/\*\*

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

\* @file tools\_catalog.gs – v16.0

\* @module TOOLS\_CATALOG

\* @description

\* يحتوي على تجميع الأدوات (functions) المستخدمة من قبل الذكاء الاصطناعي

\* ضمن تصنيفات منظمة، بالإضافة إلى قائمة موحدة AVAILABLE\_TOOLS

\* لسهولة الوصول والاستدعاء من قبل المحرك المركزي.

\* يوفر تعريفات الأدوات (Function Declarations) لـ Gemini API.

\* تمت إعادة هيكلة تعريفات الأدوات لتضمين التصنيف والوصف بشكل مباشر.

\* تعتمد على: AI\_TOOLS\_DIALOGUE, TOOLS\_SHEETS, TOOLS\_ACCOUNTING,

\* TOOLS\_GENERAL, TOOLS\_INTERNAL, TOOLS\_SYSTEM, HELPERS

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

\*/

var TOOLS\_CATALOG = (function(

AI\_TOOLS\_DIALOGUE,

TOOLS\_SHEETS, // افتراض وجود هذا الملف لوظائف الجداول

TOOLS\_ACCOUNTING, // افتراض وجود هذا الملف لوظائف المحاسبة

TOOLS\_GENERAL, // افتراض وجود هذا الملف لوظائف عامة

TOOLS\_INTERNAL, // افتراض وجود هذا الملف لوظائف داخلية

TOOLS\_SYSTEM, // افتراض وجود هذا الملف لوظائف النظام (مثل diagnoseSystem)

HELPERS

) {

'use strict';

/\*\*

\* @typedef {Object} ToolParameterProperty

\* @property {string} type - نوع المعامل (مثال: 'STRING', 'NUMBER', 'BOOLEAN', 'OBJECT').

\* @property {string} description - وصف المعامل.

\*/

/\*\*

\* @typedef {Object} ToolParametersSchema

\* @property {string} type - يجب أن تكون 'OBJECT' لمعاملات الدوال.

\* @property {Object.<string, ToolParameterProperty>} properties - كائنات المعاملات.

\* @property {string[]} required - مصفوفة بأسماء المعاملات المطلوبة.

\*/

/\*\*

\* @typedef {Object} ToolDefinition

\* @property {string} name - اسم الدالة/الأداة (يجب أن يتطابق مع اسم الدالة الفعلية).

\* @property {function} func - مرجع للدالة الفعلية التي تنفذ الأداة.

\* @property {string} description - وصف موجز لوظيفة الأداة.

\* @property {string} category - تصنيف الأداة (مثال: 'Sheets', 'Accounting', 'General', 'System', 'Internal').

\* @property {ToolParametersSchema} [parameters] - مخطط المعاملات التي تقبلها الأداة.

\* @property {boolean} [internal=false] - هل هذه الأداة داخلية ولا يجب تقديمها لنموذج Gemini.

\*/

// --- 1. تعريفات الأدوات الشاملة (Function Definitions) ومراجع الدوال الفعلية ---

// هذا الهيكل يجمع بين معلومات الأداة الكاملة لاستخدامها داخليًا ولـ Gemini API

const ALL\_TOOLS\_DEFINITIONS = [

// -------------------------------------------------------------------------

// أدوات الجداول (TOOLS\_SHEETS) - الفئة: 'Sheets'

// -------------------------------------------------------------------------

{

name: 'deleteRow',

func: TOOLS\_SHEETS.deleteRow,

description: 'يحذف صفًا محددًا من الورقة النشطة. يتطلب رقم الصف.',

category: 'Sheets',

parameters: {

type: 'OBJECT',

properties: {

row: { type: 'NUMBER', description: 'رقم الصف المراد حذفه.' }

},

required: ['row']

}

},

{

name: 'clearRange',

func: TOOLS\_SHEETS.clearRange,

description: 'يمسح محتوى نطاق محدد في الورقة النشطة. يتطلب نطاقًا بصيغة A1 (مثال: "A1:B10").',

category: 'Sheets',

parameters: {

type: 'OBJECT',

properties: {

rangeA1: { type: 'STRING', description: 'النطاق المراد مسحه بصيغة A1.' }

},

required: ['rangeA1']

}

},

{

name: 'filterByValue',

func: TOOLS\_SHEETS.filterByValue,

description: 'يطبق فلتر على عمود معين بناءً على قيمة محددة.',

category: 'Sheets',

parameters: {

type: 'OBJECT',

properties: {

column: { type: 'NUMBER', description: 'رقم العمود (1-based) لتطبيق الفلتر عليه.' },

value: { type: 'STRING', description: 'القيمة التي سيتم الفلترة بها.' }

},

required: ['column', 'value']

}

},

{

name: 'sortColumn',

func: TOOLS\_SHEETS.sortColumn,

description: 'يقوم بفرز البيانات في عمود محدد تصاعديًا أو تنازليًا.',

category: 'Sheets',

parameters: {

type: 'OBJECT',

properties: {

column: { type: 'NUMBER', description: 'رقم العمود (1-based) للفرز.' },

ascending: { type: 'BOOLEAN', description: 'صحيح للفرز التصاعدي، خطأ للفرز التنازلي.' }

