GFM Designer

Nov 4, 2026 · @Patrick

Introduction

GFM Designer permet à des utilisateurs non développeurs de concevoir des formulaires de saisie par glisser-déposer, puis de les publier en un clic sous forme de pages Oracle APEX. Il succède à GFM Builder, un générateur que j'avais écrit il y a plusieurs années, entièrement dans APEX.

Le principe n'a pas changé : on décrit le formulaire dans des tables de métadonnées, un package PL/SQL fabrique la page APEX correspondante. Ce qui a changé, c'est la manière de concevoir les formulaires et l'architecture qui la porte. Cet article raconte cette évolution et se termine par un petit guide utilisateur.

La version précédente

GFM Builder était une application APEX classique. On y saisissait les formulaires, leurs sections et leurs champs dans des écrans de type rapport et formulaire. Un bouton lançait ensuite le package `gfm_builder`, qui créait la page dans l'application hôte en appelant les API internes d'APEX, exactement comme le fait un import.

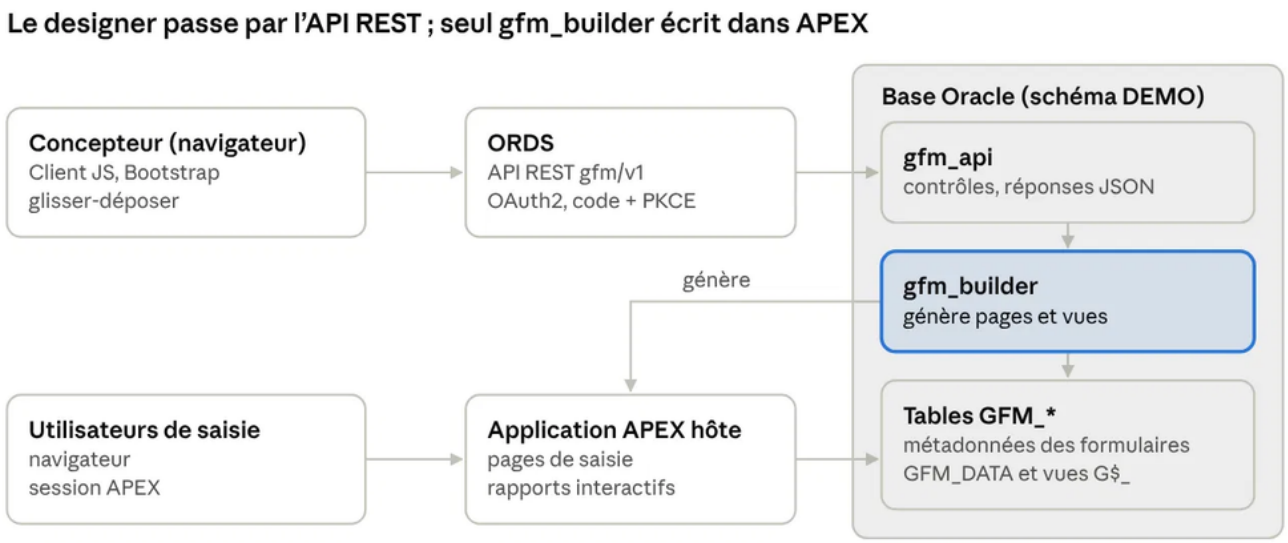
Cela fonctionnait, mais trois limites sont apparues avec le temps :

- La conception était laborieuse. Placer un champ, changer l'ordre des

sections ou la largeur d'une zone passait par des écrans de saisie et des numéros de séquence. Rien de visuel pour des utilisateurs qui ne sont pas développeurs.
- Les montées de version d'APEX cassaient la génération. Les API utilisées (`wwv_flow_api`) ne sont pas documentées.
- Le code avait vieilli. Un package d'un seul tenant, peu documenté, avec des dates stockées en texte et des choix de colonnes figés.

La nouvelle architecture

La conception quitte APEX : elle se fait dans un client web indépendant, qui ne parle à la base qu'au travers d'une API REST. La saisie, elle, reste dans les pages APEX générées.



Le concepteur et l'utilisateur de saisie ne passent jamais par le même chemin : l'un modifie des métadonnées via l'API, l'autre saisit des données dans APEX.

