/\*\*

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

\* @file tools\_code\_review.gs – v1.4

\* @module GAssistant.Tools.CodeReview

\* @description

\* هذه الوحدة تعمل كواجهة بين واجهة المستخدم (BuiltinDocsSidebar.html الموحد) ووحدة طبيب السكربت الذكي

\* (GAssistant.AI.ScriptDoctor) لمراجعة الكود.

\* توفر دوال عامة يمكن لـ google.script.run استدعاؤها لجلب تغذية راجعة حول الكود، شرحه، توليد اختبارات الوحدة،

\* توليد تعليقات JSDoc، واقتراح تحسينات الأداء.

\* تعتمد على: GAssistant.AI.ScriptDoctor, GAssistant.Helpers, GAssistant.AI.Dialogue, GAssistant.AI.Core

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

\*/

var GAssistant = GAssistant || {};

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

GAssistant.Tools.CodeReview = (function(

GA\_ScriptDoctor,

GA\_Helpers,

GA\_AI\_Dialogue,

GA\_AI\_Core // إضافة تبعية لـ AI\_Core

) {

'use strict';

/\*\*

\* @function reviewCodeForSidebar

\* @description

\* 📊 واجهة عامة تستخدمها الواجهة الأمامية (BuiltinDocsSidebar.html الموحد) لجلب تغذية راجعة حول الكود.

\* تقوم باستدعاء GAssistant.AI.ScriptDoctor.getReviewFeedbackForSidebar

\* وتمرير الكود ونوع المراجعة إليها.

\* @param {string} code - الكود البرمجي المراد مراجعته.

\* @param {string} reviewType - نوع المراجعة المطلوبة ('general', 'refactor', 'security', 'unit\_tests').

\* @returns {object} - كائن يحتوي على overallScore, summary, feedbackItems[], و patch (إذا وجد) للعرض في الشريط الجانبي.

\*/

function reviewCodeForSidebar(code, reviewType) {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'reviewCodeForSidebar:start', { length: code.length, reviewType: reviewType });

try {

// استدعاء دالة المراجعة من Gemini Script Doctor مباشرة.

// نفترض أن GA\_ScriptDoctor.getReviewFeedbackForSidebar يعيد كائنًا يحتوي على:

// { overallScore, summary, feedbackItems: [], patch: string (optional) }

const result = GA\_ScriptDoctor.getReviewFeedbackForSidebar(code, reviewType);

// التحقق من أن النتيجة تحتوي على overallScore أو error لتجنب الردود غير المكتملة

if (result?.overallScore === undefined && !result?.error) {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'reviewCodeForSidebar:incomplete\_response', { result });

throw new Error("رد غير مكتمل من وحدة التحليل.");

}

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'reviewCodeForSidebar:success', { score: result.overallScore, hasPatch: !!result.patch });

return result;

} catch (e) {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'reviewCodeForSidebar:failure', {

error: e.message,

stack: e.stack,

scriptTextPreview: code.substring(0, 100) // تسجيل جزء من الكود للمساعدة في التصحيح

});

// إرجاع كائن خطأ متوافق مع ما يتوقعه الواجهة الأمامية (BuiltinDocsSidebar.html الموحد)

return {

overallScore: 0,

summary: "حدث خطأ أثناء المراجعة: " + e.message,

feedbackItems: [],

patch: null, // تأكيد إرجاع patch حتى لو كان خطأ

error: e.message // إضافة حقل error لتسهيل التعامل في الواجهة الأمامية

};

}

}

/\*\*

\* @function explainCodeWithGemini

\* @description

\* ترسل الكود إلى نموذج Gemini لشرح وظيفته.

\* @param {string} code - الكود البرمجي المراد شرحه.

\* @returns {object} - كائن رسالة نصية من Gemini.

\*/

function explainCodeWithGemini(code) {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'explainCodeWithGemini:start', { length: code.length });

try {

const prompt = `اشرح الكود التالي بالتفصيل بطريقة واضحة وموجزة، مع التركيز على وظيفته الرئيسية، المدخلات، والمخرجات، وأي مفاهيم مهمة. استخدم تنسيق Markdown:\n\n\`\`\`javascript\n${code}\n\`\`\`\n`;

// استخدام AI\_Core.ask لإرسال الطلب إلى Gemini

const result = GA\_AI\_Core.ask(prompt, "gemini-1.5-flash"); // يمكن استخدام نموذج flash للشرح السريع

if (result?.type === 'text' && result.content) {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'explainCodeWithGemini:success', { explanationLength: result.content.length });

return result;

} else {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'explainCodeWithGemini:no\_text\_response', { result });

return GA\_AI\_Dialogue.showError("خطأ", "لم يتمكن Gemini من شرح الكود بشكل نصي.");

}

} catch (e) {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'explainCodeWithGemini:failure', { error: e.message });

return GA\_AI\_Dialogue.showError("خطأ", `فشل شرح الكود: ${e.message}`);

}

}

/\*\*

\* @function generateUnitTestsWithGemini

\* @description

\* ترسل الكود إلى نموذج Gemini لتوليد اختبارات الوحدة له.

