/\*\*

\* @fileoverview ملف SETTINGS\_PROVIDER.gs - v1.0

\* @module SETTINGS\_PROVIDER

\* @description

\* يوفر واجهة موحدة للوصول إلى إعدادات التطبيق.

\* يقوم بدمج الإعدادات الافتراضية مع تلك المخزنة في UserProperties و ScriptProperties.

\* يعتمد على: لا توجد تبعيات خارجية مباشرة (يعرف إعداداته الافتراضية داخلياً).

\*/

var SETTINGS\_PROVIDER = (function() {

'use strict';

// ⏱️ مدة صلاحية الكاش بالثواني (30 دقيقة)

const CACHE\_DURATION\_SECONDS = 1800;

// 📦 خدمة الكاش لـ Apps Script

const CACHE = CacheService.getDocumentCache();

/\*\*

\* @const {object} DEFAULT\_SETTINGS

\* @description

\* كائن يحتوي على جميع الإعدادات الافتراضية للنظام.

\* هذه الإعدادات هي الأساس الذي يتم دمج الخصائص المحفوظة فوقه.

\*/

const DEFAULT\_SETTINGS = {

ENVIRONMENT: "production",

DEBUG\_MODE: true,

SYSTEM\_LANGUAGE: "ar",

MASTER\_INSTRUCTIONS\_SHEET\_NAME: "التعليمات الرئيسية",

OPERATION\_LOG\_SHEET\_NAME: "سجل العمليات",

MAX\_LOG\_ENTRIES: 1000,

MAX\_LOG\_ENTRIES\_FOR\_CONTEXT: 5,

DEFAULT\_MODEL: "gemini-1.5-pro",

MODEL\_FOR\_SIMPLE\_TASKS: "gemini-1.5-flash",

ALLOWED\_MODELS: ["gemini-1.5-pro", "gemini-1.5-flash", "gemini-1.0-pro"],

API\_ENDPOINT: "https://generativelanguage.googleapis.com/v1beta/models/",

API\_KEYS\_NAMES: {

GEMINI: "GEMINI\_API\_KEY",

OPENAI: "OPENAI\_API\_KEY"

},

GENERATION\_CONFIG: {

temperature: 0.7,

topP: 0.95,

topK: 60,

maxOutputTokens: 2048,

responseMimeType: "text/plain",

},

SAFETY\_SETTINGS: [

{ category: "HARM\_CATEGORY\_HARASSMENT", threshold: "BLOCK\_NONE" },

{ category: "HARM\_CATEGORY\_HATE\_SPEECH", threshold: "BLOCK\_NONE" },

{ category: "HARM\_CATEGORY\_SEXUALLY\_EXPLICIT", threshold: "BLOCK\_NONE" },

{ category: "HARM\_CATEGORY\_DANGEROUS\_CONTENT", threshold: "BLOCK\_NONE" },

],

TOOLS\_ENABLED: true,

TOOL\_CONFIG: {

"functionCallingConfig": {

"mode": "AUTO"

},

"thinkingConfig": {

"enableThinking": true,

"thinkingModel": "gemini-1.5-flash",

"thinkingTemperature": 0.2,

"thinkingMaxOutputTokens": 512

}

},

SALES\_SHEET: "المبيعات",

EXPENSES\_SHEET: "المصروفات",

ACCOUNTS\_SHEET: "الحسابات",

CLIENTS\_SHEET: "العملاء",

VENDORS\_SHEET: "الموردين",

DAILY\_SUMMARY\_SHEET: "الملخص اليومي",

DATE\_FORMAT: "yyyy-MM-dd",

DEFAULT\_TAX\_RATE: 0.15,

PROTECTED\_RANGES: ["A1:Z1"],

MAX\_ENTRIES\_PER\_DAY: 100,

ALLOWED\_USERS: [],

DEFAULT\_CURRENCY: "SAR",

SIDEBAR\_FILE: "Sidebar",

UNIFIED\_SIDEBAR\_FILE: "BuiltinDocsSidebar",

PRIMARY\_HEADER\_COLOR: "#d9ead3",

HIGHLIGHT\_COLOR: "#fce8e6",

TEMPLATES: {

SALES: {

SHEET\_NAME: "المبيعات",

HEADERS: ["التاريخ", "المنتج", "الكمية", "سعر الوحدة", "الإجمالي", "العميل", "الوصف"]

},

EXPENSES: {

SHEET\_NAME: "المصروفات",

HEADERS: ["التاريخ", "الوصف", "المبلغ", "الفئة", "المورد"]

},

},

STAGING\_SHEET\_NAME: 'Developer Workshop',

TOOL\_FILE\_SUGGESTION\_PATTERNS: [

{ keywords: ['getactivespreadsheet', 'spreadsheetapp', 'createmenu', 'onopen', 'onsidebar'], file: 'ui.gs', weight: 10 },

