**Manuel d'Architecture : Usine à Tests**

**Version : 3.0** **Date : 2025-07-21**

**1. Vision et Contexte Historique**

Ce système a été conçu pour automatiser entièrement la création de questionnaires (Google Forms) et de leurs feuilles de calcul de traitement associées ("Kits"). L'objectif est de transformer une simple ligne de configuration en un système de test complet, robuste et flexible.

Le projet a connu trois grandes phases d'évolution :

* **Version 0 - Le Prototype Artisanal :** Un système "tout-en-un" où une seule feuille de calcul gérait un unique test de A à Z.
* **Version 1 - L'Usine Spécialisée :** Une séparation entre l'usine [MOTEUR] et le produit [TEMPLATE], mais chaque Kit contenait encore une logique de calcul qui lui était propre.
* **Version 2 - L'Usine Universelle (Actuelle) :** L'architecture où toute la logique de calcul est externalisée dans les données (format JSON) et traitée par un moteur universel. Cette version a été considérablement renforcée par :
  + La mise en place d'un **framework de débogage dédié** (LoggerLib).
  + La transformation du système d'e-mails en un **moteur dynamique piloté par la base de données**.
  + Le développement d'**interfaces utilisateur ergonomiques** pour la configuration et le retraitement.

**2. Principes Directeurs**

Le développement du projet respecte trois principes fondamentaux qui garantissent sa robustesse et sa maintenabilité.

* **Zéro Convention Implicite :** Tout paramètre doit être explicitement défini dans la configuration. Le code ne doit rien deviner.
  + *Exemple récent :* La résolution du bug des e-mails était due à une colonne de configuration manquante (Repondant\_Email\_Actif).
  + *Exemple récent :* Le lien entre un test et son e-mail de restitution est maintenant explicite via la colonne ID\_Gabarit\_Email\_Repondant dans [CONFIG].
* **Le JSON comme Source Unique de Vérité :** Toute la logique de calcul d'une question (type, profils, scores) est contenue dans un champ JSON dans la base de données. Le code n'est qu'un interpréteur.
  + *Exemple récent :* La résolution du bug des scores en anglais a démontré l'importance capitale que le contenu du JSON (libelle) soit parfaitement aligné avec les réponses du formulaire.
* **Le Code plus Intelligent que la Configuration :** Le code doit être conçu pour gérer des configurations variées de manière robuste.
  + *Exemple récent :* Le nouveau script Utilities.gs est un exemple parfait : il est capable de construire une infinité d'e-mails différents en interprétant les "gabarits" définis dans la base de données, sans qu'aucune modification de son code ne soit nécessaire.

**3. Architecture Globale des Composants**

Le système s'articule autour de **cinq** composants principaux :

* **[CONFIG] Usine à Tests :** Feuille de calcul servant de panneau de contrôle. C'est ici que l'utilisateur décrit les tests à générer via une interface de configuration (FormulaireUI.html et menu.gs). Son script a été mis à jour pour traduire les choix de l'utilisateur ("Niveau de contenu") en ID de gabarit d'e-mail.
* **[MOTEUR] Usine à Tests :** Le projet Google Apps Script qui lit une ligne de configuration dans [CONFIG] et orchestre la construction du Formulaire et du Kit de traitement.
* **[BDD] Tests & Profils :** La base de données centrale. Elle contient toutes les questions, les profils, les traductions, les pièces jointes et, désormais, les **gabarits d'e-mails**.
* **[TEMPLATE] Kit de Traitement V2 :** Le modèle de feuille de calcul autonome qui est copié pour chaque nouveau test. Son script interne gère le traitement des réponses et inclut l'**interface de retraitement** (RetraitementUI.html).
* **LoggerLib :** Une librairie Google Apps Script partagée et centralisée dont le seul rôle est de fournir des fonctions de journalisation pour le débogage.

**4. Le Workflow Utilisateur**

Le déploiement d'un nouveau test suit un processus simple en trois étapes :

1. **CONFIGURER :** Dans [CONFIG] Usine à Tests, utiliser le menu 🚀 Actions Usine -> Configurer un nouveau test....
2. **DÉPLOYER :** Dans [MOTEUR] Usine à Tests, lancer 🏭 Usine à Tests -> 🚀 Déployer un test de A à Z....
3. **ACTIVER :** Dans le "Kit de Traitement" nouvellement généré, cliquer une seule fois sur ⚙️ Actions du Kit -> Activer le traitement des réponses.

