/\*\*

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

\* @file ai\_dispatcher.gs

\* @module GAssistant.AI.Dispatcher

\* @description

\* مسؤول عن تحليل ردود Gemini API وتوجيهها.

\* يتعامل مع النصوص، استدعاءات الأدوات، والأخطاء.

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

\*/

var GAssistant = GAssistant || {};

GAssistant.AI = GAssistant.AI || {};

GAssistant.AI.Dispatcher = (() => {

'use strict';

// --- حقن التبعيات (النسخة الآمنة) ---

// نشير إلى الوحدات الكاملة لتجنب مشاكل ترتيب التحميل.

const Utils = GAssistant.Utils;

const ToolsCatalog = GAssistant.Tools.Catalog;

// سيتم استدعاء GAssistant.UI.Dialogue مباشرة عند الحاجة.

/\*\*

\* يحلل رد النموذج وينفذ الإجراء المناسب.

\* @param {object} apiResponse - الرد الكامل من Gemini API.

\* @returns {object} - كائن رسالة موحد للواجهة الأمامية.

\*/

function processApiResponse(apiResponse) {

Utils.log('AI.Dispatcher.processApiResponse: Start', { response: apiResponse });

// الوصول إلى Dialogue بشكل آمن

const Dialogue = GAssistant.UI.Dialogue;

const candidate = apiResponse?.candidates?.[0];

if (!candidate) {

Utils.error('AI.Dispatcher.processApiResponse: No valid candidate found in API response', apiResponse);

return Dialogue.createError('فشل استلام رد صالح من النموذج. قد تكون هناك مشكلة في الخدمة.');

}

// التحقق من سبب الإنهاء (مثل إعدادات الأمان)

if(candidate.finishReason && candidate.finishReason !== 'STOP' && candidate.finishReason !== 'TOOL\_USE') {

Utils.warn('AI.Dispatcher.processApiResponse: Non-standard finish reason', { reason: candidate.finishReason });

const safetyMessage = \_handleSafetyRatings(candidate.safetyRatings);

if (safetyMessage) return Dialogue.createError(safetyMessage);

}

const part = candidate.content?.parts?.[0];

if (!part) {

return Dialogue.createInfo("أنهى النموذج مهمته دون تقديم مخرجات إضافية.");

}

// 1. معالجة استدعاءات الأدوات (Function Calls)

if (part.functionCall) {

Utils.log('AI.Dispatcher.processApiResponse: Handling function call', { call: part.functionCall });

const { name, args } = part.functionCall;

const toolFunction = ToolsCatalog.getFunction(name);

if (toolFunction) {

// استخدام `executeSafely` لتنفيذ الأداة بأمان ومعالجة أي أخطاء داخلية.

const result = Utils.executeSafely(toolFunction, [args], `Executing tool: ${name}`);

// `executeSafely` يُرجع بالفعل كائن رسالة موحد (نجاح أو خطأ).

return result;

} else {

Utils.error(`AI.Dispatcher.processApiResponse: Tool '${name}' not found in catalog.`);

return Dialogue.createError(`حاول النموذج استدعاء أداة غير موجودة: ${name}`);

}

}

// 2. معالجة النص العادي أو JSON

if (part.text) {

Utils.log('AI.Dispatcher.processApiResponse: Handling text response', { length: part.text.length });

const trimmedText = part.text.trim();

// محاولة تحليل النص كـ JSON إذا كان يبدو كذلك

if ((trimmedText.startsWith('{') && trimmedText.endsWith('}')) || (trimmedText.startsWith('[') && trimmedText.endsWith(']'))) {

try {

const jsonData = JSON.parse(trimmedText);

Utils.log('AI.Dispatcher.processApiResponse: Parsed JSON response successfully.');

// تحويل كائن JSON إلى نص منسق للعرض

const formattedJson = JSON.stringify(jsonData, null, 2);

return Dialogue.createInfo(`تم استلام بيانات منظمة:\n\`\`\`json\n${formattedJson}\n\`\`\``);

} catch(e) {

// ليس JSON صالحًا، سيتم التعامل معه كنص عادي في السطر التالي.

Utils.warn('AI.Dispatcher.processApiResponse: Text looked like JSON but failed to parse.');

}

}

return Dialogue.createInfo(part.text);

}

// 3. حالة افتراضية

Utils.warn('AI.Dispatcher.processApiResponse: No recognizable content part found', { part });

return Dialogue.createInfo("تم استلام رد من النموذج ولكن لا يحتوي على محتوى قابل للعرض.");

}

/\*\*

\* يحلل تقييمات الأمان ويُرجع رسالة خطأ إذا تم حظر المحتوى.

\* @private

\*/

function \_handleSafetyRatings(safetyRatings) {

if (!safetyRatings) return null;

for (const rating of safetyRatings) {

if (rating.probability !== 'NEGLIGIBLE' && rating.probability !== 'LOW') {

Utils.warn('AI.Dispatcher: Content flagged by safety settings', { rating });

return `تم حظر الرد بسبب إعدادات الأمان المتعلقة بـ ${rating.category}.`;

}

}

return null;

}

return {

processApiResponse

};

})();