AP4 - Documentation technique

Sommaire

1) Technologie

L'application M2L a été faite sous Flutter qui est fait en Dart.

2) Installation

1. Allez sur le site https://docs.flutter.dev/get-started/install/windows et télécharger le dossier zip dans la partie Get the Flutter SDK.

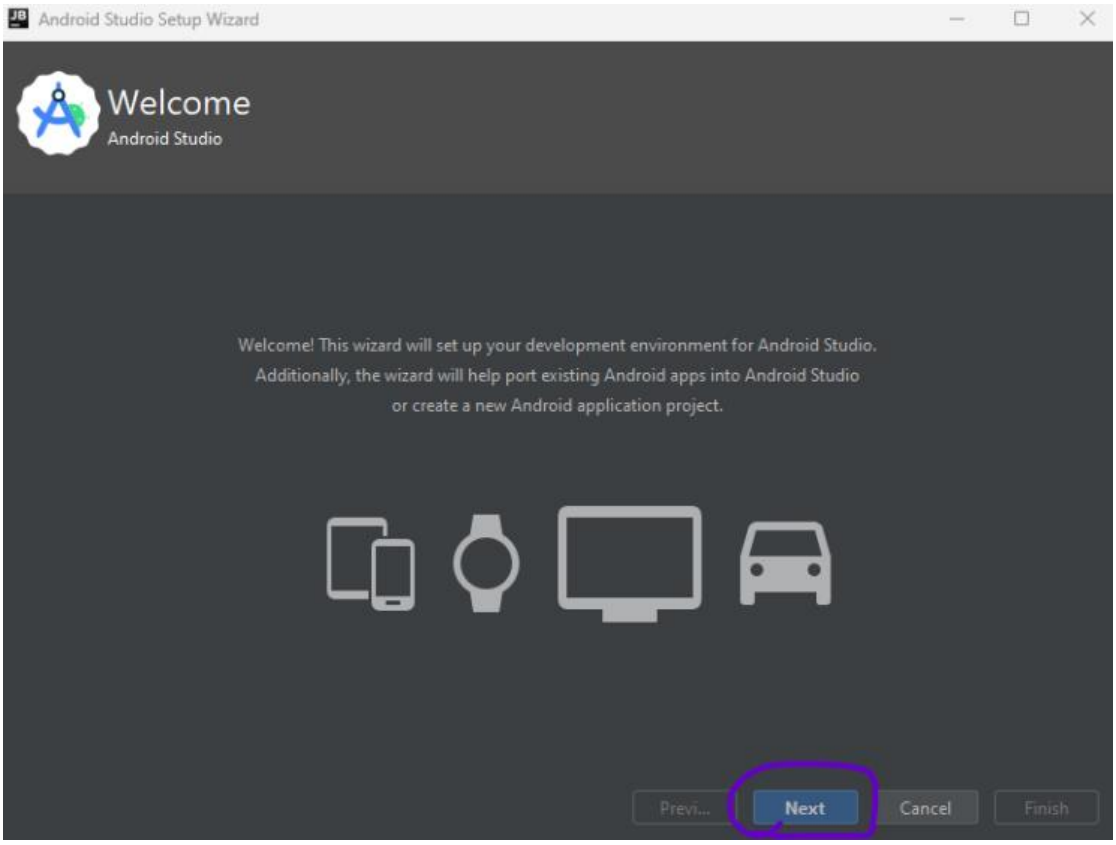
2. Extraire le fichier zip et placer le dossier flutter dans un emplacement facile d'accès.

