

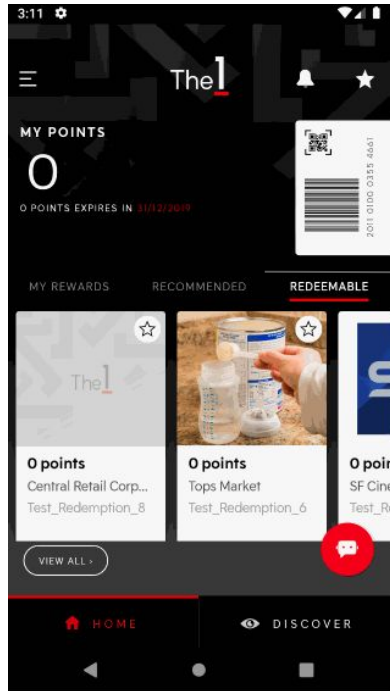


# Android Fragment Navigation

Oak

Growth Session #24 - May 16-17 2019

# Intro



Activity

Fragment

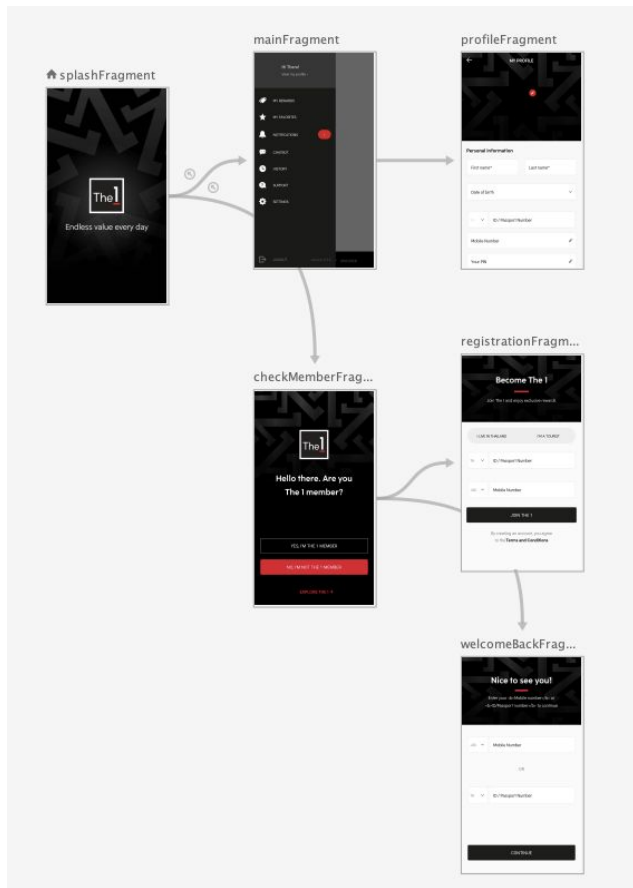
# What is Fragment Navigation?

- A tool to help implementing navigation, from simple navigation to more complex ones, such as app bars and navigation drawers.
- It ensures a consistent and predictable user experience by following Material Design guidelines.

# What is Fragment Navigation?

- Handling fragment transactions.
- Handling Up and Back actions correctly by default.
- Providing standardized resources for animations and transitions.
- Including Navigation UI patterns, such as navigation drawers and bottom navigation, with minimal additional work.
- Safe Args - Type safety when navigating and passing data between destinations with code generations.

# Intro



```


<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/nav_main"
    app:startDestination="@id/splashFragment">

    <fragment
        android:id="@+id/splashFragment"
        android:name="th.co.the1.the1app.fragment.register.SplashFragment"
        android:label="fragment_splash"
        tools:layout="@layout/fragment_splash">
        <action
            android:id="@+id/toMain"
            app:destination="@id/mainFragment"
            app:enterAnim="@anim/side_in_right"
            app:exitAnim="@anim/side_out_left"
            app:popEnterAnim="@anim/side_in_left"
            app:popExitAnim="@anim/side_out_right"
            app:popUpTo="@id/nav_main"
            app:popUpToInclusive="true" />
        <action
            android:id="@+id/toCheckMember"
            app:destination="@id/checkMemberFragment"
            app:enterAnim="@anim/nav_default_enter_anim"
            app:exitAnim="@anim/nav_default_exit_anim"
            app:popEnterAnim="@anim/nav_default_pop_enter_anim"
            app:popExitAnim="@anim/nav_default_pop_exit_anim"
            app:popUpTo="@id/nav_main"
            app:popUpToInclusive="true" />
        </fragment>

    <fragment
        android:id="@+id/mainFragment"
        android:name="th.co.the1.the1app.fragment.MainFragment"
        android:label="fragment_main"
        tools:layout="@layout/activity_main">
        <action
            android:id="@+id/toProfile"
            app:destination="@id/profileFragment"
            app:enterAnim="@anim/side_in_right"
            app:exitAnim="@anim/side_out_left"
  
