**Guide de Démarrage : Application Mobile Flutter de A à Z pour Gnut 06**

Ce guide complet vous explique comment installer Flutter, créer votre projet, comprendre son architecture et lancer l'application mobile Gnut 06.

**Étape 1 : Prérequis - Installation de Flutter**

Avant de commencer, vous devez installer l'environnement de développement Flutter.

1. **Installer le Flutter SDK** : Suivez les instructions officielles sur le site de [Flutter](https://flutter.dev/docs/get-started/install) pour votre système d'exploitation (Windows, macOS ou Linux). C'est l'étape la plus importante.
2. **Installer un Éditeur de Code** :
   * **Visual Studio Code (Recommandé)** : Installez l'extension Flutter pour une expérience de développement optimale.
   * **Android Studio** : Un environnement complet qui inclut les outils pour le développement Android. Installez le plugin Flutter.
3. **Vérifier l'Installation** : Ouvrez un nouveau terminal et exécutez la commande flutter doctor. Cet outil va analyser votre installation et vous indiquer s'il manque des composants (comme le SDK Android ou Xcode pour iOS). Suivez ses recommandations jusqu'à ce que tout soit coché en vert.

**Étape 2 : Création du Projet Flutter**

1. Ouvrez votre terminal et naviguez jusqu'au dossier où vous souhaitez créer votre projet.
2. Exécutez la commande suivante pour créer un nouveau projet Flutter :
3. flutter create gnut06\_mobile\_app
4. Une fois le processus terminé, déplacez-vous dans le dossier du projet :
5. cd gnut06\_mobile\_app

**Étape 3 : Analyse de la Structure du Projet**

Voici une analyse des dossiers et fichiers les plus importants que vous venez de créer :

* lib/ : C'est le cœur de votre projet. 99% de votre code sera écrit dans ce dossier. Le fichier principal est lib/main.dart, qui est le point d'entrée de votre application.
* pubspec.yaml : L'équivalent du package.json en JavaScript. Ce fichier gère les métadonnées de votre projet et, surtout, ses **dépendances** (les "packages" ou bibliothèques externes). C'est ici que vous ajouterez des fonctionnalités comme les icônes, les requêtes HTTP, etc.
* assets/ (à créer) : Un dossier que vous créerez vous-même pour stocker des ressources statiques comme les images, les polices de caractères, etc.
* android/ et ios/ : Ces dossiers contiennent les projets natifs Android et iOS. Vous n'aurez que très rarement besoin de les modifier directement.

**Étape 4 : Lancer l'Application**

1. Assurez-vous qu'un appareil est connecté ou qu'un émulateur/simulateur est lancé.
   * Vous pouvez voir la liste des appareils disponibles avec la commande flutter devices.
   * Pour lancer un émulateur, vous pouvez utiliser Android Studio ou la ligne de commande.
2. Dans votre terminal, à la racine du projet, exécutez :
3. flutter run

La première compilation peut prendre quelques minutes. Les suivantes seront beaucoup plus rapides grâce au "Hot Reload" de Flutter.

**Étape 5 : Architecture du Code dans** lib/

Pour un projet propre et évolutif, il est recommandé d'organiser le dossier lib comme suit :

* lib/main.dart : Point d'entrée, configuration principale.
* lib/screens/ : Contient les différents écrans de votre application (ex: home\_screen.dart, events\_screen.dart).
* lib/widgets/ : Contient des composants réutilisables (widgets) que vous utiliserez sur plusieurs écrans (ex: custom\_app\_bar.dart, event\_card.dart).
* lib/models/ : Contient les classes qui modélisent vos données (ex: event\_model.dart).
* lib/services/ : Pour la logique métier, comme les appels à une API backend.

Vous disposez maintenant d'un projet Flutter fonctionnel, avec une structure claire et prête pour le développement de l'application Gnut 06.