**5. Le Cœur du Système : La Configuration par les Données**

La flexibilité du système repose sur la manière dont la logique est stockée dans la base de données.

* **5.1. Le JSON pour les Questions :** La colonne Paramètres (JSON) dans les feuilles de questions de la BDD reste le pilier du moteur de calcul. Les modes validés sont QCU\_CAT, QRM\_CAT, et QCU\_DIRECT.
* **5.2. Les Gabarits d'E-mails (NOUVEAU) :** Le contenu et la structure des e-mails sont désormais entièrement pilotés par l'onglet Gabarits\_Emails dans la BDD. Le script injecte dynamiquement des variables (ex: {nom\_repondant}, {profilFinal}, {scoresData}) dans les textes du gabarit choisi.

**6. Fonctionnalités Clés**

* **Les Blocs Méta :** Les questions standards (Nom, Email, etc.) sont gérées comme des "blocs méta" sélectionnables depuis l'interface de configuration, offrant une grande flexibilité.
* **L'Interface de Retraitement (NOUVEAU) :** Une barre latérale (sidebar) permet de retraiter une réponse existante. L'utilisateur est d'abord invité à entrer un numéro de ligne, puis l'interface s'ouvre et permet de surcharger des paramètres (comme la langue de l'e-mail) avant de relancer le traitement.

**7. Structure de la Base de Données [BDD]**

* **Questions\_[Type]\_[Langue] :** Stocke les questions. Colonnes clés : ID, Type, Paramètres (JSON).
* **Questions\_META\_[Langue] :** Stocke les questions standards réutilisables.
* **Profils\_[Type]\_[Langue] :** Stocke les descriptions des profils. Structure de colonnes fixe attendue par le script de calcul.
* **traductions :** Gère la traduction des chaînes de texte statiques de l'interface et des e-mails.
* **Gabarits\_Emails (NOUVEAU) :** Stocke les modèles d'e-mails. Colonnes clés : ID\_Gabarit, Langue, Sujet, Corps\_..., Niveau\_Details\_Resultats, Niveau\_Pieces\_Jointes.
* **sys\_PiecesJointes :** Définit les fichiers à joindre aux e-mails.

**8. Dépannage (Troubleshooting)**

* **Problème :** Les scores détaillés sont absents d'un e-mail dans une langue spécifique (ex: anglais).
  + **Causes Possibles :**
    1. La feuille Profils\_[Type]\_[Langue] (ex: Profils\_Couleurs\_EN) est manquante ou mal nommée.
    2. La feuille Questions\_[Type]\_[Langue] a une incohérence entre la colonne Type et le "mode" dans le JSON.
    3. Les libelle dans le JSON de la feuille Questions\_[Type]\_[Langue] ne correspondent pas au texte des réponses du formulaire dans cette langue.
* **Problème :** Erreur La colonne 'ID\_Gabarit\_Email\_Repondant' n'est pas définie... dans les logs d'un Kit.
  + **Cause :** La colonne ID\_Gabarit\_Email\_Repondant est manquante, mal orthographiée, ou la cellule correspondante est vide dans l'onglet Paramètres Généraux de [CONFIG] Usine à Tests.
* **Problème :** La colonne ID\_Gabarit\_Email\_Repondant reste vide après avoir utilisé le formulaire "Configurer un nouveau test".
  + **Cause :** Le script menu.gs de [CONFIG] Usine à Tests est obsolète et ne contient pas la logique pour traduire le "Niveau de contenu" choisi par l'utilisateur en un ID de gabarit.

**9. Maintenance et Débogage**

Pour améliorer la stabilité, une architecture de débogage dédiée a été mise en place.

* **Composants :**
  + LoggerLib : Librairie externe centralisant les fonctions de journalisation.
  + MODE\_DEBUG\_ACTIF : Interrupteur (true/false) dans TraitementReponses.gs pour activer/désactiver les logs détaillés.
* **Procédure :**
  1. Ouvrir l'éditeur de script du Kit de Traitement à analyser.
  2. Dans TraitementReponses.gs, passer la constante MODE\_DEBUG\_ACTIF à true.
  3. Exécuter le test ou retraiter une réponse.
  4. Consulter les journaux détaillés dans le menu Exécutions.
  5. **Important :** S'assurer que l'interrupteur est sur false dans le [TEMPLATE] maître.