```


## Passing data between fragment (manually)

### Sender



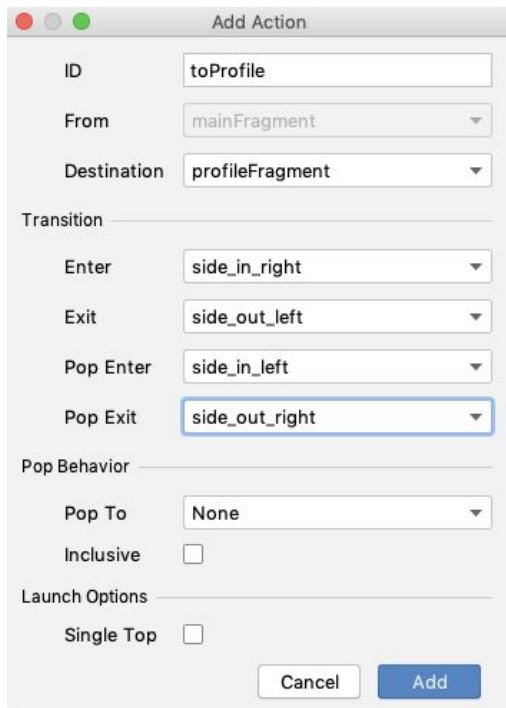
```
val bundle = Bundle()  
bundle.putString("firstName", firstName)  
bundle.putString("lastName", lastName)  
  
val profileFragment = ProfileFragment()  
  
supportFragmentManager().beginTransaction()  
    .add(R.id.container, profileFragment)  
    .commit()
```

### Receiver



```
val firstName = arguments.getString("firstName")  
val lastName = arguments.getString("lastName")
```

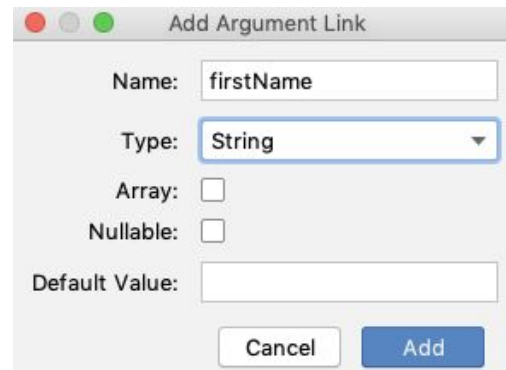
## Passing data between fragment with Fragment Navigation



The 'Add Action' dialog is shown with the following configuration:

- ID:** toProfile
- From:** mainFragment
- Destination:** profileFragment
- Transition:**
  - Enter: side\_in\_right
  - Exit: side\_out\_left
  - Pop Enter: side\_in\_left
  - Pop Exit: side\_out\_right
- Pop Behavior:**
  - Pop To: None
  - Inclusive: ☐
- Launch Options:**
  - Single Top: ☐

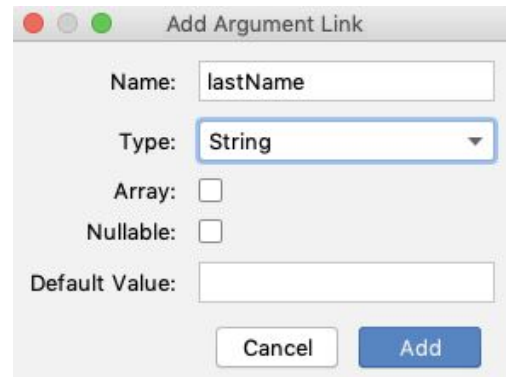
Buttons: Cancel, Add



The 'Add Argument Link' dialog for 'firstName' is shown with the following configuration:

- Name:** firstName
- Type:** String
- Array:** ☐
- Nullable:** ☐
- Default Value:** (empty)

Buttons: Cancel, Add



The 'Add Argument Link' dialog for 'lastName' is shown with the following configuration:

- Name:** lastName
- Type:** String
- Array:** ☐
- Nullable:** ☐
- Default Value:** (empty)

Buttons: Cancel, Add

# Passing data between fragment with Fragment Navigation

## Sender



```
findNavController().navigate(MainFragmentDirections.toProfile("hello", "password"))
```

## Receiver



```
val args = ProfileFragmentArgs.fromBundle(getArguments())
```

```
val firstName = args.firstName
```

```
val lastName = args.lastName
```



Explore the new way of implementing navigation in Android, to see how it works and how it could be applied to our existing applications

Successfully apply Fragment Navigation  
to one of our existing app

## Conclusion

- It helps us generate boilerplate code when navigating between fragments.
- Navigating to specific fragment becomes easier.
- It provides tools to work with the existing UI such as Toolbar or BottomNavigationView (Some of them don't work properly)

# Thanks!

## Contact Nimble

[nimblehq.co](https://nimblehq.co)

[hello@nimblehq.co](mailto:hello@nimblehq.co)

## Bangkok

399 Interchange 21 Sukhumvit Road, Unit  
#2402-03, Klong Toei, Wattana, Bangkok  
10110, Thailand

## Singapore

28C Stanley St, Singapore 068737

## Hong Kong

20th Floor, Central Tower  
28 Queen's Road, Central, Hong Kong

