---

Projet: Usine à Tests

Date: 2025-07-14

Version: 3.1

---

### 1. Résumé des Échanges du Jour

La session a été consacrée à un audit architectural complet du projet "Usine à Tests". L'objectif était de me donner un contexte total avant d'entamer la refonte vers la V2. Nous avons débuté par une décision stratégique majeure : l'\*\*abandon de la maintenance de la V1 pour concentrer 100% des efforts sur la V2\*\*.

Pour ce faire, vous m'avez fourni l'ensemble des scripts et documents historiques, ce qui a permis de reconstituer la chronologie complète du projet, de sa version "0" (un prototype monolithique) à sa version 1 (l'usine spécialisée). Cette analyse a permis de comprendre en profondeur les architectures existantes et de valider la pertinence de la stratégie V2.

Nous avons également défini et validé le format de ce "Document d'État V3" ainsi que le prompt qui servira à le générer, instaurant ainsi un workflow clair pour nos futures collaborations. Suite à une vérification, nous avons convenu d'enrichir ce document d'une section dédiée à la structure des données dans les feuilles de calcul.

### 2. État Actuel du Projet

\* \*\*Tâches réalisées :\*\*

\* Prise de décision stratégique : Focus exclusif sur la V2.

\* Audit complet et analyse de l'architecture des versions 0 et 1 via les scripts des projets `[CONFIG]`, `[MOTEUR]` et `[TEMPLATE] Kit`.

\* Définition et validation du format du "Document d'État V3" et de son prompt de génération.

\* Enrichissement du format du Document d'État avec une section sur la structure des données (colonnes).

\* \*\*Tâches en cours :\*\*

\* Génération de la version 3.1 de ce Document d'État.

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

\* Amorcer la refonte du `[MOTEUR]` pour y intégrer la logique V2 (lecture du JSON) en remplacement de la logique V1 (switch sur le type de question).

### 3. Contexte Historique du Projet

L'audit a révélé une évolution du projet en trois phases distinctes :

\* \*\*Version 0 - Le Prototype Artisanal :\*\* Un système "tout-en-un" où une seule feuille de calcul gérait le test "Couleurs" de A à Z (BDD, génération de formulaire, traitement des réponses).

\* \*\*Version 1 - L'Usine Spécialisée :\*\* Une évolution majeure séparant l'usine (`[MOTEUR]`) du produit (`[TEMPLATE] Kit`). L'usine industrialise la \*création\* des tests, mais chaque Kit généré contient encore une logique de calcul spécifique à un type de test (`Logique\_Couleurs.js`, `Logique\_MBTI.js`, etc.).

\* \*\*Version 2 - L'Usine Universelle (Cible) :\*\* L'architecture visée, où la logique de calcul est extraite du code et définie dans les données via des paramètres `JSON`. Un `Logique\_Universel.gs` remplace tous les scripts de logique spécialisés.

### 4. Orientations Stratégiques

La décision du jour d'abandonner la V1 renforce la stratégie V2. L'analyse des architectures V0 et V1 a mis en évidence la nécessité absolue des principes définis pour la V2 :

\* \*\*Zéro Convention Implicite :\*\* La V1 repose sur des noms de questions en dur (`Nom Complet`, `Adresse Email`), ce que la V2 doit éliminer.

\* \*\*Le JSON comme Source Unique de Vérité :\*\* La V1 utilise un `switch` sur un `TypeQuestion`. La V2 s'appuiera exclusivement sur le `JSON` pour définir la logique, la rendant infiniment plus flexible.

\* \*\*Le Code plus Intelligent que la Configuration :\*\* La V1 est fragile. La V2 doit intégrer des vérifications systématiques pour ne jamais planter à cause d'une mauvaise configuration.

\* \*\*Isoler les Services Externes :\*\* Le code de la V0 montre une gestion des erreurs basique pour `MailApp`. La V2 doit généraliser l'isolation des appels externes pour une robustesse maximale.

### 5. Architecture Technique & Ressources Clés

#### 5.1. État des Scripts et Fichiers

\*(Cette section listera les scripts complets qui auront été modifiés au cours d'une session. Pour cette session initiale, les scripts ont été analysés mais pas modifiés.)\*

#### 5.2. Architecture Détaillée et Traçabilité

\*\*Projet : `[CONFIG] Usine à Tests`\*\*

\* \*\*Rôle :\*\* Panneau de commande pour configurer de nouveaux tests.

\* \*\*Fichiers :\*\*

\* `Menu.js` : Côté serveur, gère la logique d'affichage et de traitement du formulaire.

\* `FormulaireUI.html` : L'interface utilisateur (sidebar) pour la saisie des configurations.

\* \*\*Dépendances Sheets :\*\* `Paramètres Généraux`, `sys\_ID\_Fichiers`, `sys\_Options\_Parametres`.

\*\*Projet : `[MOTEUR] Usine à Tests` (Architecture V1 Identifiée)\*\*

\* \*\*Rôle :\*\* Usine qui lit `[CONFIG]` et génère les Kits de test.

\* \*\*Fichiers :\*\*

\* `Interface.js` : Crée le menu pour lancer la génération.

\* `Moteur.js` : Orchestre la création (copie du template, création du form, etc.).

\* `Utils.js` : Fonctions de support (lecture config, création d'items de form V1).

\* \*\*Logique Identifiée :\*\* Architecture V1. La création de question (`creerItemFormulaire`) se base sur un `switch` sur le type de question, sans lecture de JSON.

\*\*Projet : `[TEMPLATE] Kit de Traitement` (Architecture V1 Identifiée)\*\*

\* \*\*Rôle :\*\* Modèle autonome pour chaque test, gérant le calcul des résultats.

