

Créer un formulaire web avec Google Sheets & Apps Script

Ce document vous guidera à travers les étapes essentielles pour créer un formulaire web fonctionnel, capable de collecter des données directement dans une feuille Google Sheets. Nous explorerons comment Google Apps Script fait le pont entre votre formulaire HTML et le stockage des données, rendant ce processus accessible même aux développeurs débutants. Vous apprendrez à préparer votre feuille de calcul, à écrire le code nécessaire et à déployer votre solution pour une utilisation pratique.

Création d'un formulaire web avec Google Sheets et Apps Script

Coolerl Ngdlerlaxj Hamn | Date: 6.36 11.9

🇮🇳 🏠 📄 📊 76e 22.203 🌐 🌐

🏠 Project Name	🏠 Colgo Linemas	🏠 Base Date
📅 Présentik'iva Nelonc	📅 Correction des données	📅 Due Date
📅 Talk Ujestrloa	📅 Drogel&erne	📅 Due Die
📅 Task /Description	📅 Desk des lectione	📅 Due Toe
📅 Task Destination	📅 Des Desl&ermei	
📅 Duk Des Date	📅 Die Polistrgeets	
📅 Due Deccomers	📅 Des Correctening	

6.8	Task Date	22.10	Task
7.0	Pod hatur	201.10	Task
6.8	Paiz Tang	23.1.10	Task
2.8	Yarjo lbe	30.1.10	Task
2.0	Pod Coat	20.1.10	Task
6.8	Pod Coat	20.1.10	Task

Introduction à Google Apps Script pour la collecte de données

Google Apps Script est une plateforme de développement basée sur JavaScript qui permet d'étendre les fonctionnalités des applications Google Workspace, y compris Google Sheets. C'est un outil puissant pour automatiser des tâches, intégrer différents services Google et, comme nous le verrons, créer des applications web simples mais efficaces comme des formulaires de collecte de données.

L'avantage principal d'utiliser Apps Script pour un formulaire web est qu'il élimine le besoin d'un serveur externe ou d'une base de données complexe. Toutes les données sont directement stockées dans Google Sheets, qui agit comme votre base de données. Cela simplifie considérablement la gestion et l'analyse des informations collectées.

Automatisation Facile

Exécutez des scripts pour automatiser les tâches répétitives au sein de Google Workspace.

Intégration Transparente

Connectez Google Sheets à d'autres services comme Gmail, Docs, ou Calendar.

Déploiement Simplifié

Transformez des scripts en applications web accessibles via une URL unique.

Étape 1 : Préparer votre feuille de calcul Google Sheets

Avant de pouvoir collecter des données, il est crucial de structurer correctement votre feuille de calcul Google Sheets. Cette feuille servira de "base de données" pour toutes les réponses de votre formulaire. Une bonne préparation garantit que vos données seront organisées et faciles à analyser.

Création et nomination de la feuille

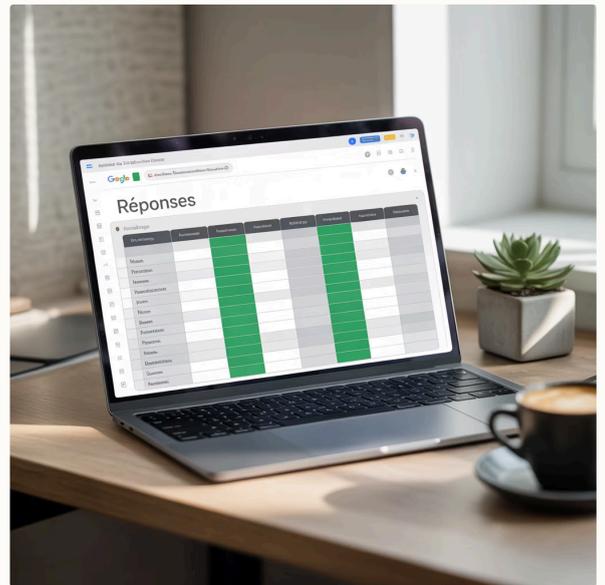
- Ouvrez [Google Sheets](#) et créez une nouvelle feuille de calcul.
- Renommez la première feuille (par défaut "Feuille 1") en "Réponses". Ce nom est important car il sera référencé dans votre code Apps Script.
- Assurez-vous que cette feuille est vierge avant de définir les en-têtes.

