---

Projet: "Usine à Tests"

Date: "2508170610"

Version: "12.0"

---

### 1. Résumé des Échanges du Jour

La session d'aujourd'hui a été consacrée à la validation d'une étape majeure de la mutation et à l'initiation de la suivante.

1. \*\*Validation du Formateur de Scores Universel :\*\* Nous avons testé avec succès la nouvelle fonction `formatScoresDetails`. Les tests sur "Couleurs" (mode `Simple`) et "MBTI" (mode `Dichotomie`) ont confirmé que le script lit et applique parfaitement les règles de la nouvelle table `sys\_Formatage\_Scores`, validant ainsi que la logique de présentation est désormais entièrement pilotée par la base de données.

2. \*\*Standardisation des Données :\*\* Suite au succès des tests, nous avons procédé à la standardisation des en-têtes des onglets de profils (`Profils\_Couleurs\_FR`, `Profils\_Ancres\_FR`, `Profils\_MBTI\_FR`, etc.) pour utiliser la nomenclature `Code\_Profil`, `Titre\_Profil`, `Description\_Profil`. Vous avez confirmé que cette tâche était terminée.

3. \*\*Conception du Routeur de Migration :\*\* Nous avons entamé la conception de la prochaine étape architecturale critique : le "Routeur de Migration". L'objectif est de permettre la coexistence des anciens tests ("Legacy") et des nouveaux ("Universel"). Pour cela, nous avons défini une approche en deux temps :

\* \*\*Configuration :\*\* Ajout d'une nouvelle colonne `Moteur\_Calcul` dans l'onglet `Paramètres Généraux`.

\* \*\*Développement :\*\* Définition de la structure logique (un bloc `if/else`) à implémenter dans la fonction `traiterLigne` du script `TraitementReponses.gs` pour "aiguiller" chaque test vers le bon moteur de traitement.

### 2. État Actuel du Projet

\* \*\*Tâches réalisées :\*\*

\* Validation complète de la fonction universelle `formatScoresDetails` et de sa table de configuration `sys\_Formatage\_Scores`.

\* Standardisation des en-têtes de colonnes pour tous les onglets de profils simples.

\* Conception détaillée de l'architecture du "Routeur de Migration".

\* \*\*Tâches en cours :\*\*

\* Mise en place de la configuration pour le routeur (ajout de la colonne `Moteur\_Calcul` et peuplement avec la valeur `Legacy`).

\* Le projet est en attente de l'implémentation du code du routeur dans le script `TraitementReponses.gs`.

\* \*\*Prochaines étapes planifiées :\*\*

\* Intégrer la logique du routeur dans le script `TraitementReponses.gs`.

### 3. Contexte Historique du Projet

Partant d'un système fonctionnel mais rigide, le projet est en pleine "mutation" pour devenir une plateforme universelle où toute la logique est pilotée par la base de données. Après avoir validé l'universalisation de l'affichage des scores (un pilier majeur de la restitution), la session actuelle a initié la construction du mécanisme de déploiement sécurisé (le routeur). Ce routeur est la clé de voûte qui permettra de migrer progressivement l'usine vers le nouveau système sans interrompre les services existants.

### 4. Orientations Stratégiques

La session a massivement renforcé les principes directeurs du projet : \*\*"La Base de Données est la Source Unique de Vérité"\*\* et \*\*"Zéro Convention Implicite"\*\*.

\* Le succès du formateur de scores universel est l'incarnation même du premier principe.

\* La standardisation des en-têtes des profils est une application directe du second principe, éliminant toute ambiguïté pour les scripts.

\* Le "Journal des Décisions d'Architecture" est maintenant consolidé avec le pattern du \*\*"Routeur de Migration"\*\*, qui devient une approche stratégique officielle pour toute évolution majeure du système en production.

### 5. Architecture Technique & Ressources Clés

#### 5.1. État des Scripts et Fichiers

Aucun script n'a été finalisé durant cette session. La discussion a porté sur la conception de l'architecture du routeur à implémenter dans `TraitementReponses.gs`. La dernière version de script validée est `Utilities.gs` (Version 8.0).