},

required: ['column', 'ascending']

}

},

{

name: 'writeToCell',

func: TOOLS\_SHEETS.writeToCell,

description: 'يكتب قيمة محددة في خلية معينة.',

category: 'Sheets',

parameters: {

type: 'OBJECT',

properties: {

cellA1: { type: 'STRING', description: 'عنوان الخلية بصيغة A1 (مثال: "C5").' },

value: { type: 'STRING', description: 'القيمة المراد كتابتها في الخلية.' }

},

required: ['cellA1', 'value']

}

},

{

name: 'highlightEmptyCells',

func: TOOLS\_SHEETS.highlightEmptyCells,

description: 'يقوم بتمييز الخلايا الفارغة في نطاق محدد بلون معين. مفيد لتحديد البيانات المفقودة.',

category: 'Sheets',

parameters: {

type: 'OBJECT',

properties: {

sheetName: { type: 'STRING', description: 'اسم الورقة التي سيتم تطبيق التمييز عليها.' },

rangeA1: { type: 'STRING', description: 'النطاق المراد فحصه بصيغة A1 (مثال: "A1:Z100").' },

color: { type: 'STRING', description: 'اللون الذي سيتم استخدامه للتمييز (مثال: "#FF0000" للأحمر).' }

},

required: ['sheetName', 'rangeA1', 'color']

}

},

// -------------------------------------------------------------------------

// أدوات المحاسبة (TOOLS\_ACCOUNTING) - الفئة: 'Accounting'

// -------------------------------------------------------------------------

{

name: 'logExpenseEntry',

func: TOOLS\_ACCOUNTING.logExpenseEntry,

description: 'يسجل إدخال مصروف جديد في سجل المصروفات.',

category: 'Accounting',

parameters: {

type: 'OBJECT',

properties: {

amount: { type: 'NUMBER', description: 'مبلغ المصروف.' },

description: { type: 'STRING', description: 'وصف المصروف.' },

category: { type: 'STRING', description: 'فئة المصروف.' }

},

required: ['amount', 'description', 'category']

}

},

{

name: 'logRevenueEntry',

func: TOOLS\_ACCOUNTING.logRevenueEntry,

description: 'يسجل إدخال إيراد جديد في سجل الإيرادات.',

category: 'Accounting',

parameters: {

type: 'OBJECT',

properties: {

amount: { type: 'NUMBER', description: 'مبلغ الإيراد.' },

description: { type: 'STRING', description: 'وصف الإيراد.' },

source: { type: 'STRING', description: 'مصدر الإيراد.' }

},

required: ['amount', 'description', 'source']

}

},

{

name: 'generateIncomeStatement',

func: TOOLS\_ACCOUNTING.generateIncomeStatement,

description: 'ينشئ بيان دخل (قائمة الأرباح والخسائر) للفترة المحددة.',

category: 'Accounting',

parameters: {

type: 'OBJECT',

properties: {

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

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

},

required: ['startDate', 'endDate']

}

},

// -------------------------------------------------------------------------

// أدوات عامة (TOOLS\_GENERAL) - الفئة: 'General'

// -------------------------------------------------------------------------

{

name: 'exportToPdfAndEmail',

func: TOOLS\_GENERAL.exportToPdfAndEmail,

description: 'يقوم بتصدير الورقة النشطة إلى ملف PDF وإرساله بالبريد الإلكتروني إلى مستلم محدد.',

category: 'General',

parameters: {

type: 'OBJECT',

properties: {

recipientEmail: { type: 'STRING', description: 'عنوان البريد الإلكتروني للمستلم.' },

subject: { type: 'STRING', description: 'موضوع البريد الإلكتروني.' },

body: { type: 'STRING', description: 'نص رسالة البريد الإلكتروني (اختياري).' }

},

required: ['recipientEmail', 'subject']

}

},

{

name: 'createBackup',

func: TOOLS\_GENERAL.createBackup,

description: 'ينشئ نسخة احتياطية من جدول البيانات الحالي في Google Drive.',

category: 'General',

parameters: {

type: 'OBJECT',

properties: {

folderName: { type: 'STRING', description: 'اسم المجلد في Google Drive لحفظ النسخة الاحتياطية فيه (اختياري).' }

},

required: []

}

},

{

name: 'listWorkshopSnippets',

func: TOOLS\_GENERAL.listWorkshopSnippets,

description: 'يسرد مقتطفات التعليمات البرمجية المتاحة في ورشة العمل.',

category: 'General',

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

},

{

name: 'proposeAndStageCode',

func: TOOLS\_GENERAL.proposeAndStageCode,

description: 'يقترح مقتطف كود جديد لإضافته إلى ورشة العمل للمراجعة.',

category: 'General',

parameters: {

type: 'OBJECT',

properties: {

functionName: { type: 'STRING', description: 'اسم الوظيفة المقترحة (مثال: "myNewFunction").' },

functionDescription: { type: 'STRING', description: 'وصف موجز لوظيفة الكود المقترح.' },

functionCode: { type: 'STRING', description: 'مقتطف الكود الفعلي للوظيفة.' }

},

required: ['functionName', 'functionDescription', 'functionCode']

