/\*\*

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

\* @file tools\_catalog.gs

\* @module GAssistant.Tools.Catalog

\* @description

\* الكتالوج المركزي لجميع الأدوات المتاحة لنموذج Gemini.

\* هذا الملف هو المصدر الوحيد لـ Function Declarations.

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

\*/

var GAssistant = GAssistant || {};

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

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

'use strict';

const { log, error } = GAssistant.Utils;

/\*\*

\* @typedef {Object} ToolDefinition

\* @property {string} name - اسم الدالة (يجب أن يتطابق مع الدالة في Code.gs).

\* @property {string} description - وصف واضح للدالة ليفهمه النموذج.

\* @property {object} parameters - مخطط JSON للمعاملات (JSON Schema).

\* @property {string} functionPath - المسار الكامل للدالة داخل GAssistant (e.g., "GAssistant.Tools.Sheets.createSheet").

\*/

/\*\*

\* @const {ToolDefinition[]} ALL\_TOOL\_DEFINITIONS

\* @description

\* القائمة الكاملة لتعريفات الأدوات. هذه هي "قائمة المهارات" التي يمتلكها المساعد.

\*/

const ALL\_TOOL\_DEFINITIONS = [

// --- Sheets Tools ---

{

name: "summarizeActiveSheetWithGemini",

description: "ينشئ ملخصًا ذكيًا ومفصلاً للبيانات في ورقة العمل النشطة حاليًا.",

parameters: { type: "OBJECT", properties: {}, required: [] },

functionPath: "GAssistant.Tools.Sheets.summarizeActiveSheetWithGemini"

},

{

name: "suggestFormulaWithGemini",

description: "يقترح صيغة Google Sheets بناءً على وصف نصي للنتيجة المطلوبة.",

parameters: {

type: "OBJECT",

properties: {

description: { type: "STRING", description: "وصف واضح للصيغة المطلوبة. مثال: 'مجموع العمود C إذا كانت القيمة في العمود A هي 'مبيعات''" }

},

required: ["description"]

},

functionPath: "GAssistant.Tools.Sheets.suggestFormulaWithGemini"

},

// --- Accounting Tools ---

{

name: "calculateGrossProfit",

description: "يحسب الربح الإجمالي (الإيرادات - المصروفات) لفترة زمنية محددة.",

parameters: {

type: "OBJECT",

properties: {

startDate: { type: "STRING", description: "تاريخ البدء للحساب (صيغة YYYY-MM-DD)." },

endDate: { type: "STRING", description: "تاريخ الانتهاء للحساب (صيغة YYYY-MM-DD)." }

},

required: ["startDate", "endDate"]

},

functionPath: "GAssistant.Tools.Accounting.calculateGrossProfit"

},

{

name: "generateDailySummary",

description: "يولد ملخصًا ماليًا لليوم المحدد (أو اليوم الحالي) ويضيفه إلى ورقة الملخص اليومي.",

parameters: {

type: "OBJECT",

properties: {

date: { type: "STRING", description: "التاريخ المطلوب للملخص (صيغة YYYY-MM-DD). إذا لم يحدد، سيستخدم اليوم الحالي." }

},

required: []

},

functionPath: "GAssistant.Tools.Accounting.generateDailySummary"

},

// --- Developer Tools ---

{

name: "reviewCode",

description: "يقدم مراجعة شاملة لكود Google Apps Script. يحلل الأداء، قابلية القراءة، أفضل الممارسات، وقد يقترح تعديلات.",

parameters: {

type: "OBJECT",

properties: {

code: { type: "STRING", description: "الكود البرمجي الكامل المراد مراجعته." },

reviewType: { type: "STRING", "enum": ["general", "performance", "security", "readability"], description: "نوع المراجعة المطلوبة." }

},

required: ["code", "reviewType"]

},

functionPath: "GAssistant.Tools.CodeReview.reviewCode"

},

{

name: "getBuiltinFunctionDoc",

description: "يجلب توثيقًا مفصلاً لدالة Apps Script مدمجة، بما في ذلك الوصف، المعاملات، ومثال للاستخدام.",

parameters: {

type: "OBJECT",

properties: {

functionName: { type: "STRING", description: "اسم الدالة الكامل. مثال: 'SpreadsheetApp.getActiveSpreadsheet'" }

},

required: ["functionName"]

},

functionPath: "GAssistant.Tools.Developer.getBuiltinFunctionDoc"

}

];

/\*\*

\* تُرجع قائمة بتعريفات الدوال (Function Declarations) لـ Gemini API.

\* @returns {ToolDefinition[]}

\*/

function getDeclarations() {

log('Tools.Catalog.getDeclarations: Fetching tool declarations.');

// إزالة functionPath قبل إرسالها إلى Gemini

return ALL\_TOOL\_DEFINITIONS.map(({ functionPath, ...rest }) => rest);

}

/\*\*

\* تجلب الدالة الفعلية المرتبطة باسم الأداة.

\* @param {string} toolName - اسم الأداة.

\* @returns {Function|null} - الدالة الفعلية أو null.

\*/

function getFunction(toolName) {

const tool = ALL\_TOOL\_DEFINITIONS.find(t => t.name === toolName);

if (!tool) {

error(`Tools.Catalog.getFunction: Tool '${toolName}' not found.`);

return null;

}

const func = GAssistant.Utils.getFunctionByPath(tool.functionPath);

if (!func) {

error(`Tools.Catalog.getFunction: Function path '${tool.functionPath}' for tool '${toolName}' is invalid.`);

return null;

}

return func;

}

return {

getDeclarations,

getFunction

};

})();