// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// --- START OF FILE: 30\_tools/tools\_developer.gs ---

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*\*

\* @file 30\_tools/tools\_developer.gs

\* @module System.ToolsDeveloper

\* @version 20

\* @author عبدالعزيز

\* @description

\* أدوات مخصصة لمساعدة المطورين، تدعم سير عمل متكامل للمراجعة،

\* التصنيف، التجهيز، والتوثيق الذكي.

\* المراحل المعمارية:

\* • 1: تحويل الوحدة إلى defineModule وربط التبعيات

\* • 8: تكامل مع AgentDeveloper لسير العمل التلقائي

\* • 10: حفظ الأحداث في الذاكرة طويلة الأمد

\* • 17: تسجيل مقاييس الاستخدام في أوراق Google Sheets

\*/

defineModule('System.ToolsDeveloper', ({ Utils, Agents, Tools, Config, UI, AI, DocsManager }) => {

// تسجيل الوثائق (مرحلة 9 لدعم DocsManager)

DocsManager.registerModuleDocs('System.ToolsDeveloper', [

{

name: 'reviewAndStageCode',

description: 'تراجع الكود البرمجي، تقترح ملفًا، ثم تجهز كل شيء في ورشة العمل.',

parameters: {

type: 'OBJECT',

properties: {

functionCode: { type: 'STRING', description: 'الكود المراد مراجعته وتجهيزه.' },

category: { type: 'STRING', description: 'فئة الكود (اختياري).' }

},

required: ['functionCode']

}

},

{

name: 'getBuiltinFunctionDoc',

description: 'يجلب توثيق دالة مدمجة في Apps Script، مع استخدام الكاش أولاً.',

parameters: {

type: 'OBJECT',

properties: {

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

},

required: ['functionName']

}

},

{

name: 'exportToolsDocumentationToDoc',

description: 'يصدر توثيق جميع الأدوات إلى مستند Google Docs.',

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

}

]);

/\*\*

\* يراجع الكود، يقترح ملفًا، ثم يجهز كل شيء في ورشة العمل.

\*/

function reviewAndStageCode({ functionCode, category }) {

return Utils.executeSafely(() => {

Utils.log('ToolsDeveloper.reviewAndStageCode: Starting...', { codeLength: functionCode.length });

// اقتراح اسم الملف

const suggestedFile = \_suggestTargetFile(functionCode);

// مراجعة الكود باستخدام أدوات CodeReview

const reviewResponse = Tools.CodeReview.reviewCode({ code: functionCode, reviewType: 'general' });

// صياغة ملخص المراجعة

let reviewSummary = 'فشلت عملية المراجعة التلقائية.';

if (reviewResponse.type === 'success' && reviewResponse.data) {

reviewSummary = `التقييم: ${reviewResponse.data.overallScore || 'N/A'}/100. ملخص: ${reviewResponse.data.summary || 'لا يوجد'}`;

} else if (reviewResponse.type === 'table') {

reviewSummary = reviewResponse.text;

}

const inferredCategory = category || \_categorizeCode(functionCode);

// تجهيز في ورشة العمل

const uiResponse = \_stageCodeInWorkshop(functionCode, suggestedFile, reviewSummary, inferredCategory);

// المرحلة 10: حفظ حدث المراجعة والتجهيز في الذاكرة طويلة الأمد

AI.LongTermMemory.save('ReviewAndStage', {

source: 'ToolsDeveloper',

category: inferredCategory,

suggestedFile,

codeLength: functionCode.length

});

// المرحلة 17: تسجيل مقاييس الاستخدام في ورقة Metrics

const metricsSheet = Utils.getSheet(

'DeveloperTools\_Metrics',

['Timestamp', 'Action', 'Category', 'SuggestedFile', 'CodeLength']

);

metricsSheet.appendRow([

new Date(),

'reviewAndStageCode',

inferredCategory,

suggestedFile,

functionCode.length

]);

return uiResponse;

}, [], 'ToolsDeveloper.reviewAndStageCode');

}

/\*\*

\* يجهز الكود في ورشة العمل المسماة في الإعدادات.

\* @private

\*/

function \_stageCodeInWorkshop(code, file, summary, category) {

const { Utils, Config, UI } = GAssistant.Utils.Injector.get('Utils', 'Config', 'UI');

const sheetName = Config.get('DEVELOPER\_WORKSHOP\_SHEET');

const headers = ['التاريخ', 'الفئة', 'الملف المقترح', 'ملخص المراجعة', 'الكود', 'الحالة'];

const sheet = Utils.getSheet(sheetName, headers);

if (!sheet) return UI.Dialogue.createError('فشل الوصول إلى ورشة عمل المطور.');

sheet.appendRow([new Date(), category, file, summary, code, 'للمراجعة']);

return UI.Dialogue.createInfo(`✅ تم تجهيز الكود في ورشة العمل: "${sheetName}".`);

}

/\*\*

\* يقترح ملفًا مناسبًا استنادًا إلى أنماط محددة في الإعدادات.

