

---

# WordPress Elementor Pro depuis Figma : token, File ID et prompt universel

Mode d'emploi simple pour créer un token Figma, récupérer le File ID, préparer le dossier d'images exportées, puis utiliser le prompt universel exactement tel quel.

**Développement Web** **Productivité IA** **10 min de lecture** **Niveau**

---

Document généré le 11/07/2026 à 20h44 · [nouv.fr/wiki/wordpress-elementor-pro-figma-token-file-id-prompt](https://nouv.fr/wiki/wordpress-elementor-pro-figma-token-file-id-prompt)

# Sommaire

9 section(s) · 10 min de lecture

## 1) Créer le token API Figma (figd\_)

↳ Étapes

↳ Important

## 2) Récupérer le File ID Figma

## 3) Préparer le dossier de travail (images)

## 4) Prompt universel à utiliser (copier-coller tel quel)

### Pré-requis à fournir

### Le prompt

### Exemple d'utilisation

Ce wiki explique **exactement** comment :

1. créer ton token API Figma,
  2. récupérer le File ID depuis l'URL Figma,
  3. préparer le dossier avec les images exportées,
  4. copier-coller le prompt universel **tel quel** pour l'envoyer ensuite dans Cursor.
- 

## 1) Créer le token API Figma (figd\_)

---

### Étapes

1. Connecte-toi à Figma.
2. Ouvre les paramètres du compte (Settings).
3. Va dans la section **Personal access tokens**.
4. Clique sur **Generate new token**.
5. Donne un nom clair au token (ex : cursor-elementor-import).
6. Coche ces scopes : `file_content:read`, `library_assets:read`, `library_content:read`, `file_metadata:read`.
7. Copie immédiatement le token généré (il commence par figd\_).

### Important

- Garde ce token privé (comme un mot de passe).
  - Ne le commit jamais dans Git.
  - Si fuite : révoque-le et recrée-en un nouveau.
- 

## 2) Récupérer le File ID Figma

---

Dans une URL Figma de ce type :

`https://www.figma.com/design/JwlfkUCwcky5aAC0U214j/Mon-Projet?...`

Le **File ID** est :

`JwlfkUCwcky5aAC0U214j`

Règle simple : c'est la partie entre `/design/` et le `/` suivant.

---

## 3) Préparer le dossier de travail (images)

---

Avant d'envoyer le prompt dans Cursor :

- exporte les images de ta maquette Figma (PNG/JPG/WebP),
- place ces fichiers **dans le dossier courant du projet**,

- le prompt demandera ensuite de les lister et mapper automatiquement aux layers.

## 4) Prompt universel à utiliser (copier-coller tel quel)

*Copie-colle ce prompt tel quel dans une nouvelle conversation Claude. Remplace uniquement les valeurs entre `{{...}}`.*

### Pré-requis à fournir

Élément	Quoi donner
Token API Figma	Ton token personnel (commence par figd_)
File ID Figma	L'ID dans l'URL : <code>figma.com/design/{FILE_ID}/...</code>
Images	Exportées depuis Figma dans le dossier de travail courant

### Le prompt

```
Crée un plugin WordPress qui importe en UN CLIC un site Elementor Pro complet :  
- Un HEADER réutilisable (Elementor Theme Builder → Header)  
- Une PAGE D'ACCUEIL avec toutes les sections  
- Un FOOTER réutilisable (Elementor Theme Builder → Footer)  
Tout est éditable dans Elementor, responsive tablette + mobile, pixel-perfect Figma.
```

```
=== ÉTAPE 1 : EXTRAIRE LE DESIGN DEPUIS FIGMA ===
```

```
Token API Figma : {{TON_TOKEN_FIGMA}}  
File ID : {{TON_FILE_ID}}
```

```
Utilise l'API Figma avec curl pour extraire TOUTES les données du design :
```

```
curl -s -H "X-Figma-Token: {{TON_TOKEN_FIGMA}}"  
"https://api.figma.com/v1/files/{{TON_FILE_ID}}"
```

```
À partir de la réponse JSON, extrais pour CHAQUE élément :
```