\* @param {string} code - الكود البرمجي المراد توليد اختبارات له.

\* @returns {object} - كائن رسالة نصية تحتوي على كود اختبارات الوحدة من Gemini.

\*/

function generateUnitTestsWithGemini(code) {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'generateUnitTestsWithGemini:start', { length: code.length });

try {

const prompt = `قم بتوليد اختبارات وحدة (Unit Tests) شاملة للكود التالي باستخدام إطار عمل اختبار مناسب لـ Google Apps Script (مثل QUnit أو مجرد دوال اختبار بسيطة). يجب أن تغطي الاختبارات الحالات الإيجابية والسلبية والحالات الحافة. قدم الكود كاملاً داخل كتلة كود JavaScript:\n\n\`\`\`javascript\n${code}\n\`\`\`\n`;

// استخدام AI\_Core.ask لإرسال الطلب إلى Gemini

const result = GA\_AI\_Core.ask(prompt, "gemini-1.5-pro"); // قد يتطلب نموذج أقوى لتوليد الكود

if (result?.type === 'text' && result.content) {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'generateUnitTestsWithGemini:success', { testsLength: result.content.length });

return result;

} else {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'generateUnitTestsWithGemini:no\_text\_response', { result });

return GA\_AI\_Dialogue.showError("خطأ", "لم يتمكن Gemini من توليد اختبارات الوحدة بشكل نصي.");

}

} catch (e) {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'generateUnitTestsWithGemini:failure', { error: e.message });

return GA\_AI\_Dialogue.showError("خطأ", `فشل توليد اختبارات الوحدة: ${e.message}`);

}

}

/\*\*

\* @function generateJSDocWithGemini

\* @description

\* ترسل الكود إلى نموذج Gemini لتوليد تعليقات JSDoc له.

\* @param {string} code - الكود البرمجي المراد توليد تعليقات JSDoc له.

\* @returns {object} - كائن رسالة نصية تحتوي على تعليقات JSDoc من Gemini.

\*/

function generateJSDocWithGemini(code) {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'generateJSDocWithGemini:start', { length: code.length });

try {

const prompt = `قم بتوليد تعليقات JSDoc كاملة ودقيقة لكل دالة في الكود التالي. يجب أن تتضمن التعليقات وصفًا للدالة، ومعاملاتها (مع النوع والوصف)، والقيمة المرجعة (مع النوع والوصف). ضع تعليقات JSDoc مباشرة فوق كل دالة. قدم الكود كاملاً مع التعليقات:\n\n\`\`\`javascript\n${code}\n\`\`\`\n`;

const result = GA\_AI\_Core.ask(prompt, "gemini-1.5-pro"); // قد يتطلب نموذج أقوى لتوليد تعليقات دقيقة

if (result?.type === 'text' && result.content) {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'generateJSDocWithGemini:success', { jsdocLength: result.content.length });

return result;

} else {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'generateJSDocWithGemini:no\_text\_response', { result });

return GA\_AI\_Dialogue.showError("خطأ", "لم يتمكن Gemini من توليد تعليقات JSDoc بشكل نصي.");

}

} catch (e) {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'generateJSDocWithGemini:failure', { error: e.message });

return GA\_AI\_Dialogue.showError("خطأ", `فشل توليد تعليقات JSDoc: ${e.message}`);

}

}

/\*\*

\* @function optimizePerformanceWithGemini

\* @description

\* ترسل الكود إلى نموذج Gemini لاقتراح تحسينات الأداء.

\* @param {string} code - الكود البرمجي المراد تحسين أدائه.

\* @returns {object} - كائن رسالة نصية تحتوي على اقتراحات تحسين الأداء من Gemini.