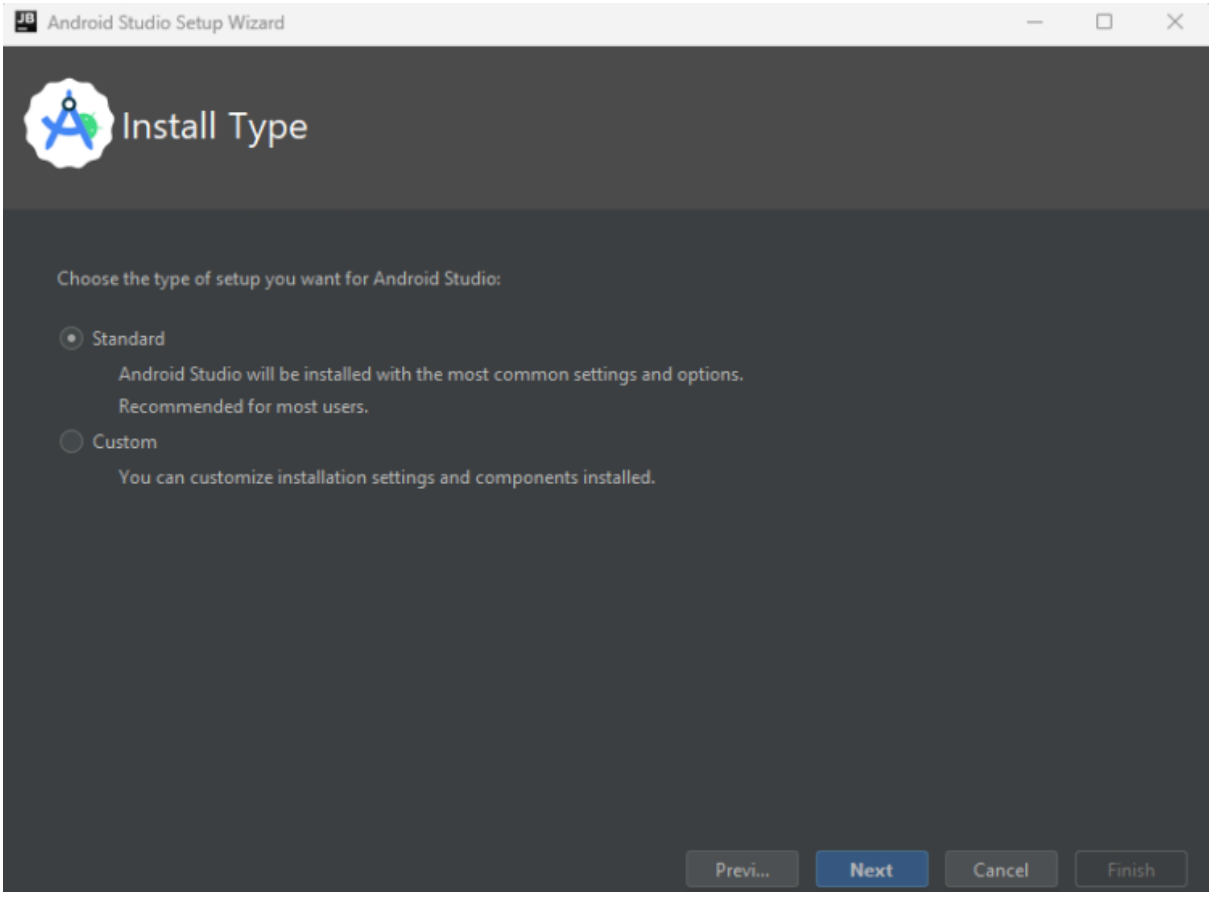
3. Il va falloir changer vos variables d'environnements. Dans la barre de recherche de Windows, tapez env puis cliquez sur modifier les variables d'environnement. Rechercher la variable Path, cliquez sur modifier et vous y insérer le chemin du SDK installer précédemment (flutter/bin).