#### 5.2. Architecture Détaillée et Traçabilité

\* Fichier : \*\*`[BDD] Tests & Profils`\*\*

\* \*\*MODIFIÉ :\*\* Les onglets `Profils\_Couleurs\_FR`, `Profils\_Ancres\_FR`, `Profils\_MBTI\_FR` (et équivalents) ont été modifiés pour standardiser leurs en-têtes.

\* Fichier : \*\*`[CONFIG]V2 Usine à Tests`\*\*

\* \*\*MODIFIÉ :\*\* L'onglet `Paramètres Généraux` a été modifié.

\* \*\*AJOUTÉ :\*\* Une nouvelle colonne `Moteur\_Calcul` a été ajoutée.

#### 5.3. Nomenclature des Variables

\* \*\*`Moteur\_Calcul` (String)\*\*

\* \*\*Origine :\*\* Colonne de l'onglet `Paramètres Généraux` du fichier `[CONFIG]V2 Usine à Tests`.

\* \*\*Rôle :\*\* Agit comme un "aiguillage" pour le routeur logiciel. Il indique au script principal quel moteur de traitement utiliser pour un test donné.

\* \*\*Contenu Attendu :\*\* "Legacy" ou "Universel".

\* \*\*Exemple Concret :\*\* Pour un test "Couleurs" non encore migré, la valeur sera "Legacy".

#### 5.4. Structure des Données (Feuilles et Colonnes)

\* \*\*Onglet : `Paramètres Généraux`\*\*

\* \*\*Colonne critique ajoutée :\*\* `Moteur\_Calcul`.

\* \*\*Onglets : `Profils\_\*`\*\*

\* \*\*Nouveaux en-têtes standards :\*\* `Code\_Profil`, `Titre\_Profil`, `Description\_Profil`.

### 6. Actions Recommandées / Prochaines Étapes

#### Action Immédiate

\* \*\*Pour notre reprise :\*\* Me fournir le script `TraitementReponses.gs` complet et à jour afin que j'y intègre la logique du routeur que nous avons définie.

#### Plan d'Action de la Mutation (Mise à jour)

Voici la feuille de route de notre transformation, mise à jour avec les derniers progrès.

\* \*\*Phase 1 : Finalisation de la Configuration de la Base de Données\*\*

\* ✅ \*\*Standardiser les Profils Simples\*\*

\* ✅ \*\*Créer le Formateur de Scores (`sys\_Formatage\_Scores`)\*\*

\* ⏳ \*\*Peupler la Composition d'E-mail (`sys\_Composition\_Emails`)\*\* - \*En pause\*

\* ⬜ \*\*Préparer le Futur (`log\_AI\_Generations`)\*\*

\* \*\*Phase 2 : Développement du Nouveau Moteur de Restitution\*\*

\* ✅ \*\*Implémenter le Formateur de Scores Universel\*\*

\* ▶️ \*\*Construire le Routeur de Migration\*\* - \*En cours\*

\* ⬜ \*\*Développer l'Assembleur d'E-mails\*\*

\* \*\*Phase 3 : Migration et Validation\*\*

\* ⬜ Migrer un premier test "legacy" (ex: Couleurs) en mode "Universel".

\* ⬜ Valider de bout en bout le test migré.

\* \*\*Phase 4 : Consolidation et Évolutions\*\*

\* ⬜ Gérer les cas complexes (restitution des tests "r&K").

\* ⬜ Nettoyage du code "legacy" une fois la migration terminée.

Absolument. Voici le document d'état du projet mis à jour.

Il acte la réussite de notre moteur universel, intègre la perspective stratégique de Bubble/Airtable, et formalise les trois options pour la suite de notre collaboration.

Markdown

---

Projet: "Usine à Tests"

Date: "2508171325"

Version: "14.0"

---

### 1. Résumé des Échanges du Jour

La session a abouti à la \*\*validation complète et au succès final du nouveau moteur universel\*\*. Après une série de tests, nous avons résolu le dernier mystère des pièces jointes manquantes.