- Type (TEXT, RECTANGLE, INSTANCE, FRAME, LINE...)
- Position (x, y) et dimensions (width, height)
- Fills : couleurs hex exactes (convertir r/g/b float 0-1 → hex #RRGGBB)
- Strokes : couleur, épaisseur
- Corner radius
- Effects (drop shadow, blur)
- Pour les TEXT : fontFamily, fontSize, fontWeight, lineHeight, letterSpacing, textAlignHorizontal, characters (contenu texte), couleur du fill
- Pour les IMAGE fills : noter le nom du layer (correspondance avec les fichiers)

```
Identifie le frame DESKTOP principal (le plus grand en largeur).
```

```
S'il existe un frame MOBILE (iPhone/petit), analyse-le aussi pour les breakpoints.
```

```
Note toutes les couleurs uniques, toutes les fonts utilisées, tous les espacements.
```

## === ÉTAPE 2 : MAPPER LES IMAGES ===

Les images exportées depuis Figma sont dans le dossier de travail courant.  
Fais un `find` ou `ls` pour lister tous les .png/.jpg/.webp disponibles.  
Mappe chaque image au layer Figma correspondant par son nom.  
Copie-les dans le plugin sous assets/images/ avec des noms propres.

## === ÉTAPE 3 : CONVERTIR FIGMA → WEB ===

Le design Figma est en pixels haute résolution. Convertis :

- Largeur du frame Figma → max-width ~1100px web
- Tailles de police : diviser par le ratio (largeur Figma / 1100), puis ajuster pour le web (titres 36-56px, body 15-18px, labels 14-16px)
- Espacements/paddings : même ratio, arrondir aux multiples de 4 ou 8
- Corner radius : garder tel quel si < 30px, sinon proportionner
- Les fonts système (Times New Roman, Arial...) restent telles quelles
- Les Google Fonts doivent être enqueued via wp\_enqueue\_style

## === ÉTAPE 4 : CONSTRUIRE LE PLUGIN ===

Structure :

```
{{nom-du-plugin}}/  
├── {{nom-du-plugin}}.php  
└── assets/  
    └── images/
```

Le fichier PHP principal doit :

### 1. ADMIN PAGE

- Menu admin avec bouton "Importer le site"
- Affichage erreurs + liens "Voir" / "Éditer dans Elementor" après import
- Vérifier qu'Elementor Pro est actif

### 2. IMPORT (sur clic du bouton) – dans cet ordre :

- Importer toutes les images dans la médiathèque (wp\_insert\_attachment)
- Créer le HEADER (Elementor Theme Builder template)
- Créer le FOOTER (Elementor Theme Builder template)
- Créer la PAGE D'ACCUEIL
- Définir comme page d'accueil statique
- wp\_safe\_redirect + exit

### 3. HEADER – Template Elementor Theme Builder

Créer un post de type 'elementor\_library' avec :

- \_elementor\_template\_type = "header"
- \_elementor\_data = JSON du header
- Condition d'affichage : include/general (s'affiche partout)
- Contenu : navigation avec liens, style pill actif, fond et border fidèles au Figma
- RESPONSIVE : sur mobile, le header doit s'adapter (liens plus petits ou masqués si nécessaire – Elementor Pro gère le responsive natif des sections via les settings tablette/mobile de chaque widget)

Pour appliquer la condition "Afficher partout" :

```
update_post_meta( $id, '_elementor_conditions', ['include/general'] );
```

### 4. FOOTER – Template Elementor Theme Builder

Même principe que le header :

- \_elementor\_template\_type = "footer"
- \_elementor\_conditions = ['include/general']
- Contenu : colonnes de liens, mentions légales, réseaux sociaux
- RESPONSIVE : colonnes empilées sur mobile, icônes centrées

### 5. PAGE D'ACCUEIL – Elementor Page

- \_wp\_page\_template = "elementor\_header\_footer" (utilise le header/footer créés au-dessus, PAS "elementor\_canvas")

- `_elementor_data` = JSON de toutes les sections du body

Widgets Elementor Pro disponibles :

- heading, text-editor, image, button, divider, spacer
- google\_maps, social-icons, icon-list
- form (widget natif Elementor Pro – PAS de HTML custom)
- nav-menu (widget natif pour la navigation dans le header)
- image-box, call-to-action
- slides, image-carousel
- posts, portfolio