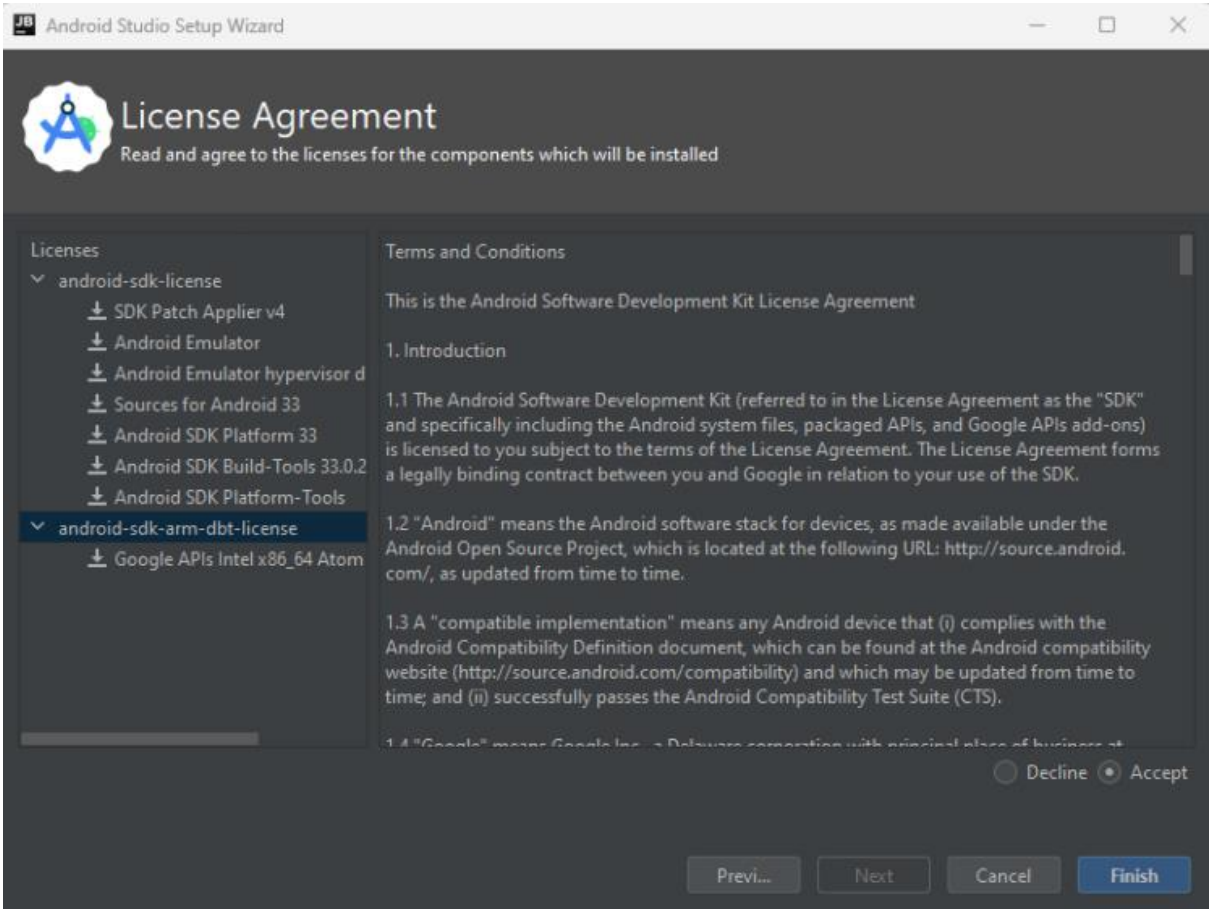
4. Pour vérifier si Flutter fonctionne en faisant la commande flutter doctor dans cmd et on peut voir qu'il y a plusieurs cases (validé ou non). Les cases qui nous intéressent sont la partie Flutter. On remarque qu'il nous manque le SDK d'Android et Android Studio.

