/\*\*

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

\* @file Code.gs

\* @description

\* هذا الملف يحتوي على الدوال التي يمكن استدعاؤها مباشرة من الواجهة الأمامية (Sidebar.html)

\* عبر google.script.run.

\* يعتمد على الوحدات الأخرى في المشروع للقيام بالمهام المعقدة.

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

\*/

// تأكد من أن هذه الوحدات معرفة كـ 'var' في ملفاتها الخاصة

// وأنها تم تحميلها قبل هذا الملف في مشروع Apps Script.

// إذا كان هذا الملف يستخدم نمط IIFE، يجب تمريرها كمعاملات.

// مثال على حقن التبعيات إذا كان Code.gs يستخدم نمط IIFE

// var Code = (function(AI\_CONFIG, AI\_CORE, AI\_TOOLS\_DIALOGUE, HELPERS, TOOLS\_INTERNAL, TOOLS\_SHEETS) {

// 'use strict';

/\*\*

\* @function getAvailableModels

\* @description

\* تُرجع قائمة بالنماذج المتاحة للواجهة الأمامية من إعدادات AI\_CONFIG.

\* هذه الدالة يتم استدعاؤها من Sidebar.html لملء القائمة المنسدلة للنماذج.

\* @returns {object} - كائن يحتوي على نوع الرد وقائمة النماذج.

\*/

function getAvailableModels() {

// يجب التأكد من أن AI\_CONFIG ووحدة HELPERS معرفة ومتاحة هنا.

try {

const config = AI\_CONFIG.getConfig(); // جلب الإعدادات من AI\_CONFIG

HELPERS.logOperation('Code.gs', 'getAvailableModels:success', { models: config.ALLOWED\_MODELS });

return { type: 'model\_list', models: config.ALLOWED\_MODELS };

} catch (e) {

HELPERS.logOperation('Code.gs', 'getAvailableModels:error', { error: e.message });

// في حالة الخطأ، يمكن إرجاع قائمة فارغة أو رسالة خطأ مناسبة للواجهة الأمامية

return { type: 'error', content: `فشل جلب النماذج المتاحة: ${e.message}` };

}

}

/\*\*

\* @function askGemini

\* @description

\* نقطة الدخول الرئيسية لاستدعاء المساعد الذكي من الواجهة الأمامية.

\* تقوم بتمرير طلب المستخدم والنموذج المختار إلى AI\_CORE.

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

\* @param {string} modelName - اسم النموذج المختار.

\* @returns {object} - الرد النهائي من AI\_CORE.

\*/

function askGemini(userPrompt, modelName) {

// يجب التأكد من أن AI\_CORE معرفة ومتاحة هنا.

HELPERS.logOperation('Code.gs', 'askGemini:start', { prompt: userPrompt, model: modelName });

const response = AI\_CORE.ask(userPrompt, modelName);

HELPERS.logOperation('Code.gs', 'askGemini:end', { responseType: response.type });

return response;

}

/\*\*

\* @function executeConfirmedAction

\* @description

\* تُنفذ الإجراءات التي تتطلب تأكيد المستخدم. يتم استدعاؤها بعد موافقة المستخدم في الواجهة.

\* @param {string} action - اسم الإجراء المراد تنفيذه (مثلاً '\_internal\_userConfirmed').

\* @param {object} params - المعاملات الخاصة بالإجراء.

\* @returns {object} - نتيجة تنفيذ الإجراء.

\*/

function executeConfirmedAction(action, params) {

// يجب التأكد من أن TOOLS\_INTERNAL معرفة ومتاحة هنا.

HELPERS.logOperation('Code.gs', 'executeConfirmedAction:start', { action, params });

// نفترض أن TOOLS\_INTERNAL.executeInternalAction هي الدالة التي تعالج هذه الإجراءات

// ستحتاج إلى إنشاء ملف TOOLS\_INTERNAL.gs إذا لم يكن موجودًا.

const result = TOOLS\_INTERNAL.executeInternalAction(action, params);

HELPERS.logOperation('Code.gs', 'executeConfirmedAction:end', { resultType: result.type });

return result;

}

/\*\*

\* @function getSheetInsightForPrompt

\* @description

\* تُولد prompt ذكي بناءً على تحليل ورقة العمل النشطة.

\* يتم استدعاؤها من تبويب "توليد Prompts".

\* @returns {string} - نص الـ prompt المقترح.

\*/

function getSheetInsightForPrompt() {

// يجب التأكد من أن TOOLS\_SHEETS معرفة ومتاحة هنا.

HELPERS.logOperation('Code.gs', 'getSheetInsightForPrompt:start', {});

const insight = TOOLS\_SHEETS.generateSheetAnalysisPrompt(); // نفترض هذه الدالة موجودة في TOOLS\_SHEETS.gs

HELPERS.logOperation('Code.gs', 'getSheetInsightForPrompt:end', { insightLength: insight.length });

return insight;

}

// الدوال العائمة (Floating Action Buttons)

// هذه الدوال يمكن أن تستدعي وظائف من TOOLS\_SHEETS أو وحدات أخرى

function createMinimalTestSheet() {

HELPERS.logOperation('Code.gs', 'createMinimalTestSheet:start', {});

const result = TOOLS\_SHEETS.createMinimalTestSheet(); // نفترض هذه الدالة موجودة

HELPERS.logOperation('Code.gs', 'createMinimalTestSheet:end', { resultType: result.type });

return result;

}

function summarizeActiveSheet() {

HELPERS.logOperation('Code.gs', 'summarizeActiveSheet:start', {});

const result = TOOLS\_SHEETS.summarizeActiveSheet(); // نفترض هذه الدالة موجودة

HELPERS.logOperation('Code.gs', 'summarizeActiveSheet:end', { resultType: result.type });

return result;

}

function suggestTargetFile() {

HELPERS.logOperation('Code.gs', 'suggestTargetFile:start', {});

const result = TOOLS\_SHEETS.suggestTargetFile(); // نفترض هذه الدالة موجودة

HELPERS.logOperation('Code.gs', 'suggestTargetFile:end', { resultType: result.type });

return result;

}

// }) (AI\_CONFIG, AI\_CORE, AI\_TOOLS\_DIALOGUE, HELPERS, TOOLS\_INTERNAL, TOOLS\_SHEETS); // نهاية IIFE إذا كنت تستخدمها