- Le client est écrit en JavaScript et Bootstrap 5, sans étape de compilation. SortableJS assure le glisser-déposer. Il peut être servi depuis n'importe où, y compris GitHub Pages.
- ORDS expose le module REST `gfm.designer`. L'accès est protégé par OAuth2 : flux « code d'autorisation » avec PKCE, et un rôle dédié aux concepteurs.
- Le package `gfm_api` reçoit les appels, contrôle les règles métier et renvoie le formulaire complet en JSON après chaque modification. Le client n'a plus qu'à le redessiner.
- Le package `gfm_builder`, refondu et documenté, garde ses signatures publiques. Il crée les vues et les pages dans l'application hôte.

Toutes les données et toute la logique restent dans Oracle. Le client ne stocke rien, à part le jeton de session.

Stockage générique et pages générées

Tous les formulaires enregistrent leurs données dans une seule table, `GFM_DATA`, sans aucune création de table par formulaire. Elle offre des colonnes « emplacements » de trois types :

Type	Colonnes	Usage
Texte	C001 à C050	champs texte, listes, zones de texte
Nombre	N001 à N020	champs numériques ; N001 sert de clé vers la ligne parente d'une grille
Date	D001 à D020	dates, stockées enfin en vrai type DATE

À la génération, chaque champ reçoit un emplacement libre du bon type, et deux vues donnent un visage lisible aux données :

- `G$_<CODE>` porte le formulaire de saisie, avec un déclencheur `INSTEAD OF` qui écrit dans `GFM_DATA` ;
- `G$_<CODE>_R` alimente le rapport, avec les libellés des listes et les colonnes d'audit (créé le, par, modifié le, par).

Pour un formulaire, le package produit deux pages dans l'application hôte :

1. Une page de saisie : sections en onglets, en régions ou avec un sélecteur, champs typés, champs et sections conditionnels, grilles éditables pour les sous-formulaires.
2. Un rapport interactif sur la vue `_R`, avec un bouton « Nouveau » et un lien vers chaque fiche.

Après « Submit », l'utilisateur est redirigé vers une page `END` de l'application hôte. Le package appelle désormais directement `wwv_flow_imp_page`, l'API qu'utilisent les exports d'APEX 26.1, ce qui a fait disparaître les erreurs de contraintes. Par prudence, une page existante n'est remplacée que si elle a bien été générée pour ce formulaire.

Guide utilisateur

GFM Designer s'ouvre dans un navigateur. Le menu du haut donne accès à trois écrans : Formulaires, Composants et Paramètres.

1. Se connecter

Cliquez sur « Se connecter ». Vous êtes redirigé vers la page de connexion de la base, puis ramené dans le designer. Seuls les comptes qui ont le rôle de concepteur y ont accès.

2. Choisir l’application hôte (une fois)

Dans Paramètres, choisissez l’application APEX qui recevra les pages générées. Seules les applications du groupe APEX « GFM Apps » sont proposées. Si vous changez d’application, les formulaires devront être générés à nouveau.

3. Créer un formulaire

Dans Formulaires, cliquez sur « Nouveau formulaire », donnez-lui un nom et choisissez son type :