5. Télécharger Android Studio ici:
https://developer.android.com/studioInstaller

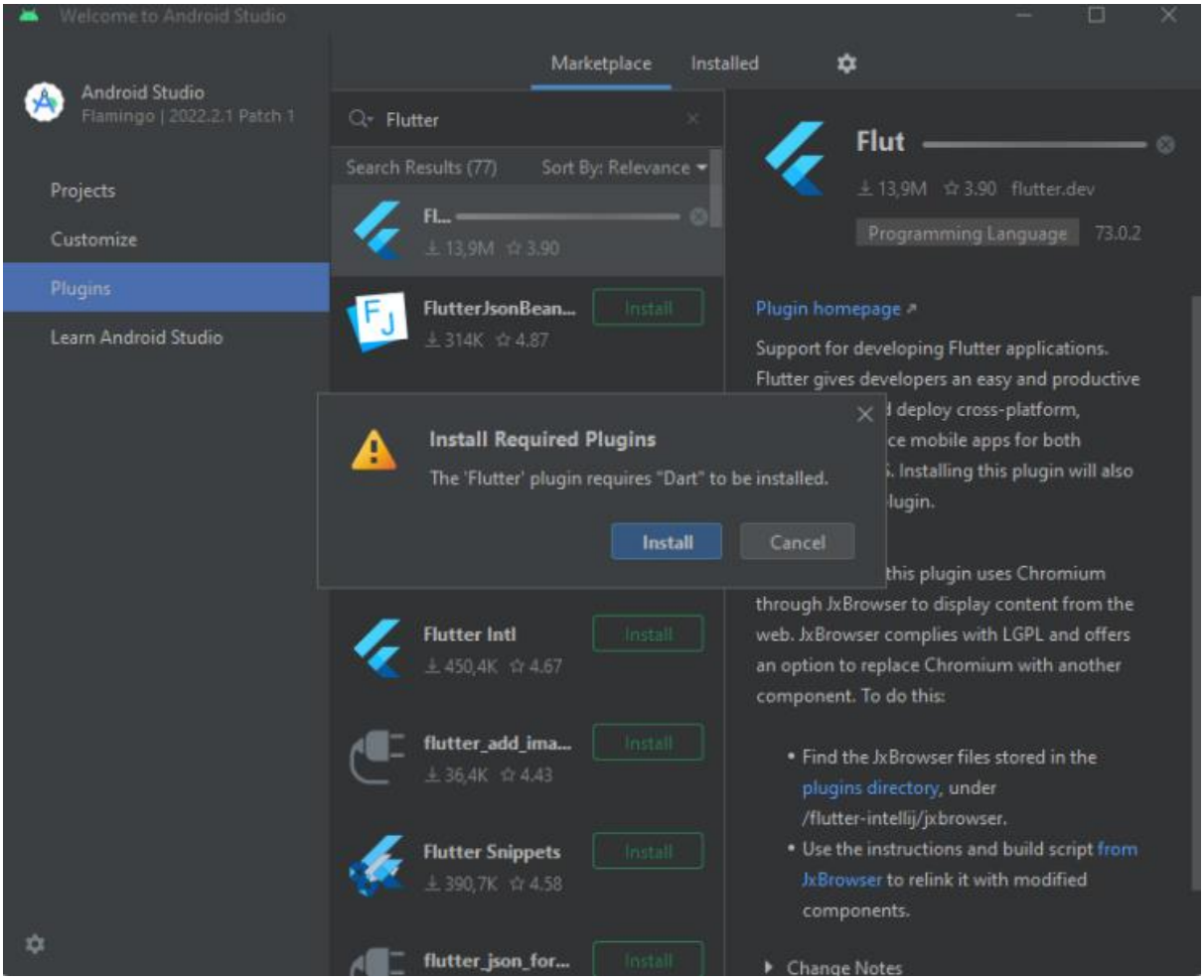
6. Ouvrir Android Studio et suivez les étapes suivantes :




Android Studio Setup Wizard
Install Type
Choose the type of setup you want for Android Studio:
Standard
Android Studio will be installed with the most common settings and options.
Recommended for most users.
Custom
You can customize installation settings and components installed.
Previ...
Next
Cancel
Finish


Android Studio Setup Wizard
License Agreement
Read and agree to the licenses for the components which will be installed
Licenses
android-sdk-license
SDK Patch Applier v4
Android Emulator
Android Emulator hypervisor d
Sources for Android 33
Android SDK Platform 33
Android SDK Build-Tools 33.0.2
Android SDK Platform-Tools
android-sdk-arm-dbt-license
Google APIs Intel x86_64 Atom
Terms and Conditions
This is the Android Software Development Kit License Agreement
1. Introduction
1.1 The Android Software Development Kit (referred to in the License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of the License Agreement. The License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.
1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: http://source.android.com/, as updated from time to time.
1.3 A "compatible implementation" means any Android device that (i) complies with the Android Compatibility Definition document, which can be found at the Android compatibility website (http://source.android.com/compatibility) and which may be updated from time to time; and (ii) successfully passes the Android Compatibility Test Suite (CTS).
Decline
Accept
Previ...
Next
Cancel
Finish

7. Une fois android installé on va installer les plugins flutter, aller dans Plugins puis tapez Flutter et installer le plugin, également il vous demandera d'installer Dart en même temps, vous acceptez bien évidemment.



