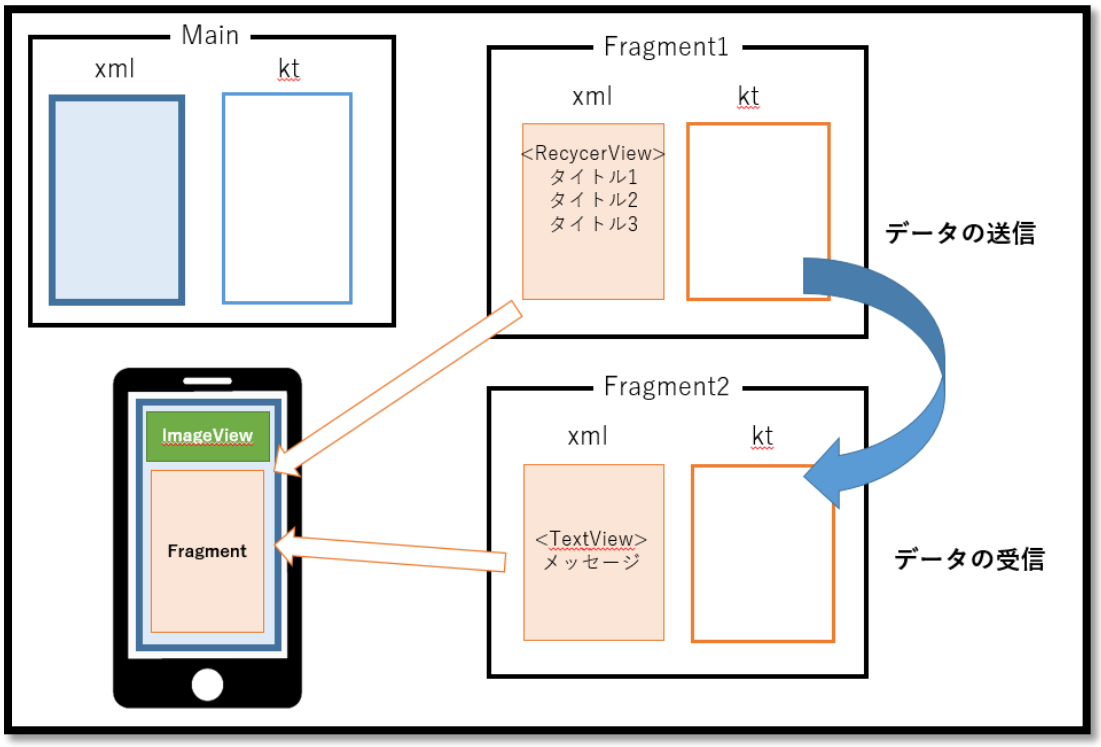
本日の内容

・Fragment の画面遷移
・setFragmentResult

今回は Fragment の画面遷移と値の受渡しにフォーカスを当てて学んでいきます。



既に Activity 経由で Fragment の遷移は行いましたが、今回は Fragment 自身での遷移も試していきましょう。
フラグメント ｜ Android デベロッパー ｜ Android Developers

フラグメント間でデータを渡す ｜ Android デベロッパー ｜ Android Developers

今回は GitHub のコードを動かしながらやっていきます。
下記のプロジェクトを[クローン]or [zip ダウンロード]し動作を確認します。

https://github.com/iyStudy/Kotlin_Fragment_Sample

実行すると右図のような画面で
クリックしても何も反応しません。

この状態から、次の SecondFragment へ遷移する方法を学びます。

■ParentFragmentManager

ParentFragmentManager は、Android 開発において、フラグメント間のナビゲーションやデータの交換、ライフサイクルの管理を助けるためのクラスです。フラグメントは Android アプリの UI 部分をモジュール化して再利用するためのコンポーネントで、ParentFragmentManager はこれらのフラグメントを管理するためのものです。

以下は、ParentFragmentManager の主な用途と特徴です。

◆主な用途

・フラグメントの追加、削除、置換

ParentFragmentManager を使用して、フラグメントを追加、削除、または置換できます。これにより、アプリのナビゲーションや UI のダイナミックな更新が可能になります。

・フラグメントのバックスタックの管理

ユーザーがナビゲーションを進めたり戻ったりする際のフラグメントのスタックを管理します。これにより、ユーザーはアプリ内でスムーズにナビゲーションできます。

・フラグメント間の通信

複数のフラグメント間でデータを交換する際にも ParentFragmentManager を利用できます。これにより、フラグメント間の連携が向上します。

◆使い方の例

```
val newFragment = ExampleFragment()  // 新しいフラグメントのインスタンスを作成
val transaction = parentFragmentManager.beginTransaction()  // トランザクションを開始

// フラグメントを置換
transaction.replace(R.id.fragment_container, newFragment)
transaction.addToBackStack(null)  // 置換前のフラグメントをバックスタックに追加

transaction.commit()  // トランザクションをコミットして変更を適用
```

以下のようにまとめて書くことも可能

```
parentFragmentManager.beginTransaction()
    .replace(R.id.fragmentContainerView,SecondFragment())
    .addToBackStack(null)
    .commit()
```

■setFragmentResult

setFragmentResult は、フラグメント間でデータをやり取りするためのメソッドです。これは、特にフラグメントから別のフラグメントにデータを送信する際に役立ちます。setFragmentResult を使用すると、FragmentResultListener を通じて結果を受け取ることができます。

◆使い方

□データを送信するフラグメント

```
val bundle = Bundle().apply {
    putInt("key", 123) //  データを Bundle に追加
}
// setFragmentResult を使用してデータを送信
parentFragmentManager.setFragmentResult("requestKey", bundle)
```

このコードは、キーと値のペアを Bundle オブジェクトに追加して、setFragmentResult メソッドを使用してそのデータを送信しています。"requestKey"は、この結果を識別するためのキーです。

以下のような使い方も出来ます

```
val adapter = ProfileAdapter(profileList) { selectedProfile ->
    parentFragmentManager.setFragmentResult(
        REQUEST_PROFILE_DETAIL,
        bundleOf("SELECTED_PROFILE" to selectedProfile)
    )
```

□データを受け取るフラグメント

```
parentFragmentManager.setFragmentResultListener("requestKey",this,
FragmentResultListener { requestKey, result ->
        //  データを取り出す
        val data = result.getInt("key")
        Log.d("TAG", "Received result: $data")
})
```

このコードは、setFragmentResultListener メソッドを使用して、特定のリクエストキーに関連付けられた結果をリッスンしています。リスナーは、リクエストキーと結果の Bundle をパラメータとして受け取る FragmentResultListener を使用します。

以下のような使い方も出来ます

```
// argumentsでFragmentに渡されたメニューデータを取得し、それを利用してUIを更新する。
parentFragmentManager.setFragmentResultListener(FirstFragment.REQUEST_PROFILE_DETAIL, this) { _, bundle ->
    val selectedProfile: Profile = bundle.getParcelable("SELECTED_PROFILE")!!
```

◆注意点

データを送受信する際には、リクエストキーとバンドルのキーを正確に指定する必要があります。

setFragmentResult は、フラグメントのライフサイクルを考慮したデータの送受信を可能にするため、Activity 間でのデータの送受信に startActivityForResult と onActivityResult を使用するのとは異なり、より簡潔で安全にデータを送受信することができます。