{ keywords: ['getrange', 'setvalue', 'appendrow', 'sheet', 'range', 'getvalues', 'setvalues'], file: 'tools\_sheets.gs', weight: 9 },

{ keywords: ['revenue', 'expense', 'profit', 'currency', 'income', 'cost', 'ledger'], file: 'tools\_accounting.gs', weight: 8 },

{ keywords: ['logoperation', 'error', 'message', 'dialogue', 'cache'], file: 'helpers.gs', weight: 7 },

{ keywords: ['model', 'gemini', 'api', 'prompt', 'ask'], file: 'ai\_core.gs', weight: 7 },

{ keywords: ['context', 'session', 'remember', 'recall', 'memory'], file: 'ai\_memory.gs', weight: 6 },

{ keywords: ['message', 'show', 'alert', 'dialogue', 'prompt'], file: 'ai\_tools\_dialogue.gs', weight: 6 },

{ keywords: ['test', 'assert', 'expect', 'runtest'], file: 'tests.gs', weight: 10 },

{ keywords: ['reviewcode', 'patch', 'jsdoc', 'performance'], file: 'tools\_code\_review.gs', weight: 9 },

{ keywords: ['project', 'analyze', 'insights', 'files', 'scan'], file: 'tools\_project\_insights.gs', weight: 10 }

],

// TOOLS\_DEFINITIONS سيتم جلبها ديناميكيًا من TOOLS\_CATALOG في GAssistant.Config

TOOLS\_DEFINITIONS: []

};

// كائن لتخزين الإعدادات النشطة في الذاكرة بعد قراءتها من الخصائص أو الكاش

let \_activeSettings = null;

/\*\*

\* @function \_readSettingsFromProperties

\* @description

\* تقرأ الإعدادات من ScriptProperties و UserProperties وتدمجها مع الإعدادات الافتراضية.

\* @returns {object} - كائن الإعدادات المدمجة.

\*/

function \_readSettingsFromProperties() {

const scriptProps = PropertiesService.getScriptProperties().getProperties();

const userProps = PropertiesService.getUserProperties().getProperties();

const settings = JSON.parse(JSON.stringify(DEFAULT\_SETTINGS)); // نسخة عميقة من الافتراضيات

// دمج الخصائص من ScriptProperties (أولوية أعلى من الافتراضيات)

for (const key in scriptProps) {

if (settings.hasOwnProperty(key)) {

try {

settings[key] = JSON.parse(scriptProps[key]);

} catch (e) {

settings[key] = scriptProps[key];

}

}

}

// دمج الخصائص من UserProperties (أولوية أعلى من ScriptProperties)

for (const key in userProps) {

if (settings.hasOwnProperty(key)) {

try {

settings[key] = JSON.parse(userProps[key]);

} catch (e) {

settings[key] = userProps[key];

}

}

}

// ملاحظة: جلب TOOLS\_DEFINITIONS من TOOLS\_CATALOG يجب أن يتم في GAssistant.Config

// لأن SETTINGS\_PROVIDER لا يجب أن يعتمد على وحدات أخرى (للحفاظ على استقلاليته).

// سيتم تعيين TOOLS\_DEFINITIONS في GAssistant.Config عند تهيئته.

return settings;

}

/\*\*

\* @function getSettings

\* @description

\* تجلب الإعدادات النشطة من الكاش أو تقرأها من الخصائص وتخزنها في الكاش.

\* @returns {object} - كائن الإعدادات النشطة.

\*/

function getSettings() {

if (\_activeSettings) {

return \_activeSettings; // إرجاع الإعدادات المخزنة في الذاكرة

}

let cachedSettings = CACHE.get('activeSettings');

if (cachedSettings) {

\_activeSettings = JSON.parse(cachedSettings);

return \_activeSettings;

}

const settings = \_readSettingsFromProperties();

\_activeSettings = settings; // تخزين في الذاكرة

CACHE.put('activeSettings', JSON.stringify(settings), CACHE\_DURATION\_SECONDS);

return settings;

}

/\*\*

\* @function refreshSettingsCache

\* @description

\* تُجبر النظام على إعادة قراءة الإعدادات من الخصائص وتحديث الكاش.

\*/

function refreshSettingsCache() {

\_activeSettings = null; // مسح الإعدادات من الذاكرة

CACHE.remove('activeSettings'); // مسح الإعدادات من الكاش

getSettings(); // إعادة قراءة وتخزين

console.log("SETTINGS\_PROVIDER: Settings cache refreshed.");

}

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

return {

getSettings: getSettings,

refreshSettingsCache: refreshSettingsCache

};

})();