8. Une fois l'installation faite il vous demande de redémarrer l'IDE pour prendre en compte les plugins installés. Refaisons un point sur l'invite de commande. Retapons 'flutter doctor'. Normalement une autre case a été validé, celle d'Android Studio, mais maintenant on nous demande les licences Android, utilisez la commande :
flutter doctor --android-licenses

9. Acceptez avec la touche Y puis relancez 'flutter doctor'

10. Maintenant il nous manque l'émulateur mobile que nous allons créer via Android Studio. Vous avez deux choix possibles :
● si vous avez un smartphone sous Android, connectez-le à votre ordinateur et il devrait être reconnu.
● Installez un émulateur avec les étapes ci-dessous. Nous allons dans Android Studio. Puis dans "More Actions" et "Virtual Device Manager", "Create virtual device". Puis nous pouvons choisir un téléphone, choisissez celui que vous voulez, choisissez également la version d'Android (une récente de préférence (mais pas trop non plus)) et téléchargez-la.
Finalement si nous relançons flutter doctor, nous avons tout de valider :).

3) Fonctionnement

1. Pour se connecter il faut un compte administrateur et une fois connecté vous pouvez tout modifier comme edit les stocks, ajouter des articles, supprimer des articles, modifier des articles.

2. Ce code définit une méthode statique `Delete` qui envoie une requête HTTP DELETE à l'URL de base (`$baseUrl/articles/$id` où `$id` est un paramètre d'entrée) avec l'ID de la question à supprimer. Si la requête renvoie un code de statut 200, l'application navigue vers la page "/liste", sinon elle navigue vers la page principale ("/").

```
static Delete(BuildContext context, int id) async {
  var res = await http.delete(Uri.parse('$baseUrl/articles/$id'),
      body: id.toString());
  if (res.statusCode == 200) {
    Navigator.pushNamed(context, '/liste');
  } else {
    Navigator.pushNamed(context, '/');
  }
}
}
```

3. Il y a la méthode "Update" qui prend en paramètres un contexte BuildContext, un id de type entier et 4 chaînes de caractères : nom, prix et stock et image.
Cette méthode envoie une requête PUT à une API avec les données mises à jour. Si la requête renvoie un code d'état 200, la méthode utilise le contexte pour naviguer vers la page '/liste'. Sinon, elle navigue vers la page '/' (la page d'accueil). Si une erreur se produit pendant l'exécution de la requête, la méthode retourne une erreur Future.

```
static ajout(BuildContext context, String name, String prix,
    String image, String quantite) async {
  try {
    var res = await http.post(
      Uri.parse("$baseUrl/articles"),
      headers: <String, String>{
        'Content-Type': 'application/json; charset=UTF-8',
      },
      body: jsonEncode(<String, String>{
        'name': name,
        'prix': prix,
        'image': image,
        'quantite': quantite
      }),
    );
    if (res.statusCode == 200) {
      Navigator.pushNamed(context, '/liste');
    } else {
      Navigator.pushNamed(context, '/');
    }
  } catch (error) {
    return Future.error(error);
  }
}
```

4. Il y a la méthode "ajout" qui envoie une requête HTTP POST à l'URL spécifiée avec les données de thème, question et réponse encodées en JSON dans le corps de la requête. Si la réponse renvoyée à un code d'état 200, cela signifie que la requête a réussi et il redirige l'utilisateur vers la page '/liste'. Sinon, il redirige l'utilisateur vers la page principale ('/') en cas d'erreur.

```
static Update(BuildContext context, int id, String name, String prix,
    String image, String quantite) async {
  try {
    var res = await http.put(
      Uri.parse("$baseUrl/articles/$id"),
      headers: <String, String>{
        'Content-Type': 'application/json; charset=UTF-8',
      },
      body: jsonEncode(<String, String>{
        'name': name,
        'prix': prix,
        'image': image,
        'quantite': quantite,
        'id': id.toString()
      }),
    );
    if (res.statusCode == 200) {
      Navigator.pushNamed(context, '/liste');
    } else {
      Navigator.pushNamed(context, '/');
    }
  } catch (error) {
    return Future.error(error);
  }
}
```