/\*\*

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

\* @file builtinDocs.gs – v3.1

\* @module BUILTIN\_DOCS

\* @description

\* توثيق الأدوات المدمجة تلقائيًا داخل مشروع G-Assistant.

\* تم تحديثه لدعم البحث عن دوال Apps Script المدمجة باستخدام نموذج Gemini.

\* يتعامل مع:

\* - فتح واجهة البحث عن دوال Apps Script (BuiltinDocsSidebar.html)

\* - جلب توثيق دوال Apps Script من خلال LLM (Gemini)

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

\*/

var BUILTIN\_DOCS = (function(TOOLS\_CATALOG, SETTINGS\_PROVIDER, AI\_TOOLS\_DIALOGUE) {

'use strict';

// API Key for Gemini API. This will be automatically provided by the Canvas environment.

const API\_KEY = ""; // Leave empty for Canvas environment to inject

const GEMINI\_API\_URL = `https://generativelanguage.googleapis.com/v1beta/models/gemini-2.0-flash:generateContent?key=${API\_KEY}`;

/\*\*

\* @function showDocumentationSidebar

\* @description

\* 🔍 يعرض الشريط الجانبي للبحث عن دوال Apps Script المدمجة.

\* يقوم بتحميل BuiltinDocsSidebar.html.

\*/

function showDocumentationSidebar() {

try {

const html = HtmlService.createHtmlOutputFromFile('BuiltinDocsSidebar.html')

.setTitle('📚 البحث في دوال Apps Script')

.setWidth(500);

SpreadsheetApp.getUi().showSidebar(html);

} catch (error) {

AI\_TOOLS\_DIALOGUE.showError(`حدث خطأ أثناء عرض الشريط الجانبي للتوثيق: ${error.message}`);

console.error("Error in showDocumentationSidebar:", error);

}

}

/\*\*

\* @function getBuiltinFunctionDoc

\* @description

\* يستخدم نموذج Gemini لجلب توثيق لدالة Apps Script مدمجة.

\* يرسل استعلامًا إلى Gemini ويحلل الاستجابة لتقديمها في الواجهة الأمامية.

\* @param {string} functionName - اسم دالة Apps Script المراد توثيقها (مثال: 'SpreadsheetApp.getActiveSpreadsheet').

\* @returns {object} - كائن يحتوي على توثيق الدالة بصيغة HTML أو نص عادي، أو رسالة خطأ.

\*/

function getBuiltinFunctionDoc(functionName) {

try {

const prompt = `قدم توثيقًا شاملاً لدالة Google Apps Script التالية: \`${functionName}\`.

يجب أن يتضمن التوثيق:

- وصفًا موجزًا للدالة.

- فئة الدالة (مثال: SpreadsheetApp, DocumentApp, Utilities).

- الصيغة الكاملة للدالة مع أنواع المعاملات (Parameters) ووصف لكل معامل.

- قيمة الإرجاع (Return Value) مع نوعها ووصفها.

- مثال عملي لاستخدام الدالة في كود Apps Script.

قدم هذه المعلومات في تنسيق JSON بالهيكل التالي:

\`\`\`json

{

"functionName": "اسم\_الدالة",

"description": "وصف\_الدالة",

"category": "فئة\_الدالة",

"syntax": "صيغة\_الدالة(param1: Type, param2: Type)",

"parameters": [

{"name": "param1", "type": "Type", "description": "وصف\_المعامل1"},

{"name": "param2", "type": "Type", "description": "وصف\_المعامل2"}

],

"returnValue": {"type": "Type", "description": "وصف\_قيمة\_الإرجاع"},

"exampleCode": "كود\_مثال\_باستخدام\_الدالة"

}

\`\`\`

إذا لم تتمكن من العثور على توثيق، أعد كائن JSON فارغًا أو رسالة خطأ مناسبة.

`;

const payload = {

contents: [{ role: "user", parts: [{ text: prompt }] }],

generationConfig: {

responseMimeType: "application/json",

responseSchema: {

type: "OBJECT",

properties: {

"functionName": { "type": "STRING" },

"description": { "type": "STRING" },

"category": { "type": "STRING" },

"syntax": { "type": "STRING" },

"parameters": {

"type": "ARRAY",

"items": {

"type": "OBJECT",

"properties": {

"name": { "type": "STRING" },

"type": { "type": "STRING" },

"description": { "type": "STRING" }

}

}

},

"returnValue": {

"type": "OBJECT",

"properties": {

"type": { "type": "STRING" },

"description": { "type": "STRING" }

}

},

"exampleCode": { "type": "STRING" }

}

}

}

};

const options = {

method: 'post',

contentType: 'application/json',

payload: JSON.stringify(payload),

muteHttpExceptions: true

};

const response = UrlFetchApp.fetch(GEMINI\_API\_URL, options);

const statusCode = response.getResponseCode();

const responseText = response.getContentText();

if (statusCode !== 200) {

throw new Error(`Gemini API Error: ${statusCode} - ${responseText}`);