Définition des en-têtes de colonnes

Dans la première ligne (ligne 1) de votre feuille "Réponses", entrez les noms de colonnes qui correspondent aux champs de votre futur formulaire. Ces en-têtes doivent être descriptifs et refléter le type d'information que vous collecterez. Par exemple :

- **Horodatage** : pour enregistrer automatiquement la date et l'heure de chaque soumission.
- **Nom** : pour le nom de l'utilisateur.
- **Prénom** : pour le prénom de l'utilisateur.
- **Email** : pour l'adresse e-mail.
- **Question** : ou "Message" pour le contenu principal de la soumission.

Chaque en-tête doit être dans une cellule distincte (A1, B1, C1, etc.).



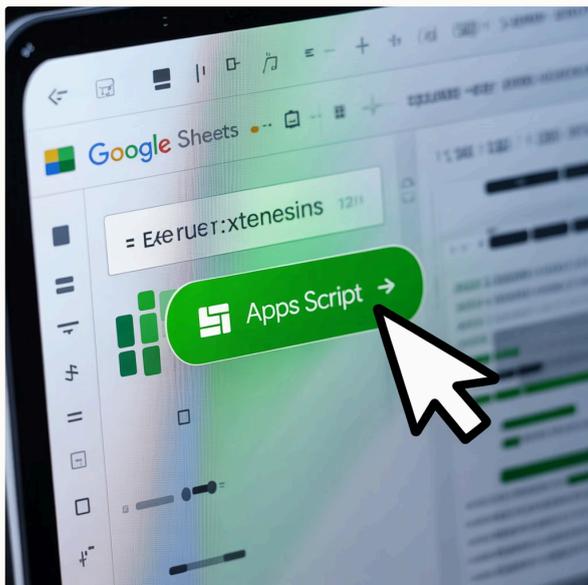
Exemple de feuille Google Sheets prête à recevoir les données du formulaire.

Étape 2 : Ouvrir l'éditeur Google Apps Script

Avec votre feuille Google Sheets préparée, la prochaine étape consiste à accéder à l'environnement de développement où vous écrirez le code de votre formulaire. Cet éditeur est intégré à Google Workspace et est accessible directement depuis votre feuille de calcul.

Accéder à l'éditeur

1. Dans votre feuille Google Sheets, cliquez sur le menu "**Extensions**" en haut de l'écran.
2. Passez votre curseur sur "**Apps Script**" et cliquez dessus. Cela ouvrira une nouvelle fenêtre ou un nouvel onglet avec l'éditeur Apps Script.



Nommer votre projet Script

Une fois l'éditeur ouvert, il vous sera demandé de donner un nom à votre projet. C'est une bonne pratique de lui donner un nom descriptif et pertinent. Par exemple, FormulaireWeb, CollecteDonnées, ou MonPremierFormulaire.

Ce nom aidera à identifier votre script dans votre tableau de bord Apps Script et à le distinguer d'autres projets que vous pourriez créer à l'avenir. Le nom peut être modifié ultérieurement si nécessaire, mais il est préférable de le définir correctement dès le début.

Conseil : Si vous ne nommez pas le projet immédiatement, il sera sauvegardé sous un nom générique comme "Projet sans titre". Vous pouvez le renommer en cliquant sur le nom actuel dans le coin supérieur gauche de l'éditeur.

Étape 3 : Écrire le code du formulaire (HTML et Apps Script)

Le cœur de votre formulaire réside dans le code. Il se compose de deux parties principales : le code HTML qui définit l'apparence et les champs du formulaire, et le code Apps Script (JavaScript) qui gère la logique côté serveur, y compris l'enregistrement des données dans Google Sheets.

1

Créer le fichier HTML (index.html)

Dans l'éditeur Apps Script, cliquez sur **Fichier** > **Nouveau** > **Fichier HTML**. Nommez-le `index.html`. Ce fichier contiendra la structure de votre formulaire web.

```
<!DOCTYPE html>
<html>
<head>
<base target="_top">
<style>/* Styles CSS ici */</style>
</head>
<body>
<h1>Formulaire de contact</h1>
<form id="myForm">
<label for="nom">Nom :</label><br>
<input type="text" id="nom"
name="nom" required><br><br>

<label for="email">Email :</label><br>
<input type="email" id="email"
name="email" required><br><br>

<label for="message">Message :</label>
<br>
<textarea id="message"
name="message" rows="5" required>
</textarea><br><br>

<input type="submit" value="Envoyer">
</form>
<div id="result"></div>

<script>
```

2

Écrire le code Apps Script (Code.gs)

Dans le fichier `Code.gs` (créé par défaut), insérez le code suivant. Il y a deux fonctions clés : `doGet()` qui sert le fichier HTML, et `processForm()` qui reçoit les données du formulaire et les écrit dans Google Sheets.

```
function doGet() {
  return
  HtmlService.createTemplateFromFile('in
dex').evaluate();
}

function processForm(data) {
  const sheet =
  SpreadsheetApp.getActiveSpreadsheet().
  getSheetByName("Réponses");
  sheet.appendRow([new Date(),
  data.nom, data.email, data.message]);
  return "
"
```

Vue d'ensemble : Formulaire Avant et Après soumission

Pour mieux comprendre l'interaction entre le formulaire HTML et Google Sheets, voici une illustration du formulaire web avant d'être rempli et l'état de la feuille de calcul après une soumission réussie.

