Navigation ou routage avec Getx

La navigation ou routage permet de définir la route entre les différentes pages d'un projet. Ici nous allons admettre que vous avez déjà pris en main Getx et que le souci observé est juste la notion de navigation. Pour cela nous allons admettre que nous avons deux pages. Une page d'ouverture qui possède un bouton "Commencer" qui est sensé nous rediriger sur la page d'enregistrement.

- Le fichier main.dart est le suivant

```
import 'package:flutter/material.dart';
import 'widgets/page_ouverture.dart';
import 'widgets/accueil.dart';
import 'package:get/get.dart';
import 'widgets/modules/enregistrement/views/enregistrement_view.dart';
import 'widgets/modules/enregistrement/bindings/enregistrement_binding.dart';
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      theme: ThemeData(
        primarySwatch: Colors.blueGrey,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      //on définit les routes
      getPages: [
        GetPage(name: "/page_ouverture", page:() => Page_Ouverture()),
        GetPage(name: "/accueil", page:() => Accueil() ),
        GetPage(name: "/enregistrement", page:() => Enregistrement(), binding:
RegisterBinding(), ),
      ],
      initialRoute: "/page_ouverture",
    );
  }
}
```

Dans le main nous avons définit l'attribut *getPages* de *GetMaterialApp*. Nous avons ainsi définit les pages utilisées par notre projet. Par exemple:

```
GetPage(name: "/page_ouverture", page:() => Page_Ouverture()),
```

Est une page dont le nom est */page_ouverture* et elle retourne une instance de la classe *Page_Ouverture()* qui est dans le fichier *import 'widgets/page_ouverture.dart';*.

Pour définir la page initiale on définit l'attribut *initialRoute* de *GetMaterialApp()*:

```
initialRoute: "/page_ouverture",
```

Il est à observer qu'il y'a une page qui est définit avec l'attribut *binding*. Cet attribut permet de gérer la notion de dépendance.

```
GetPage(name: "/enregistrement", page:() => Enregistrement(), binding:
RegisterBinding(),)
```

Les routes ainsi définies, il est possible d'y accéder. Lorsque l'application est lancée, la page initiale s'ouvre. Et de là pour aller à la page d'enregistrement il suffit d'écrire l'instruction suivante:

```
Get.toNamed('/enregistrement');
```

Il est utile de spécifier que pour la navigation seul les vues sont concernées. Les contrôleurs et autres ne sont pas pris en compte car il n'affichent rien.

> Si l'on voulait aller à la page accueil il suffirait de mettre le nom de la page en question définit dans *getPages()*. Les dépendances donc nous avons évoqué précédemment. Avec Getx nous avons la possibilité d'effectuer une intégration complète des routes, du gestionnaire d'état et du gestionnaire de dépendances. Lorsqu'une route est supprimée de la pile, tous les contrôleurs, variables et instances d'objets qui lui sont associés sont supprimés de la mémoire. Nous allons déclarer une classe Biding qui permettra de découpler l'injection des dépendances tout en liant les routes au gestionnaire d'état et au gestionnaire de dépendances.

```
import 'package:get/get.dart';

import '../controllers/enregistrement_controller.dart';

class RegisterBinding extends Bindings {
  @override
  void dependencies() {
    Get.lazyPut<RegisterController>(
      () => RegisterController(),
    );
  }
}
```

où *RegisterController* est une instance du contrôleur de la page d'enregistrement. Il est possible de naviguer vers la prochaine fenêtre sans donner la possibilité de revenir en arrière (très utile lorsqu'on fait les login/register). Cela se fait ainsi:

```
Get.offNamed("/Fenetre_suivante");
```

la méthode GetPage() possèdde un attribut pour effectuer des transitions entre des fenêtres. On peut présenter:

```
transition: Transition.zoom,
transition: Transition.cupertino,
```

> NOUTCHA NGAPI Jonathan