/\*\*

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

\* @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 = "";

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 للبحث عن الدوال: ${error.message}`);

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

}

}

/\*\*

\* @function getBuiltinFunctionDoc

\* @description

\* 📚 يجلب توثيق دالة Apps Script محددة باستخدام نموذج Gemini.

\* هذه الدالة يتم استدعاؤها من BuiltinDocsSidebar.html.

\* @param {string} functionName - اسم دالة Apps Script المراد البحث عنها (مثال: 'getRange', 'SpreadsheetApp').

\* @returns {object} كائن يحتوي على 'table' (HTML لجدول المعاملات) و/أو 'text' (نص التوثيق).

\*/

function getBuiltinFunctionDoc(functionName) {

try {

if (!functionName || typeof functionName !== 'string') {

throw new Error("اسم الدالة غير صالح.");

}

// بناء الطلب للنموذج اللغوي

const prompt = `Provide the documentation for the Google Apps Script function or class named "${functionName}".

Include its purpose, syntax, parameters (name, type, description), return value, and a simple, runnable code example.

Format the response as a JSON object with the following structure:

{

"functionName": "string",

"description": "string",

"syntax": "string",

"parameters": [

{ "name": "string", "type": "string", "description": "string" }

],

"returnValue": { "type": "string", "description": "string" },

"exampleCode": "string"

}

If it's a class (like SpreadsheetApp), describe its purpose and list its most common methods with brief descriptions. For classes, the 'parameters' and 'returnValue' fields can be empty arrays/objects.`;

const payload = {

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

generationConfig: {

responseMimeType: "application/json",

responseSchema: {

type: "OBJECT",

properties: {

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

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

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

"parameters": {

"type": "ARRAY",

"items": {

"type": "OBJECT",

"properties": {

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

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

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

},

"propertyOrdering": ["name", "type", "description"]

}

},

"returnValue": {

"type": "OBJECT",

"properties": {

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

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

},

"propertyOrdering": ["type", "description"]

},

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

},

"propertyOrdering": ["functionName", "description", "syntax", "parameters", "returnValue", "exampleCode"]

}

}

};

const options = {

method: 'post',

contentType: 'application/json',

payload: JSON.stringify(payload)

};

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

const result = JSON.parse(response.getContentText());

if (result.candidates && result.candidates.length > 0 &&

result.candidates[0].content && result.candidates[0].content.parts &&

result.candidates[0].content.parts.length > 0) {

const jsonString = result.candidates[0].content.parts[0].text;

const docData = JSON.parse(jsonString);

let htmlTable = '';

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

htmlTable += '<h4 style="margin-top:15px; margin-bottom: 8px; color: #333; font-size: 1.1em;">المعاملات:</h4>';

htmlTable += '<table style="width:100%; border-collapse: collapse; margin-bottom: 15px;">';

htmlTable += '<thead><tr style="background-color: #e0e0e0;"><th style="padding: 8px; border: 1px solid #ddd; text-align: right;">الاسم</th><th style="padding: 8px; border: 1px solid #ddd; text-align: right;">النوع</th><th style="padding: 8px; border: 1px solid #ddd; text-align: right;">الوصف</th></tr></thead>';

htmlTable += '<tbody>';

docData.parameters.forEach(p => {

htmlTable += `<tr>

<td style="padding: 8px; border: 1px solid #ddd; text-align: right;"><code>${p.name}</code></td>

<td style="padding: 8px; border: 1px solid #ddd; text-align: right;">${p.type}</td>

<td style="padding: 8px; border: 1px solid #ddd; text-align: right;">${p.description || '—'}</td>

</tr>`;

});

htmlTable += '</tbody></table>';

}

let plainText = `<h3>${docData.functionName}</h3>`;

plainText += `<p><strong>الوصف:</strong> ${docData.description || 'لا يوجد وصف.'}</p>`;

if (docData.syntax) {

plainText += `<p><strong>الصيغة:</strong> <code>${docData.syntax}</code></p>`;

}

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

plainText += `<p><strong>قيمة الإرجاع:</strong> ${docData.returnValue.type} (${docData.returnValue.description || 'لا يوجد وصف.'})</p>`;

}

if (docData.exampleCode) {

plainText += `<h4 style="margin-top:15px; margin-bottom: 8px; color: #333; font-size: 1.1em;">مثال:</h4>`;

plainText += `<pre style="background-color: #f0f0f0; padding: 10px; border-radius: 5px; overflow-x: auto;"><code>${docData.exampleCode}</code></pre>`;

}

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,

// تم إزالة exportDocumentationToDoc من هنا لتركيز الوحدة على توثيق دوال Apps Script المدمجة

};

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

// لتمكين google.script.run من استدعاء الدالة مباشرة، يجب أن تكون متاحة في النطاق العام.

// هذه الدالة هي الواجهة العامة لـ getBuiltinFunctionDoc.

function getBuiltinFunctionDoc(functionName) {

return BUILTIN\_DOCS.getBuiltinFunctionDoc(functionName);

}