

OK. Pour envoyer les informations du formulaire vers une feuille Google Sheets, nous devons modifier le code JavaScript pour qu'il envoie les données à un script spécifique (un Google Apps Script) que vous devrez créer et lier à votre feuille de calcul. Je vais mettre à jour le code HTML/JavaScript pour inclure cette logique d'envoi. Ensuite, je vous fournirai les instructions et un exemple de Google Apps Script que vous devrez mettre en place de votre côté.

1. Code HTML/JavaScript Mis à Jour

Voici la page avec le JavaScript modifié pour envoyer les données.

Changements importants dans le JavaScript:

1. **SCRIPT_URL:** Une constante SCRIPT_URL a été ajoutée. Vous devez remplacer "VOTRE_URL_GOOGLE_APPS_SCRIPT_ICI" par l'URL que vous obtiendrez après avoir déployé votre Google Apps Script (voir étape 2).
2. **Gestionnaire submit:**
 - Il utilise maintenant fetch() pour envoyer une requête POST à votre SCRIPT_URL.
 - Les données du formulaire sont récupérées avec new FormData() et converties.
 - Un champ timestamp est ajouté automatiquement pour savoir quand la commande a été passée.
 - Le mode: 'no-cors' est utilisé pour simplifier l'interaction avec Google Apps Script depuis un site externe sans configuration CORS complexe. Cela signifie que le script JavaScript ne pourra pas lire la réponse exacte du script Google, mais il saura si la requête a été envoyée.
 - Des messages de succès ou d'erreur sont affichés dans une nouvelle div (#formStatus) sous le bouton.
 - Un indicateur de chargement (spinner) est ajouté au bouton pendant l'envoi.

2. Instructions pour Google Apps Script

Vous devez créer un script lié à votre feuille Google Sheets pour recevoir les données :

1. **Créez une Google Sheet :** Allez sur sheets.google.com et créez une nouvelle feuille de calcul. Nommez-la (par exemple, "Commandes Livre SSIAP3").
2. **Nommez les colonnes :** Dans la première ligne de votre feuille, mettez les en-têtes correspondant aux champs du formulaire, plus un pour l'horodatage. Par exemple : Timestamp | Prénom | Nom | Email | Téléphone | Adresse | Méthode Paiement
3. **Ouvrez l'éditeur de script :** Dans votre feuille de calcul, allez dans Extensions > Apps Script.
4. **Collez le code suivant :** Effacez tout code existant dans l'éditeur et collez le code ci-dessous :

JavaScript

```
/**
```

```
 * Fonction déclenchée par une requête POST (typiquement depuis un formulaire HTML).
```

```
 * Ajoute les données reçues comme une nouvelle ligne dans la feuille active.
```

```
 *
```

```
 * @param {Object} e L'objet événement contenant les paramètres de la requête.
```

```

*/
function doPost(e) {
    // Verrouiller le script pour éviter les problèmes d'accès concurrents
    var lock = LockService.getScriptLock();
    lock.waitLock(30000); // Attendre jusqu'à 30 secondes

    try {
        // Récupérer la feuille de calcul active (celle où le script est lié)
        var sheet = SpreadsheetApp.getActiveSheet();

        // --- Récupérer les données du formulaire depuis les paramètres de la requête ---
        // Les noms (e.g., 'firstName') doivent correspondre aux attributs 'name' des inputs
        HTML
        var timestamp = e.parameter.timestamp || new Date().toLocaleString('fr-FR'); //
        Horodatage (avec fallback)
        var firstName = e.parameter.firstName || "";
        var lastName = e.parameter.lastName || "";
        var email = e.parameter.email || "";
        var phone = e.parameter.phone || "";
        var address = e.parameter.address || "";
        var paymentMethod = e.parameter.paymentMethod || "";

        // Créer un tableau représentant la nouvelle ligne à ajouter
        // L'ordre DOIT correspondre à l'ordre des colonnes dans votre feuille Sheets
        var newRow = [
            timestamp,
            firstName,
            lastName,
            email,
            phone,
            address,
            paymentMethod
        ];

        // Ajouter la nouvelle ligne à la fin de la feuille
        sheet.appendRow(newRow);

        // Retourner une réponse de succès (même si 'no-cors' l'ignore côté client)
        // Utile pour le débogage direct de l'URL du script
        return ContentService
            .createTextOutput(JSON.stringify({ "result": "success", "row": newRow }))
            .setMimeType(ContentService.MimeType.JSON);
    }
}

```

```

} catch (error) {
    // En cas d'erreur, l'enregistrer dans les logs Apps Script pour le débogage
    Logger.log('Erreur dans doPost: ' + error);

    // Retourner une réponse d'erreur
    return ContentService
        .createTextOutput(JSON.stringify({ "result": "error", "error": error.toString() }))
        .setMimeType(ContentService.MimeType.JSON);
} finally {
    // Toujours libérer le verrou
    lock.releaseLock();
}
}

// --- OPTIONNEL: Fonction pour tester l'ajout de données manuellement ---
function testAddRow() {
    var sheet = SpreadsheetApp.getActiveSheet();
    sheet.appendRow([new Date().toLocaleString('fr-FR'), "Test", "Utilisateur",
        "test@example.com", "12345", "Adresse Test", "paypal"]);
    Logger.log("Ligne de test ajoutée.");
}

```

5. **Enregistrez le script** : Cliquez sur l'icône de disquette (Enregistrer le projet) et donnez-lui un nom (par exemple, "Script Commandes SSIAP3").
6. **Déployez le script** :
 - Cliquez sur le bouton bleu **"Déployer"** en haut à droite.
 - Choisissez **"Nouveau déploiement"**.
 - Cliquez sur l'icône d'engrenage à côté de "Sélectionner le type" et choisissez **"Application Web"**.
 - Dans la description, mettez quelque chose comme "Réception commandes livre SSIAP3".
 - Pour **"Exécuter en tant que"**, laissez **"Moi ([adresse e-mail supprimée])"**.
 - Pour **"Qui a accès"**, choisissez **"N'importe qui"**. C'est crucial pour que votre formulaire HTML puisse appeler le script sans que l'utilisateur ait besoin d'être connecté à Google.
 - Cliquez sur **"Déployer"**.
7. **Autorisez le script** : Google vous demandera d'autoriser le script à accéder à votre feuille de calcul. Cliquez sur **"Autoriser l'accès"**, choisissez votre compte Google, cliquez sur **"Paramètres avancés"** (si un avertissement apparaît), puis sur **"Accéder à [Nom de votre script] (non sécurisé)"** et enfin sur **"Autoriser"**.
8. **Copiez l'URL de l'application Web** : Une fois le déploiement réussi, une fenêtre apparaîtra avec l'URL de l'application Web. Copiez cette URL.

9. **Collez l'URL dans le code HTML :** Retournez à votre code HTML (dans l'éditeur ici) et remplacez "VOTRE_URL_GOOGLE_APPS_SCRIPT_ICI" dans la variable SCRIPT_URL par l'URL que vous venez de copier.

Maintenant, lorsque quelqu'un remplira et soumettra le formulaire sur votre page, le JavaScript enverra les données à votre Google Apps Script, qui les ajoutera à votre feuille Google Sheets.