}

},

// -------------------------------------------------------------------------

// أدوات النظام (TOOLS\_SYSTEM) - الفئة: 'System'

// -------------------------------------------------------------------------

{

name: 'diagnoseSystem',

func: TOOLS\_SYSTEM.diagnoseSystem,

description: 'يقوم بتشخيص حالة النظام وتقديم تقرير عن أي مشاكل محتملة.',

category: 'System',

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

},

// -------------------------------------------------------------------------

// أدوات داخلية (TOOLS\_INTERNAL) - الفئة: 'Internal' (لا تُقدم لـ Gemini API)

// -------------------------------------------------------------------------

{

name: '\_internal\_userConfirmed', // تبدأ بـ '\_' للإشارة إلى أنها داخلية

func: TOOLS\_INTERNAL.\_internal\_userConfirmed,

description: 'تُستخدم داخليًا لتأكيد إجراء بعد موافقة المستخدم. لا يجب استدعاؤها مباشرة من النموذج.',

category: 'Internal',

parameters: {

type: 'OBJECT',

properties: {

payload: { type: 'OBJECT', description: 'الحمولة الأصلية لطلب التأكيد.' },

confirmationType: { type: 'STRING', description: 'نوع التأكيد (مثلاً "delete\_row").' }

},

required: ['payload', 'confirmationType']

},

internal: true // علامة مخصصة للإشارة إلى أنها داخلية ولا يجب تقديمها للنموذج

}

];

// --- 2. كائن لسهولة الوصول إلى الدوال الفعلية بواسطة الاسم ---

// هذا يُستخدم بواسطة AI\_DISPATCHER لتنفيذ الدالة بعد أن يطلبها النموذج

const AVAILABLE\_TOOLS = ALL\_TOOLS\_DEFINITIONS.reduce((acc, tool) => {

acc[tool.name] = tool.func; // استخدام 'func'

return acc;

}, {});

/\*\*

\* @function getFunctionList

\* @description

\* تُرجع قائمة بتعريفات الأدوات (Function Declarations) التي يمكن لنموذج Gemini استدعاؤها.

\* تستبعد الأدوات التي تم وضع علامة 'internal: true' عليها.

\* @returns {Array<object>} - مصفوفة من كائنات تعريف الدالة بالصيغة المطلوبة لـ Gemini API.

\*/

function getFunctionList() {

return ALL\_TOOLS\_DEFINITIONS

.filter(tool => !tool.internal) // تصفية الأدوات الداخلية

.map(tool => ({

name: tool.name,

description: tool.description,

parameters: tool.parameters || { type: 'OBJECT', properties: {}, required: [] } // التأكد من وجود كائن المعاملات

}));

}

/\*\*

\* @function getToolMetadata

\* @description

\* تُرجع تعريف الأداة الكامل (ToolDefinition) لأداة معينة بالاسم، بما في ذلك التصنيف.

\* تُستخدم لغرض التوثيق أو التحقق الداخلي.

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

\* @returns {ToolDefinition|null} - كائن تعريف الأداة الكامل أو null إذا لم يتم العثور عليها.

\*/

function getToolMetadata(toolName) {

return ALL\_TOOLS\_DEFINITIONS.find(t => t.name === toolName) || null;

}

/\*\*

\* @function getAllToolCategories

\* @description

\* تُرجع قائمة بجميع التصنيفات الفريدة للأدوات المسجلة.

\* @returns {string[]} - مصفوفة من أسماء التصنيفات الفريدة.

\*/

function getAllToolCategories() {

const categories = new Set();

ALL\_TOOLS\_DEFINITIONS.forEach(tool => {

if (tool.category) {

categories.add(tool.category);

}

});

return Array.from(categories).sort();

}

/\*\*

\* @function getToolsByCategory

\* @description

\* تُرجع قائمة بالأدوات ضمن تصنيف معين.

\* @param {string} category - اسم التصنيف.

\* @param {boolean} [includeInternal=false] - هل يتم تضمين الأدوات الداخلية في النتائج.

\* @returns {ToolDefinition[]} - مصفوفة من كائنات تعريف الأداة.

\*/

function getToolsByCategory(category, includeInternal = false) {

return ALL\_TOOLS\_DEFINITIONS.filter(tool =>

tool.category === category && (includeInternal || !tool.internal)

);

}

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

return {

getFunctionList,

getToolMetadata,

getAllToolCategories, // إضافة الدالة الجديدة

getToolsByCategory, // إضافة الدالة الجديدة

AVAILABLE\_TOOLS // يتم تصديرها لـ AI\_DISPATCHER لتنفيذ الدوال

};

})(

AI\_TOOLS\_DIALOGUE,

TOOLS\_SHEETS,

TOOLS\_ACCOUNTING,

TOOLS\_GENERAL,

TOOLS\_INTERNAL,

TOOLS\_SYSTEM,

HELPERS

);