/\*\*

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

\* @file 99\_Code.gs

\* @description

\* نقطة الدخول العامة للطلبات من الواجهة الأمامية (google.script.run) والقوائم.

\* يعمل كـ "Router" نظيف يوجه الطلبات إلى وحدات GAssistant المناسبة.

\* هذا الملف يجب أن يظل بسيطًا وخاليًا من منطق الأعمال.

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

\*/

// ============================================================================

// UI Functions (called from Menus & Sidebars)

// ============================================================================

/\*\*

\* دالة تُستدعى عند فتح المستند لإنشاء القوائم المخصصة.

\*/

function onOpen() {

GAssistant.UI.onOpen();

}

/\*\*

\* تعرض الشريط الجانبي الرئيسي للمساعد.

\*/

function showAssistantSidebar() {

GAssistant.UI.showAssistantSidebar();

}

/\*\*

\* تعرض الشريط الجانبي لأدوات المطورين.

\*/

function showDeveloperSidebar() {

GAssistant.UI.showDeveloperSidebar();

}

/\*\*

\* تجلب الكود من الخلية النشطة.

\* @returns {string} - محتوى الخلية النشطة.

\*/

function getCodeFromActiveCell() {

return GAssistant.UI.getCodeFromActiveCell();

}

/\*\*

\* تحدث الخلية النشطة بالكود الجديد.

\* @param {string} code - الكود المراد كتابته.

\* @returns {object} - رسالة نجاح أو خطأ.

\*/

function updateActiveCellWithCode(code) {

return GAssistant.UI.updateActiveCellWithCode(code);

}

// ============================================================================

// AI Core Function

// ============================================================================

/\*\*

\* الدالة الأساسية لإرسال طلب إلى نموذج Gemini.

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

\* @param {object} [options] - خيارات إضافية.

\* @returns {object} - كائن الرد النهائي للواجهة الأمامية.

\*/

function askGemini(userPrompt, options) {

return GAssistant.AI.Core.ask(userPrompt, options);

}

// ============================================================================

// Tool Functions (exposed to UI and Menus)

// ============================================================================

// --- Sheet Tools ---

function summarizeActiveSheetWithGemini() {

return GAssistant.Tools.Sheets.summarizeActiveSheetWithGemini();

}

function suggestFormulaWithGemini(description) {

return GAssistant.Tools.Sheets.suggestFormulaWithGemini(description);

}

// --- Developer Tools ---

function reviewCode(code, reviewType) {

return GAssistant.Tools.CodeReview.reviewCode({ code, reviewType });

}

function getBuiltinFunctionDoc(functionName) {

return GAssistant.Tools.Developer.getBuiltinFunctionDoc({ functionName });

}

function suggestTargetFile(functionCode) {

return GAssistant.Tools.Developer.suggestTargetFile(functionCode);

}

function stageCodeInWorkshop(functionCode, suggestedFile) {

return GAssistant.Tools.Developer.stageCodeInWorkshop(functionCode, suggestedFile);

}

function analyzeProject(userQuery) {

return GAssistant.Tools.ProjectInsights.analyzeProject(userQuery);

}

function exportToolsDocumentationToDoc() {

GAssistant.Tools.Developer.exportToolsDocumentationToDoc();

}

// ============================================================================

// System & Testing Functions (exposed to Menus)

// ============================================================================

/\*\*

\* يمسح ذاكرة الجلسة الحالية للمساعد.

\* هذه الدالة هي الوسيط الذي تستدعيه القائمة المخصصة.

\*/

function clearSessionContext() {

GAssistant.AI.Memory.clearSessionContext();

// عرض رسالة تأكيد للمستخدم بعد المسح

SpreadsheetApp.getUi().alert('تم مسح ذاكرة الجلسة بنجاح.');

}

/\*\*

\* يشغل جميع الاختبارات المحددة في المشروع.

\* @returns {object} - كائن يحتوي على ملخص ونتائج الاختبارات.

\*/

function runAllTests() {

const results = GAssistant.Tests.runAllTests();

// يمكن عرض النتائج هنا في alert أو sidebar

SpreadsheetApp.getUi().alert('اكتمل تشغيل الاختبارات. انظر السجلات لمزيد من التفاصيل.');

return results;

}