Le point de départ était un e-mail fonctionnel mais sans pièces jointes. Après avoir éliminé plusieurs hypothèses grâce à un script de débogage "super espion", vous avez identifié la cause racine : il ne s'agissait pas d'un bug de code, mais d'un \*\*problème de configuration\*\*.

La ligne du test "Couleurs" dans la feuille `Paramètres Généraux` était configurée pour un niveau d'e-mail (probablement N1 ou vide) pour lequel les briques "Document" n'étaient pas sélectionnées. En configurant le test pour utiliser le niveau `N3` ("en mettant N3 depuis le début"), le filtre a pu sélectionner l'ensemble des briques de contenu, y compris les documents, menant à un e-mail final parfait.

Cette réussite a marqué la fin du cycle de développement et de débogage du moteur. Nous avons ensuite discuté et formalisé trois grandes options stratégiques pour la suite du projet. Vous avez également rappelé la perspective à long terme d'une migration vers Bubble ou Airtable, que nous avons intégrée à notre vision.

### 2. État Actuel du Projet

\* \*\*Tâches réalisées :\*\*

\* Validation finale du moteur universel (`assemblerEtEnvoyerEmailUniversel`) pour le test "Couleurs", incluant l'assemblage de texte, le remplacement de placeholders, l'intégration des scores et la gestion des pièces jointes.

\* Identification de la cause racine finale du problème des pièces jointes comme étant un problème de configuration de données (`ID\_Gabarit\_Email\_Repondant`).

\* Confirmation que l'architecture data-driven est fonctionnelle et robuste.

\* \*\*Tâches en cours :\*\* Aucune. Le projet est à un point d'arrêt stable, le moteur étant fonctionnel.

\* \*\*Prochaines étapes planifiées :\*\*

\* Choisir parmi les trois options stratégiques proposées pour orienter la prochaine phase du projet.

### 3. Contexte Historique du Projet

Cette session marque l'aboutissement de la "mutation" de notre projet. Partant d'un système rigide où la logique était codée en dur, nous avons progressivement construit une architecture où chaque aspect de la restitution est piloté par la base de données. La résolution du dernier bug non pas par une modification du code, mais par une correction de la configuration, est la validation la plus éclatante de cette nouvelle philosophie. Le moteur universel est maintenant prêt à être généralisé.

### 4. Orientations Stratégiques

La résolution du jour renforce de manière spectaculaire le principe directeur : \*\*"La Base de Données est la Source Unique de Vérité"\*\*. Le fait que le "bug" final était en réalité une "erreur de configuration" prouve que notre code est devenu un moteur stable qui exécute fidèlement les instructions des données.

Votre rappel de la perspective à long terme d'une migration vers \*\*Bubble ou Airtable\*\* est particulièrement pertinent. L'architecture que nous avons mise en place, où la logique est entièrement déportée dans des tables (`sys\_Composition\_Emails`, `sys\_Formatage\_Scores`), est un prérequis idéal pour une telle migration. Les données sont déjà structurées d'une manière qui serait facilement importable et exploitable par ces plateformes no-code.