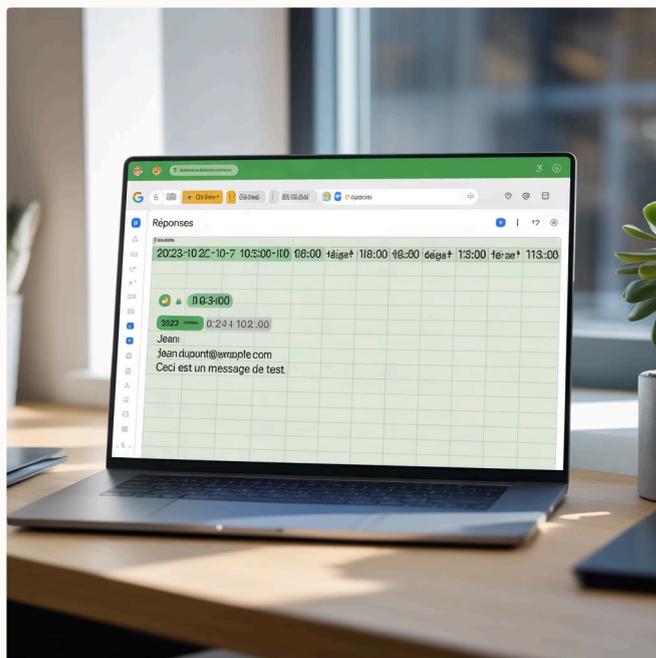
Formulaire web (Avant soumission)



A screenshot of a web browser window displaying a contact form titled "CONTACT US". The form contains three input fields: "Nom", "Email", and "Message". Below the fields is a green "Envoyer" button. The browser's address bar shows "CONTACT US". At the bottom left, there is a copyright notice "© 2024 Your Company" and at the bottom right, a link for "Privacy policy".

C'est l'interface que vos utilisateurs verront. Chaque champ correspondra à une colonne dans votre feuille Google Sheets.

Feuille Google Sheets (Après soumission)

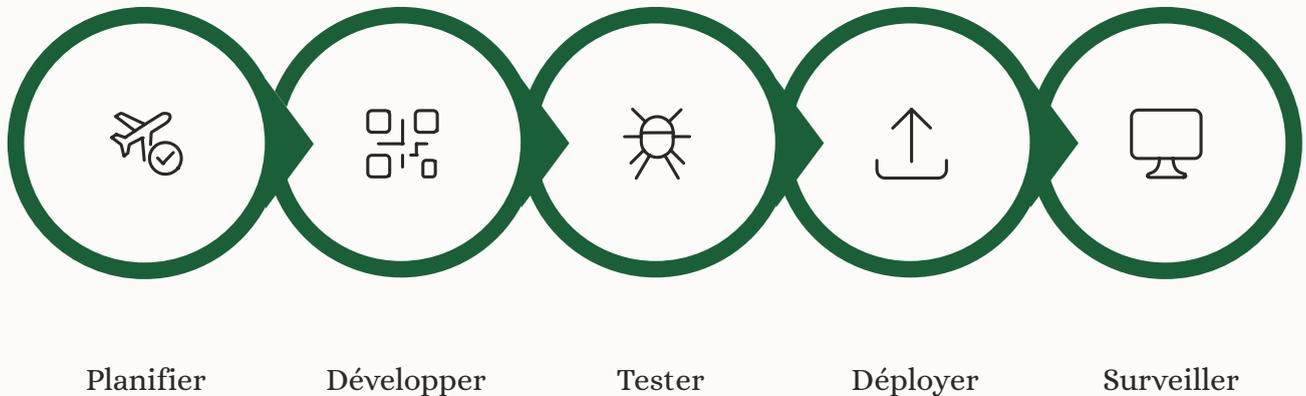


Une fois le formulaire soumis, une nouvelle ligne est automatiquement ajoutée à votre feuille "Réponses", avec les données correspondantes et un horodatage.

Cette automatisation, rendue possible par Google Apps Script, est un gain de temps considérable et assure une collecte de données précise et centralisée.

