [],

tools: [],

images: [],

final: null, // النتيجة النهائية التي ستُعرض للمستخدم

halt: false // إذا كان true، تتوقف دورة المعالجة

};

try {

// 1. معالجة النية المسبقة (أوامر فورية)

state = GA\_AI\_Dispatcher.preProcessIntent(userPrompt, state);

if (state.halt) {

GA\_Helpers.logOperation('GAssistant.AI.Core', 'ask:pre\_processed\_halt', { finalType: state.final ? state.final.type : 'none' });

return state.final;

}

// 2. تحديد النموذج المناسب

state.model = GA\_Config.getPreferredGeminiModel(userPrompt, modelOverride).model; // الحصول على اسم النموذج فقط

GA\_Helpers.logOperation('GAssistant.AI.Core', 'ask:model\_selected', { model: state.model });

// 3. إثراء السياق (System Instruction)

GA\_AI\_Context.enrichContext(state);

if (state.halt) {

GA\_Helpers.logOperation('GAssistant.AI.Core', 'ask:context\_enrichment\_halt', { finalType: state.final ? state.final.type : 'none' });

return state.final;

}

GA\_Helpers.logOperation('GAssistant.AI.Core', 'ask:context\_enriched', { systemInstructionLength: state.systemInstruction.length });

// 4. إضافة سجل المحادثة السابق

state.history = GA\_AI\_Memory.getSessionHistory();

GA\_Helpers.logOperation('GAssistant.AI.Core', 'ask:history\_loaded', { historyLength: state.history.length });

// 5. إضافة الصورة إذا وجدت

if (imageBlob) {

state.images.push(imageBlob);

GA\_Helpers.logOperation('GAssistant.AI.Core', 'ask:image\_added', {});

}

// 6. تهيئة الأدوات

state.tools = GA\_Tools\_Catalog.getFunctionList();

GA\_Helpers.logOperation('GAssistant.AI.Core', 'ask:tools\_loaded', { toolCount: state.tools.length });

// 7. استدعاء AI\_ASK لإرسال الطلب ومعالجة الردود

state = GA\_AI\_Ask.callGeminiAgent(state);

GA\_Helpers.logOperation('GAssistant.AI.Core', 'ask:gemini\_agent\_called', { halt: state.halt, finalType: state.final ? state.final.type : 'none' });

// 8. حفظ آخر طلب للذاكرة (لإعادة التنفيذ)

GA\_AI\_Memory.setLastPrompt(userPrompt, state.model);

GA\_Helpers.logOperation('GAssistant.AI.Core', 'ask:end', { finalType: state.final ? state.final.type : 'none' });

return state.final;

} catch (e) {

GA\_Helpers.logOperation('GAssistant.AI.Core', 'ask:exception', { error: e.message, stack: e.stack });

return GA\_AI\_Dialogue.showError(`💥 حدث خطأ غير متوقع في محرك الذكاء الاصطناعي: ${e.message}`);

}

}

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

return {

ask

};

})(

GAssistant.Config,

GAssistant.Helpers,

GAssistant.AI.Dispatcher,

GAssistant.AI.Memory,

GAssistant.AI.Dialogue,

GAssistant.AI.Context, // حقن AI\_CONTEXT

GAssistant.AI.Ask, // حقن AI\_ASK

GAssistant.Tools.Catalog // حقن TOOLS\_CATALOG

);