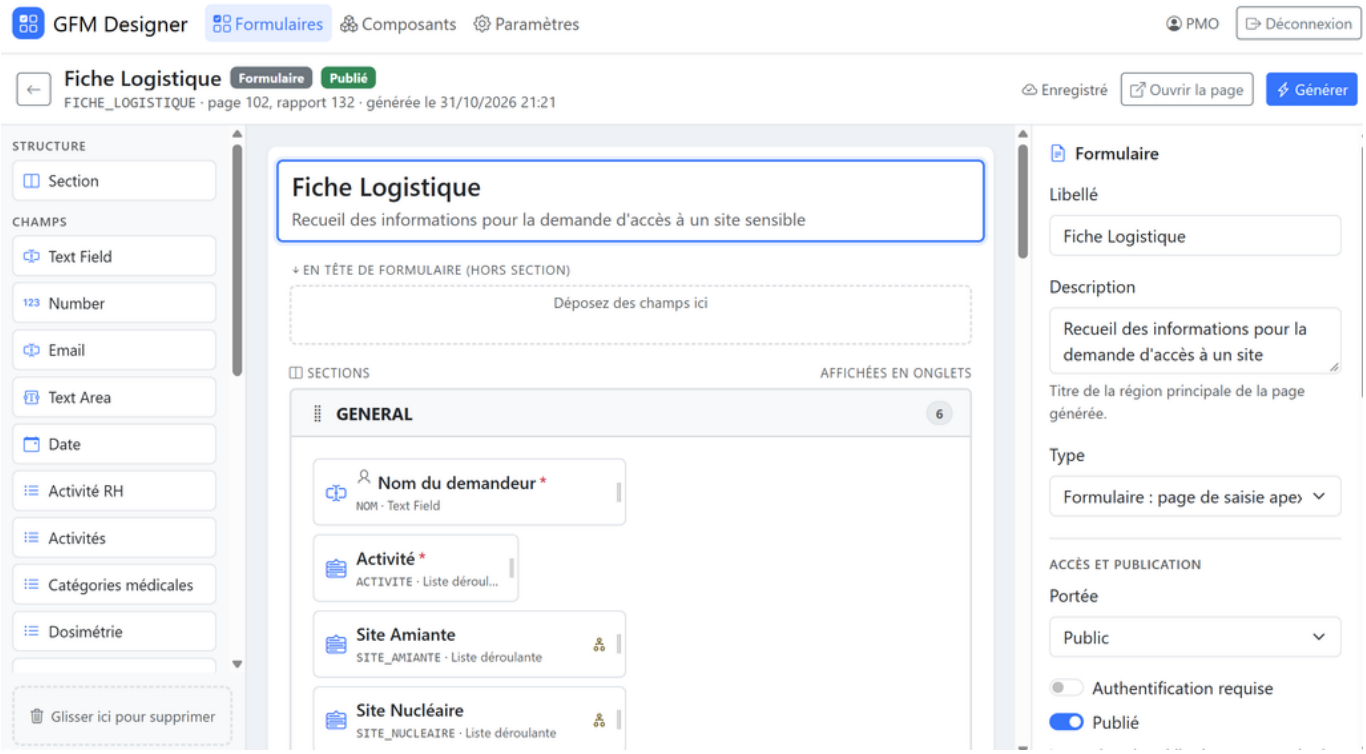
- Formulaire : une page de saisie APEX avec son rapport ;
- Grille : un tableau éditable destiné à être inclus dans un formulaire, par exemple un historique ;
- Bloc : un groupe de champs réutilisable, comme une adresse, qui n’est jamais généré seul.

4. Construire le formulaire par glisser-déposer

L’éditeur affiche la palette à gauche, le formulaire au centre et les propriétés à droite.

- Ajouter une section : glissez « Section » dans la zone des sections. Les sections deviennent des onglets dans la page générée.
- Ajouter un champ : glissez un composant (Text Field, Number, Date, liste…) dans une section, à l’endroit voulu.
- Insérer un bloc ou une grille : glissez-le depuis les groupes « Blocs » ou « Sous-formulaires ». Les champs d’un bloc sont copiés dans la section.

- Réorganiser : faites glisser les champs et les sections. Tout est enregistré immédiatement.
- Changer la largeur : tirez le bord droit d'un champ. Une ligne compte au plus 12 colonnes, soit deux demi-lignes ou trois tiers.
- Supprimer : glissez l'élément sur la corbeille en bas de la palette.



5. Régler les propriétés

Cliquez sur un champ pour modifier son libellé, son aide, son caractère obligatoire, son icône ou les valeurs d'une liste. La rubrique Affichage conditionnel permet de n'afficher un champ, ou une section entière, que si un autre champ est renseigné ou prend une valeur précise.

6. Gérer les listes de valeurs

L’écran Composants regroupe les listes partagées. On y ajoute, renomme et réordonne les valeurs par glisser-déposer, ou on les colle depuis un tableur. Une liste modifiée ici profite à tous les formulaires qui l’utilisent.

7. Générer et tester

Cliquez sur Générer. Le designer vérifie d’abord le formulaire : une liste sans valeur ou une ligne trop large est signalée par un triangle et bloque la génération. Une fois la page créée, le bouton Ouvrir la page l’affiche dans APEX, prête pour une nouvelle saisie.

Toute modification ultérieure demande une nouvelle génération. Les données déjà saisies sont conservées.

Ce que j’en retiens, et la suite

Séparer la conception de la génération a tout simplifié : l’interface peut évoluer librement, et le PL/SQL reste concentré sur ce qu’il fait bien, dans la base. Quelques leçons au passage :

- Les API internes d’APEX restent le point fragile. Se caler sur ce que produisent les exports de la version courante est la meilleure garantie.
- Une application dans le navigateur ne garde aucun secret. La vraie protection, c’est l’adresse de retour enregistrée, PKCE et le rôle de l’utilisateur.
- Un serveur de démo sans dépendance, qui imite l’API, permet de tester l’interface sans base.

Plusieurs pistes sont ouvertes : la portée et la publication des formulaires (public, groupe, privé), déjà présentes dans l'interface, et, plus loin, un moteur qui afficherait les formulaires à la volée sans générer de pages APEX.

Author

- 

  Patrick

  GPM Factory