Étape 4 : Déployer votre formulaire web

Une fois votre code HTML et Apps Script écrit, l'étape suivante consiste à déployer votre projet en tant qu'application web. C'est ce déploiement qui générera l'URL publique de votre formulaire.



1. **Nouveau Déploiement** : Dans l'éditeur Apps Script, cliquez sur le bouton "**Déployer**" (souvent une icône de fusée ou "Déployer") puis sélectionnez "**Nouveau déploiement**".
2. **Type de Déploiement** : Cliquez sur l'icône de la roue dentée à côté de "Sélectionner le type" et choisissez "**Application Web**".
3. **Configuration** :
 - **Description du déploiement** : (facultatif) Entrez une description, par ex. "Formulaire de contact initial".
 - **Exécuter en tant que** : Sélectionnez "**Moi**". Cela signifie que le script s'exécutera sous votre identité Google.
 - **Accessible à** : Choisissez "**Toute personne disposant du lien**". Cela rendra votre formulaire accessible à quiconque possède l'URL, sans nécessiter d'authentification Google.
4. **Déployer** : Cliquez sur le bouton "**Déployer**". Lors du premier déploiement, vous devrez peut-être autoriser le script à accéder à vos services Google (une fenêtre d'autorisation apparaîtra).
5. **Copier l'URL** : Après un déploiement réussi, une boîte de dialogue affichera l'URL de votre application web. Copiez cette URL. C'est le lien que vous partagerez pour que les gens puissent accéder à votre formulaire.

Astuces pour personnaliser et étendre votre formulaire

Ce formulaire de base est un excellent point de départ, mais Google Apps Script offre une flexibilité incroyable pour le personnaliser et ajouter des fonctionnalités avancées. Voici quelques astuces pour aller plus loin :



Modifier le design du formulaire

Personnalisez l'apparence de votre formulaire en modifiant le code HTML dans `index.html`. Vous pouvez ajouter du CSS pour styliser les éléments, utiliser des frameworks CSS (comme Bootstrap via un CDN) ou même insérer du JavaScript plus complexe pour des validations côté client ou des interactions dynamiques.

Exemple : Ajoutez une balise `<style>` dans la section `<head>` de votre `index.html` pour modifier les couleurs, les polices, et la mise en page.



Ajouter de nouveaux champs

Si vous avez besoin de collecter plus d'informations, ajoutez simplement de nouvelles colonnes dans votre feuille Google Sheets (par exemple, "Âge", "Ville"). Ensuite, vous devrez :

- Ajouter les balises `<label>` et `<input>` correspondantes dans votre fichier `index.html`, en vous assurant d'attribuer un attribut `name` unique à chaque nouveau champ.
- Modifier la fonction `processForm(data)` dans `Code.gs` pour inclure ces nouvelles données dans l'appel à `sheet.appendRow()`. Par exemple :

```
sheet.appendRow([  
  ., data.age,  
  data.ville]);
```



Envoyer un e-mail de confirmation

Pour offrir une meilleure expérience utilisateur, vous pouvez envoyer un e-mail de confirmation à l'utilisateur après la soumission du formulaire, ou une notification à vous-même. Utilisez le service `MailApp` dans la fonction `processForm()` :

```
MailApp.sendEmail(  
{  
  to: data.email,  
  subject:  
  "Confirmation de  
  votre soumission",  
  body: "Merci pour  
  votre message, " +  
  data.nom + "!"  
});
```

N'oubliez pas que l'envoi d'e-mails nécessite une autorisation supplémentaire lors du déploiement.

Conclusion et Perspectives

Vous avez maintenant toutes les clés pour créer un formulaire web fonctionnel et autonome, capable de collecter des données directement dans Google Sheets. Cette solution, basée sur Google Apps Script, est non seulement économique mais aussi incroyablement flexible et puissante pour une multitude d'applications, de la collecte de leads à l'inscription à des événements.

L'intégration fluide entre un simple fichier HTML et la robustesse de Google Sheets, orchestrée par Apps Script, ouvre la porte à des possibilités infinies sans nécessiter de compétences avancées en développement web ou en gestion de bases de données traditionnelles. C'est une porte d'entrée idéale pour les développeurs débutants et les créateurs de contenu technique qui souhaitent automatiser et simplifier leurs flux de travail.

Accessibilité

Solution gratuite et facile à mettre en place avec un compte Google existant.

Flexibilité

Entièrement personnalisable en termes de design et de fonctionnalités.

Efficacité

Collecte de données automatisée et centralisée pour une analyse simplifiée.

N'hésitez pas à explorer la documentation officielle de Google Apps Script pour découvrir d'autres services et étendre encore les capacités de vos projets. Les seules limites sont votre imagination et vos besoins !