\*/

function optimizePerformanceWithGemini(code) {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'optimizePerformanceWithGemini:start', { length: code.length });

try {

const prompt = `حلل الكود التالي واقترح تحسينات محددة ومفصلة لأدائه. ركز على تقليل وقت التنفيذ، وتحسين استخدام الذاكرة، وأي ممارسات برمجية يمكن أن تجعل الكود أكثر كفاءة. قدم الاقتراحات في شكل قائمة نقطية أو خطوات واضحة. استخدم تنسيق Markdown:\n\n\`\`\`javascript\n${code}\n\`\`\`\n`;

const result = GA\_AI\_Core.ask(prompt, "gemini-1.5-pro"); // قد يتطلب نموذج أقوى للتحليل العميق

if (result?.type === 'text' && result.content) {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'optimizePerformanceWithGemini:success', { optimizationLength: result.content.length });

return result;

} else {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'optimizePerformanceWithGemini:no\_text\_response', { result });

return GA\_AI\_Dialogue.showError("خطأ", "لم يتمكن Gemini من اقتراح تحسينات الأداء بشكل نصي.");

}

} catch (e) {

GA\_Helpers.logOperation('GAssistant.Tools.CodeReview', 'optimizePerformanceWithGemini:failure', { error: e.message });

return GA\_AI\_Dialogue.showError("خطأ", `فشل اقتراح تحسينات الأداء: ${e.message}`);

}

}

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

return {

reviewCodeForSidebar: reviewCodeForSidebar,

explainCodeWithGemini: explainCodeWithGemini,

generateUnitTestsWithGemini: generateUnitTestsWithGemini,

generateJSDocWithGemini: generateJSDocWithGemini, // تصدير الدالة الجديدة

optimizePerformanceWithGemini: optimizePerformanceWithGemini // تصدير الدالة الجديدة

};

})(

GAssistant.AI.ScriptDoctor, // تبعية لوحدة طبيب السكربت

GAssistant.Helpers,

GAssistant.AI.Dialogue, // لرسائل الخطأ

GAssistant.AI.Core // تبعية جديدة لـ AI\_Core

);

// الدوال العامة (Global Functions) التي يمكن استدعاؤها مباشرة بواسطة google.script.run

// هذه الدوال ضرورية لتمكين google.script.run من استدعاء الدوال من HTML.

// يجب أن تكون هذه الدوال موجودة في ملف .gs يمكن الوصول إليه من مشروعك.

/\*\*

\* @function reviewCodeForSidebar

\* @description

\* دالة عامة تُستخدم بواسطة google.script.run لجلب تغذية راجعة حول الكود لعرضها في الشريط الجانبي.

\* @param {string} code - الكود البرمجي المراد مراجعته.

\* @param {string} reviewType - نوع المراجعة المطلوبة.

\* @returns {object} - كائن جاهز للعرض في الشريط الجانبي.

\*/

function reviewCodeForSidebar(code, reviewType) {

return GAssistant.Tools.CodeReview.reviewCodeForSidebar(code, reviewType);

}

/\*\*

\* @function explainCodeWithGemini

\* @description

\* دالة عامة تُستخدم بواسطة google.script.run لطلب شرح الكود من Gemini.

\* @param {string} code - الكود المراد شرحه.

\* @returns {object} - استجابة Gemini.

\*/

function explainCodeWithGemini(code) {

return GAssistant.Tools.CodeReview.explainCodeWithGemini(code);

}

/\*\*

\* @function generateUnitTestsWithGemini

\* @description

\* دالة عامة تُستخدم بواسطة google.script.run لطلب توليد اختبارات الوحدة من Gemini.

\* @param {string} code - الكود المراد توليد اختبارات له.

\* @returns {object} - استجابة Gemini.

\*/

function generateUnitTestsWithGemini(code) {

return GAssistant.Tools.CodeReview.generateUnitTestsWithGemini(code);

}

/\*\*

\* @function generateJSDocWithGemini

\* @description

\* دالة عامة تُستخدم بواسطة google.script.run لطلب توليد تعليقات JSDoc من Gemini.

\* @param {string} code - الكود المراد توليد تعليقات JSDoc له.

\* @returns {object} - استجابة Gemini.

\*/

function generateJSDocWithGemini(code) {

return GAssistant.Tools.CodeReview.generateJSDocWithGemini(code);

}

/\*\*

\* @function optimizePerformanceWithGemini

\* @description

\* دالة عامة تُستخدم بواسطة google.script.run لطلب اقتراحات تحسين الأداء من Gemini.

\* @param {string} code - الكود المراد تحسين أدائه.

\* @returns {object} - استجابة Gemini.

\*/

function optimizePerformanceWithGemini(code) {

return GAssistant.Tools.CodeReview.optimizePerformanceWithGemini(code);

}

// نفترض أن الدالة getCodeFromActiveCell موجودة في Code.gs أو ui.gs

// نفترض أن الدالة updateActiveCellWithCode موجودة في Code.gs أو ui.gs