Pour le FORMULAIRE : utiliser le widget Elementor Pro "form" :

```
widgetType = 'form'
settings = [
  'form_name' => '...',
  'form_fields' => [
    ['_id'=>'name', 'field_type'=>'text', 'field_label'=>'Nom', 'placeholder'=>'...', 'required'=>'true', 'width'=>'50'],
    ['_id'=>'tel', 'field_type'=>'tel', 'field_label'=>'Tél', 'placeholder'=>'...', 'required'=>'true', 'width'=>'50'],
    ['_id'=>'msg', 'field_type'=>'textarea', 'field_label'=>'Message', 'placeholder'=>'...', 'required'=>'true', 'width'=>'100'],
  ],
  'button_text' => '...',
  'button_size' => 'md',
  'button_align' => 'center',
  'email_to' => get_option('admin_email'),
  'email_subject' => '...',
  // + tous les styles : couleurs, border-radius, typography...
]
```

6. CSS CUSTOM (via `wp_head`, uniquement si nécessaire)

- Hover effects sur les cartes produits
- Animations au scroll si présentes dans la maquette
- Charger les Google Fonts

=== ÉTAPE 5 : RESPONSIVE – CRITIQUE ===

Chaque widget Elementor doit avoir ses 3 breakpoints définis :

- DESKTOP : valeurs du Figma converties
- TABLETTE (< 1024px) : tailles réduites ~80%, colonnes adaptées
- MOBILE (< 767px) : tout en 1 colonne, tailles réduites ~60%

Syntaxe Elementor pour le responsive :

```
'typography_font_size' => [
  'unit' => 'px',
  'size' => 48, // desktop
  'sizes' => [
    'tablet' => 36, // tablette
    'mobile' => 28 // mobile
  ]
]
```

Appliquer le responsive sur :

- TOUTES les `typography_font_size`
- TOUS les padding de sections
- Les `min_height` de sections (hero 100vh desktop → 70vh mobile)
- Les gap entre colonnes
- Les `width` d'images
- Les margins et paddings des widgets

Pour le HEADER responsive :

- Desktop : liens en ligne horizontale
- Tablette : liens plus compacts ou sur 2 lignes
- Mobile : utiliser le widget nav-menu d'Elementor Pro qui a un

```
hamburger menu intégré (settings: toggle = 'burger', breakpoint = 'tablet')
```

Pour le FOOTER responsive :

- Desktop : 3 colonnes
- Tablette : 2 colonnes
- Mobile : 1 colonne, tout centré

Pour les GRILLES produits/images :

- Desktop : 2 colonnes (50/50)
- Mobile : 1 colonne (100%)
  - Utiliser les settings Elementor : '\_column\_size' => 50 desktop, puis sur mobile Elementor empile automatiquement, mais ajouter 'padding' responsive pour les marges

=== CONTRAINTES ===

- Compatible WordPress 6.0+ et Elementor PRO
- Tout dans UN SEUL fichier PHP + le dossier images
- Les IDs Elementor doivent être uniques (préfixe court + numéro incrémental)
- HEADER et FOOTER = templates Elementor Theme Builder séparés (pas dans la page)
- La page utilise "elementor\_header\_footer" (PAS "elementor\_canvas")
- Le formulaire utilise le widget "form" natif Elementor Pro (PAS du HTML custom)
- La navigation dans le header utilise le widget "nav-menu" d'Elementor Pro avec hamburger automatique sur mobile
- Images dans la médiathèque (pas en URL directe du plugin)
- wp\_safe\_redirect + exit après l'import
- isset() avant wp\_verify\_nonce()
- CHAQUE widget doit avoir les valeurs responsive tablet + mobile définies
- Les \_elementor\_conditions sur header/footer = ['include/general']

=== NOM DU PROJET ===

Nom du plugin : {{nom-du-plugin}}

📄 Copier

---

## Exemple d'utilisation

```
Token API Figma : figd_V0vspUQb9W3fu18H-A49hzzf2jc6vhSBEmezLSld
```

```
File ID : JwLwfkUCwcky5aAC0U214j
```

```
Nom du plugin : boulangerie-starter
```

```
Nom du site : La Boulangerie du Coin
```

📄 Copier

Les images doivent être dans le dossier courant (même endroit que le projet). Claude va les trouver automatiquement, les mapper aux layers Figma, et tout intégrer.

À la fin, zippe manuellement le dossier du plugin créé, puis importe ce fichier `.zip` dans WordPress via **Extensions > Ajouter > Téléverser une extension**.