

Synthèse eMaya

La ferme **Maya** souhaite avoir une application mobile cross-platform pour permettre aux clients de consulter les produits de la ferme et de les commander.

Pour le développement de l'application, Logma (ESN fournisseur de la ferme Maya) a choisi d'utiliser le SDK **Flutter**.

Mise en place de l'environnement

Création du projet Flutter dans VSCode. Architecture Clean Flutter (Extension).

Générer le dossier core avec la commande : Onyxio : Generate core folder.

Générer le dossier feature avec la commande : Onyxio : Generate feature folder.

Paramétrage des couleurs dans colors.dart.

Paramétrage du thème dans theme.dart.

Création de l'appBar dans le fichier home_page, les redirections vers les autres pages se font dans ce fichier.

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    bottomNavigationBar: NavigationBar(
      onDestinationSelected: (int index) {
        setState(() {
          currentPageIndex = index;
        });
      },
      indicatorColor: Colors.amber,
      selectedIndex: currentPageIndex,
      destinations: const <Widget>[

        NavigationDestination(
          selectedIcon: Icon(Icons.home),
          icon: Icon(Icons.home_outlined),
          label: 'Accueil',
        ),

        NavigationDestination(
          icon: Icon(Icons.category_sharp),
          label: 'Catégories',
        ),
```

```
body: <Widget>[
  /// page produit
  const ProduitPage(),
```

Création d'une page

Générer le dossier core avec la commande : Onyxio : Generate core folder.

Donner au dossier le nom de la page.

Renommer le fichier pages.dart qui se trouve dans presentation/pages en nomDeLaPage_page.dart.

Ce fichier permettra d'afficher ce que l'on souhaite. Pour ajouter des widgets il faudra les créer dans le dossier widget.

Utilisation d'une API

Permettre de récupérer des données de la base de données, de les consulter, modifier, supprimer.

Création d'un projet symfony qui servira pour le projet initial afin d'utiliser l'API.

Commande création : symfony new maya-api

Installer Api Platform avec composer : composer require api (dans le dossier du projet)

Mettre sa base de données dans le fichier .env

Dans des dossiers Entity et Repository, placer les fichiers correspondants qui proviennent du projet Maya Web.

Faire ces imports dans les fichiers Entity :

- use ApiPlatform\Metadata\ApiResource;
- use ApiPlatform\Metadata\ApiProperty;
- use ApiPlatform\Metadata\GetCollection;
- use ApiPlatform\Metadata\Get;
- use Symfony\Component\Serializer\Annotation\Groups;

Puis rajouter cette partie de code avant votre « class » :

```
#[ApiResponse(
    normalizationContext: ['groups' => ['categorie:read']],
    operations: [
        new Get(),
        new GetCollection(),
    ],
)]
```

Enfin, il ne reste plus qu'à rajouter « #[Groups(['categorie:read'])] » au-dessus de chacun des champ privée que vous voulez mettre en lecture.

Appeler l'API dans le projet mobile eMaya (exemple Catégories)

Récupérer ce que renvoie le GET des catégories, le mettre sur QuickType (site) et générer en Dart le code. L'implanter dans un nouveau fichier categorie.dart qui se situera dans lib/data/models.

Créer un fichier categorie_api.dart qui se situera dans le répertoire lib/data/provider et écrire ce code :

```

class CategorieApi {
  Future<List<Categorie>?> getAllCategories() async {
    var client = http.Client();
    var uri = Uri.parse('http://10.0.2.2:8000/api/categories');
    var response = await client.get(uri);
    if (response.statusCode == 200) {
      return categoriesFromJson(
        const Utf8Decoder().convert(response.bodyBytes));
    }
    return null;
  }
}

```

Créer le fichier `categorie_service.dart` qui se trouvera dans le dossier `lib/data/service` et écrire ce code :

```

class CategorieService {
  final _api = CategorieApi();

  Future<List<Categorie>?> getAllCategories() async {
    return _api.getAllCategories();
  }
}

```

Dans votre affichage de la page catégorie, il faut créer une liste contenant les Catégories et créer un `loadCategories`. Vous pourrez appeler vos noms et images de catégories plus bas dans votre code.