/\*\*

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

\* @file tools\_sheets.gs

\* @module GAssistant.Tools.Sheets

\* @description

\* أدوات متقدمة للتعامل مع Google Sheets باستخدام الذكاء الاصطناعي.

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

\*/

var GAssistant = GAssistant || {};

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

GAssistant.Tools.Sheets = (() => {

'use strict';

// --- حقن التبعيات (النسخة الآمنة) ---

const Utils = GAssistant.Utils;

const AI = GAssistant.AI.Core;

const Config = GAssistant.Config;

const Context = GAssistant.AI.Context;

// سيتم استدعاء GAssistant.UI.Dialogue مباشرة عند الحاجة.

/\*\*

\* يستخدم Gemini AI لتلخيص محتوى الورقة النشطة.

\* @returns {object} - كائن استجابة موحد.

\*/

function summarizeActiveSheetWithGemini() {

return Utils.executeSafely(() => {

// بناء سياق مؤقت فقط للحصول على معلومات الورقة

const tempState = {};

Context.build(tempState);

const sheetContext = tempState.systemInstruction;

const prompt = `بناءً على سياق ورقة العمل التالي، قدم ملخصًا موجزًا وشاملاً للبيانات، الغرض المحتمل منها، وأي رؤى أولية يمكنك استنتاجها. استخدم تنسيق Markdown:\n\n${sheetContext}\n\nملخص الورقة:`;

const result = AI.ask(prompt, { modelOverride: Config.get('GEMINI\_FLASH\_MODEL') });

Utils.log('Tools.Sheets.summarizeActiveSheetWithGemini: AI summary received.');

return result;

}, [], 'summarizing active sheet');

}

/\*\*

\* يستخدم Gemini AI لاقتراح صيغة Google Sheets.

\* @param {object} args - { description: string }.

\* @returns {object} - كائن استجابة موحد.

\*/

function suggestFormulaWithGemini(args) {

const { description } = args;

return Utils.executeSafely(() => {

const tempState = {};

Context.build(tempState);

const sheetContext = tempState.systemInstruction;

const prompt = `

بناءً على سياق ورقة Google Sheets التالية:

${sheetContext}

قم بتوليد صيغة Google Sheets دقيقة تحقق الوصف التالي: "${description}".

أعد الصيغة فقط، بدون أي شرح أو نص إضافي. يجب أن تبدأ الصيغة بعلامة =.`;

const result = AI.ask(prompt, { modelOverride: Config.get('GEMINI\_FLASH\_MODEL') });

Utils.log('Tools.Sheets.suggestFormulaWithGemini: AI formula suggestion received.');

// الوصول الآمن لوحدة Dialogue

const Dialogue = GAssistant.UI.Dialogue;

// التحقق من أن الرد هو صيغة صالحة

if (result.type === 'info' && result.text.trim().startsWith('=')) {

return Dialogue.createSuccess(`✅ الصيغة المقترحة: \n\`\`\`\n${result.text.trim()}\n\`\`\``);

} else {

Utils.warn('Tools.Sheets.suggestFormulaWithGemini: AI returned non-formula text.', { response: result.text });

return Dialogue.createWarning(`لم يتمكن المساعد من توليد صيغة صالحة. الرد كان: ${result.text}`);

}

}, [], `suggesting formula for: ${description}`);

}

/\*\*

\* ينشئ ورقة عمل جديدة بناءً على قالب محدد من المستخدم.

\* @param {object} args - { templateName: string, newSheetName: string }.

\* @returns {object} - كائن استجابة موحد.

\*/

function createSheetFromTemplate(args) {

const { templateName, newSheetName } = args;

return Utils.executeSafely(() => {

// الوصول الآمن لوحدة Dialogue

const Dialogue = GAssistant.UI.Dialogue;

const templates = {

"المالية الشهرية": ["التاريخ", "الوصف", "الإيراد", "المصروف", "الرصيد"],

"قائمة المهام": ["المهمة", "الحالة", "الأولوية", "تاريخ الاستحقاق", "المسؤول"],

"إدارة المخزون": ["معرف المنتج", "اسم المنتج", "الكمية الحالية", "نقطة إعادة الطلب", "المورد"]

};

const headers = templates[templateName];

if (!headers) {

return Dialogue.createError(`قالب "${templateName}" غير موجود. القوالب المتاحة: ${Object.keys(templates).join(', ')}.`);

}

const sheet = Utils.getSheet(newSheetName, headers);

if (sheet) {

Utils.log('Tools.Sheets.createSheetFromTemplate: Success', { newSheetName, templateName });

return Dialogue.createSuccess(`✅ تم إنشاء ورقة "${newSheetName}" بنجاح باستخدام قالب "${templateName}".`);

} else {

// getSheet سوف يسجل الخطأ بالفعل

return Dialogue.createError(`فشل إنشاء ورقة "${newSheetName}".`);

}

}, [], `creating sheet '${newSheetName}' from template '${templateName}'`);

}

return {

summarizeActiveSheetWithGemini,

suggestFormulaWithGemini,

createSheetFromTemplate

};

})();