\* \*\*Fichiers :\*\*

\* `TraitementReponses.js` : Aiguilleur principal qui se déclenche sur `onFormSubmit`.

\* `Logique\_\*.js` : Fichiers spécialisés contenant la logique de calcul pour chaque type de test (`Logique\_Couleurs`, `Logique\_MBTI`, etc.).

\* `Utilities.js` : Fonctions de support (lecture de sa propre config, envoi d'emails).

\* \*\*Logique Identifiée :\*\* Architecture V1. Un `switch` sur `config.Type\_Test` dans `TraitementReponses.js` appelle le script spécialiste adéquat.

#### 5.3. Nomenclature des Variables

\* \*\*Objet : `formObject`\*\*

\* \*\*Nom Précis :\*\* `formObject`

\* \*\*Type :\*\* `Object`

\* \*\*Origine :\*\* Construit côté client dans `FormulaireUI.html` et passé à la fonction `processNewTestConfiguration` dans `Menu.js`.

\* \*\*Contenu :\*\* Contient toutes les valeurs saisies dans le formulaire de configuration.

\* \*\*Exemple :\*\* `{ titre: "Test Couleurs", type: "Couleurs", nbQuestions: 10, repondantActif: true, ... }`

\* \*\*Objet : `config`\*\*

\* \*\*Nom Précis :\*\* `config`

\* \*\*Type :\*\* `Object`

\* \*\*Origine :\*\* Retourné par `getConfigurationFromRow(rowIndex)` dans `Utils.js` (Moteur). Lu par `lancerCreationSysteme` dans `Moteur.js`.

\* \*\*Contenu :\*\* Contient les paramètres d'une ligne de commande spécifique.

\* \*\*Exemple :\*\* `{ statut: "En construction", typeTest: "Couleurs", titreFormulaireUtilisateur: "Test Couleurs", ... }`

\* \*\*Objet : `resultats`\*\*

\* \*\*Nom Précis :\*\* `resultats`

\* \*\*Type :\*\* `Object`

\* \*\*Origine :\*\* Retourné par les fonctions spécialistes (ex: `calculerResultatCouleurs`) dans le Kit.

\* \*\*Contenu :\*\* Structure standardisée contenant le résultat d'un test.

\* \*\*Exemple :\*\* `{ profilFinal: "Rouge", titreProfil: "Le Compétiteur", descriptionProfil: "...", detailsScores: "..." }`

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

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

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

\* \*\*Rôle :\*\* Carnet de commandes des tests.

\* \*\*Colonnes Clés utilisées par les scripts :\*\* `Id\_Unique`, `Titre\_Formulaire\_Utilisateur`, `Statut`, `Type\_Test`, `ID\_Dossier\_Cible`, `nbQuestions`, `Repondant\_Email\_Actif`, `Repondant\_Quand`, `Repondant\_Contenu`, `Patron\_Email\_Mode`, `Patron\_Quand`, `Patron\_Contenu`, `Patron\_Email`, `Formateur\_Email\_Actif`, `Formateur\_Quand`, `Formateur\_Contenu`, `Formateur\_Email`, `Developpeur\_Email`, `ID\_Formulaire\_Cible`, `ID\_Sheet\_Cible`.

\* \*\*Onglet : `sys\_ID\_Fichiers`\*\*

\* \*\*Rôle :\*\* Stockage centralisé des ID de fichiers.

\* \*\*Colonnes Clés :\*\* `CléID` (implicite, col A), `ID` (implicite, col B).

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

\* \*\*Onglet : `Questions\_[Type]\_[Langue]`\*\*

\* \*\*Rôle :\*\* Base de données des questions.

\* \*\*Colonnes Clés (V1) :\*\* `ID`, `TypeQuestion`, `Titre`, `Options`, `Logique`, `Description`.

\* \*\*Colonnes Clés (V2) :\*\* `ID`, `TypeQuestion`, `Titre`, `Paramètres (JSON)`.

\* \*\*Onglet : `Profils\_[Type]\_[Langue]`\*\*

\* \*\*Rôle :\*\* Base de données des profils de résultats.

\* \*\*Colonnes Clés :\*\* `Code\_Profil`, `Titre\_Profil`, `Description\_Profil`.

\* \*\*Onglet : `sys\_PiecesJointes`\*\*

\* \*\*Rôle :\*\* Configuration des pièces jointes pour les e-mails.

\* \*\*Colonnes Clés :\*\* `Type\_Test`, `Profil\_Code`, `Email\_Niveau`, `Langue`, `ID\_Fichier\_Drive`.

\* \*\*Fichier : Kit de Traitement (Généré)\*\*

\* \*\*Onglet : `Réponses au formulaire 1`\*\* (Nom par défaut)

\* \*\*Rôle :\*\* Réception des données brutes du Google Form.

\* \*\*Colonnes Clés (lues par les scripts V1) :\*\* `Horodatage` (implicite), `Nom Complet / Full Name`, `Adresse Email`, `Langue / Language`, plus les colonnes correspondant aux titres de chaque question.

### 6. Actions Recommandées / Prochaines Étapes

\* \*\*Action Immédiate :\*\* Sauvegarder ce document `Document d'État V3.1`.

\* \*\*Pour la prochaine session :\*\* Commencer la refonte V2. L'objectif prioritaire sera de remplacer le code de `Moteur.js` et `Utils.js` (du projet Moteur) par la logique V2 capable de lire les onglets de la BDD et de créer les questions en se basant sur la colonne `Paramètres (JSON)`, rendant ainsi la fonction `creerItemFormulaire` universelle.