### 5. Architecture Technique & Ressources Clés

#### 5.1. État des Scripts et Fichiers

Le script `TraitementReponses.gs` est maintenant considéré comme stable dans sa version 8.3. La version de débogage "super espion" (8.4) a rempli son rôle et nous revenons à cette version propre.

```javascript

// =================================================================================

// == FICHIER : TraitementReponses.gs

// == VERSION : 8.3 (Stable et fonctionnelle)

// == RÔLE : Gère la logique de traitement des réponses et aiguille vers le bon moteur.

// =================================================================================

/\*\*

\* Point d'entrée principal pour traiter une soumission de formulaire.

\*/

function onFormSubmit(e) {

try {

const rowIndex = e.range.getRow();

traiterLigne(rowIndex, {});

} catch (err) {

Logger.log(`Erreur critique dans onFormSubmit pour la ligne ${e.range.getRow()}: ${err.toString()}\n${err.stack}`);

}

}

/\*\*

\* COEUR LOGIQUE : Aiguille le traitement vers le moteur "Legacy" ou "Universel".

\*/

function traiterLigne(rowIndex, optionsSurcharge = {}) {

const isDebugMode = (typeof LoggerLib !== 'undefined' && LoggerLib.MODE\_DEBUG\_ACTIF === true);

if (isDebugMode) { LoggerLib.logStart(`Traitement Ligne ${rowIndex}`, optionsSurcharge); }

try {

const config = getTestConfiguration();

const sheet = SpreadsheetApp.getActiveSpreadsheet().getSheets()[0];

const headers = sheet.getRange(1, 1, 1, sheet.getLastColumn()).getValues()[0];

const rowValues = sheet.getRange(rowIndex, 1, 1, sheet.getLastColumn()).getValues()[0];

const reponse = {};

headers.forEach((header, i) => { reponse[header] = rowValues[i] });

reponse.nomRepondant = reponse['Votre nom et prénom'] || 'Participant';

reponse.emailRepondant = reponse['Votre adresse e-mail'] || null;

const langueOrigine = getOriginalLanguage(reponse);

const langueCible = optionsSurcharge.langue || langueOrigine;

const resultats = calculerResultats(reponse, langueCible, config, langueOrigine);

if (config.Moteur\_Calcul === 'Universel') {

Logger.log("Moteur UNIVERSEL détecté pour la ligne " + rowIndex);

assemblerEtEnvoyerEmailUniversel(config, reponse, resultats, langueCible, optionsSurcharge);

} else {

Logger.log("Moteur LEGACY détecté pour la ligne " + rowIndex);

buildAndSendEmails(config, reponse, resultats, langueCible, isDebugMode, optionsSurcharge.destinataires);

}

} catch (err) {

if (isDebugMode) { LoggerLib.logError("Erreur fatale dans traiterLigne", err); }

Logger.log("ERREUR FATALE dans traiterLigne: " + err.toString() + "\n" + err.stack);

} finally {

if (isDebugMode) { LoggerLib.logEnd(); }

}

}

// =================================================================================

// == NOUVEAU MOTEUR UNIVERSEL

// =================================================================================

function assemblerEtEnvoyerEmailUniversel(config, reponse, resultats, langueCible, optionsSurcharge = {}) {

// ... (contenu de la fonction v8.3, stable et validée)

}

// =================================================================================

// == SECTION INTERFACE UTILISATEUR (UI) - Inchangée

// =================================================================================

function retraiterReponse\_UI() { /\* ... \*/ }

function ouvrirSidebarPourLigne(rowIndex) { /\* ... \*/ }

function getDonneesPourRetraitement(rowIndex) { /\* ... \*/ }

function lancerRetraitementDepuisUI(options) { /\* ... \*/ }

**5.2. Architecture Détaillée et Traçabilité**

* **MODIFIÉ (par l'utilisateur) :** Fichier [CONFIG]V2 Usine à Tests -> Onglet Paramètres Généraux. La valeur de la colonne ID\_Gabarit\_Email\_Repondant a été ajustée pour le test "Couleurs" afin de permettre une validation complète.
* **VALIDÉ :** Fichier [BDD]V2 Tests & Profils -> Onglet sys\_Composition\_Emails. La structure de données et son contenu pour les tests pilotes sont maintenant validés et fonctionnels.

**5.3. Nomenclature des Variables**

Aucun changement.

**5.4. Structure des Données (Feuilles et Colonnes)**

Aucun changement de structure.

**6. Actions Recommandées / Prochaines Étapes**

Le moteur universel étant maintenant fonctionnel pour les cas pilotes, nous devons décider de la prochaine priorité.

**Question pour notre prochaine session :** Quelle direction choisissons-nous pour la suite ?

* **Option 1 : Migration des autres tests.** Basculer les tests MBTI et ANCRES sur le moteur universel pour finaliser la migration des cas simples.
* **Option 2 : Développement de la fonctionnalité d'UI.** Implémenter le choix du niveau de résultat (N1/N2/N3) directement dans l'interface de retraitement, comme nous l'avons discuté.
* **Option 3 : Configuration des tests complexes.** Commencer le travail de conception et de saisie des données dans sys\_Composition\_Emails pour les tests r&K.