\* @private

\*/

function \_suggestTargetFile(functionCode) {

const { Config } = GAssistant.Utils.Injector.get('Config');

const text = (functionCode || '').toLowerCase();

const patterns = Config.get('TOOL\_FILE\_SUGGESTION\_PATTERNS') || [];

let bestMatch = { file: 'utils.gs', score: 0 };

const scores = {};

patterns.forEach(p => {

scores[p.file] = scores[p.file] || 0;

p.keywords.forEach(kw => {

if (text.includes(kw.toLowerCase())) scores[p.file] += p.weight;

});

if (scores[p.file] > bestMatch.score) {

bestMatch = { file: p.file, score: scores[p.file] };

}

});

return bestMatch.file;

}

/\*\*

\* يصنف الكود إلى فئة وفق كلمات مفتاحية بسيطة.

\* @private

\*/

function \_categorizeCode(code) {

const lc = (code || '').toLowerCase();

if (lc.includes('accounting') || lc.includes('profit')) return 'أدوات مالية';

if (lc.includes('sheet') || lc.includes('range')) return 'أدوات Sheets';

if (lc.includes('sidebar') || lc.includes('menu')) return 'واجهة مستخدم';

if (lc.includes('gemini') || lc.includes('prompt'))return 'ذكاء اصطناعي';

return 'وظائف عامة';

}

/\*\*

\* يجلب توثيق دالة Apps Script مدمجة، مع استخدام الكاش أولاً.

\*/

function getBuiltinFunctionDoc({ functionName }) {

return Utils.executeSafely(() => {

const cached = AI.LongTermMemory.getCachedFunctionDoc(functionName);

if (cached) {

Utils.log('ToolsDeveloper.getBuiltinFunctionDoc: Retrieved from cache.', { functionName });

return AI.Core.ask(`أعد صياغة هذا التوثيق بصيغة Markdown:\n${cached}`);

}

const prompt = `قدم توثيقًا شاملاً لدالة Google Apps Script التالية: \`${functionName}\`. أعد الرد بصيغة Markdown.`;

const result = AI.Core.ask(prompt, { modelOverride: Config.get('GEMINI\_FLASH\_MODEL') });

if (result.type === 'info') {

AI.LongTermMemory.cacheFunctionDoc(functionName, result.text);

// المرحلة 10: حفظ حدث جلب التوثيق

AI.LongTermMemory.save('GetBuiltinDoc', {

source: 'ToolsDeveloper',

functionName

});

// المرحلة 17: تسجيل مقياس جلب التوثيق

const docMetrics = Utils.getSheet('DeveloperDoc\_Metrics', ['Timestamp', 'FunctionName']);

docMetrics.appendRow([new Date(), functionName]);

}

Utils.log('ToolsDeveloper.getBuiltinFunctionDoc: Fetched from API.', { functionName });

return result;

}, [], `ToolsDeveloper.getBuiltinFunctionDoc[${functionName}]`);

}

/\*\*

\* يصدر توثيق الأدوات إلى مستند Google Docs.

\*/

function exportToolsDocumentationToDoc() {

return Utils.executeSafely(() => {

const doc = DocumentApp.create(`توثيق أدوات G-Assistant - ${new Date().toLocaleDateString()}`);

const body = doc.getBody();

body.appendParagraph('توثيق أدوات G-Assistant').setHeading(DocumentApp.ParagraphHeading.TITLE);

Tools.Catalog.getDeclarations().forEach(tool => {

body.appendParagraph(tool.name).setHeading(DocumentApp.ParagraphHeading.HEADING1);

body.appendParagraph(tool.description).setItalic(true);

const params = tool.parameters.properties || {};

if (Object.keys(params).length) {

body.appendParagraph('المعاملات:').setHeading(DocumentApp.ParagraphHeading.HEADING2);

const table = body.appendTable([['المعامل', 'النوع', 'الوصف', 'مطلوب؟']]);

table.getRow(0).setAttributes({

[DocumentApp.Attribute.BOLD]: true,

[DocumentApp.Attribute.BACKGROUND\_COLOR]: '#f3f3f3'

});

tool.parameters.required.forEach(name => {

const p = params[name];

table.appendTableRow([name, `\`${p.type}\``, p.description, 'نعم']);

});

tool.parameters.required.filter(n => !params[n]).forEach(name => {

// in case of additional or optional props

table.appendTableRow([name, '—', '—', 'لا']);

});

}

body.appendHorizontalRule();

});

// المرحلة 10: حفظ حدث تصدير التوثيق

AI.LongTermMemory.save('ExportToolsDoc', {

source: 'ToolsDeveloper',

docUrl: doc.getUrl()

});

// المرحلة 17: تسجيل مقياس تصدير التوثيق

const expMetrics = Utils.getSheet('DeveloperExport\_Metrics', ['Timestamp', 'DocUrl']);

expMetrics.appendRow([new Date(), doc.getUrl()]);

Utils.log('ToolsDeveloper.exportToolsDocumentationToDoc: Success', { url: doc.getUrl() });

SpreadsheetApp.getUi().alert(`✅ تم تصدير التوثيق. الرابط:\n${doc.getUrl()}`);

}, [], 'ToolsDeveloper.exportToolsDocumentationToDoc');

}

return {

reviewAndStageCode,

getBuiltinFunctionDoc,

exportToolsDocumentationToDoc

};

});

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// --- END OF FILE: 30\_tools/tools\_developer.gs ---

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*