}

const result = JSON.parse(responseText);

const docData = result?.candidates?.[0]?.content?.parts?.[0]?.text ? JSON.parse(result.candidates[0].content.parts[0].text) : null;

if (docData && docData.functionName) {

let htmlTable = `<table style="width:100%; border-collapse: collapse; margin-top:15px;">`;

htmlTable += `<tr><td style="background-color:#3a3a3a; color:#7bbaff; font-weight:bold;">الدالة:</td><td>${docData.functionName}</td></tr>`;

htmlTable += `<tr><td style="background-color:#3a3a3a; color:#7bbaff; font-weight:bold;">الوصف:</td><td>${docData.description || 'لا يوجد وصف.'}</td></tr>`;

htmlTable += `<tr><td style="background-color:#3a3a3a; color:#7bbaff; font-weight:bold;">الفئة:</td><td>${docData.category || 'غير محددة.'}</td></tr>`;

if (docData.syntax) {

htmlTable += `<tr><td style="background-color:#3a3a3a; color:#7bbaff; font-weight:bold;">الصيغة:</td><td><pre style="background:none; margin:0; padding:0; color:#e0e0e0; overflow-x: auto;"><code>${docData.syntax}</code></pre></td></tr>`;

}

if (docData.parameters && docData.parameters.length > 0) {

htmlTable += `<tr><td style="background-color:#3a3a3a; color:#7bbaff; font-weight:bold;">المعاملات:</td><td><ul>`;

docData.parameters.forEach(p => {

htmlTable += `<li><strong>${p.name}</strong>: <code>${p.type || 'غير محدد'}</code> - ${p.description || 'لا يوجد وصف.'}</li>`;

});

htmlTable += `</ul></td></tr>`;

}

if (docData.returnValue && docData.returnValue.type) {

htmlTable += `<tr><td style="background-color:#3a3a3a; color:#7bbaff; font-weight:bold;">قيمة الإرجاع:</td><td><code>${docData.returnValue.type}</code> - ${docData.returnValue.description || 'لا يوجد وصف.'}</td></tr>`;

}

if (docData.exampleCode) {

htmlTable += `<tr><td style="background-color:#3a3a3a; color:#7bbaff; font-weight:bold;">مثال:</td><td><pre style="background-color: #4a4a4a; padding: 10px; border-radius: 5px; overflow-x: auto; font-family: monospace; font-size: 0.85em; color: #e0e0e0;"><code>${docData.exampleCode}</code></pre></td></tr>`;

}

htmlTable += `</table>`;

// Plain text version for fallback or simpler display

let plainText = `\*\*الدالة:\*\* ${docData.functionName}\n`;

plainText += `\*\*الوصف:\*\* ${docData.description || 'لا يوجد وصف.'}\n`;

plainText += `\*\*الفئة:\*\* ${docData.category || 'غير محددة.'}\n`;

if (docData.syntax) {

plainText += `\*\*الصيغة:\*\* \`${docData.syntax}\`\n`;

}

if (docData.parameters && docData.parameters.length > 0) {

plainText += `\*\*المعاملات:\*\*\n`;

docData.parameters.forEach(p => {

plainText += ` - ${p.name}: ${p.type || 'غير محدد'} - ${p.description || 'لا يوجد وصف.'}\n`;

});

}

if (docData.returnValue && docData.returnValue.type) {

plainText += `\*\*قيمة الإرجاع:\*\* ${docData.returnValue.type} - ${docData.returnValue.description || 'لا يوجد وصف.'}\n`;

}

if (docData.exampleCode) {

plainText += `\*\*مثال:\*\*\n\`\`\`javascript\n${docData.exampleCode}\n\`\`\`\n`;

}

return { table: htmlTable, text: plainText };

} else {

console.warn("Unexpected LLM response structure:", result);

return { text: "تعذر الحصول على توثيق لهذه الدالة. قد لا تكون دالة Apps Script مدمجة أو حدث خطأ في النموذج." };

}

} catch (error) {

console.error("Error in getBuiltinFunctionDoc:", error);

return { text: `حدث خطأ أثناء جلب التوثيق: ${error.message}` };

}

}

return {

showDocumentationSidebar: showDocumentationSidebar,

getBuiltinFunctionDoc: getBuiltinFunctionDoc

};

})(TOOLS\_CATALOG, SETTINGS\_PROVIDER, AI\_TOOLS\_DIALOGUE);