Launch SDK Subsections

This page covers the capability of directly launching the desired sub-sections of the SDK to build a more seamless user experience

Prerequisites

It is recommended to read the Main Integration Guide section before going through this topic.

Deep Link Structure

```
https://upswing.access.partner/${partnerCode}?
action=webview&redirect=${route}
```

In the above deep link, there are two variables that are required to be passed:

- `partnerCode` : partner code (shared by Upswing)
- `route` : specific section route which you want to open inside the SDK

Bank Selection Homepage

To redirect the user to the tenure selection page and display a specific set of banks, use the following route:

Single Bank Selection

```
deeplink-manager/fd/FSI_FILTER?fsiList=UTKSIN
```

Multiple Bank Selection

```
val multipleBanksRoute = URLEncoder.encode("deeplink-
manager/fd/FSI_FILTER?fsiList=UTKSIN|SMCBIN", "UTF-8")
```

> ⓘ **NOTE**
>
> You need to encode the route using UTF-8 because the pipe (|) symbol is not supported directly in URLs.

List of Financial Institute Codes

Bank / NBFC	FI Code
Utkarsh Small Finance Bank	UTKSIN
Shivalik Small Finance Bank	SMCBIN
Suryoday Small Finance Bank	SURYIN
Shriram Finance	STFCIN
Bajaj Finance	BJFLIN

Bank / NBFC	FI Code
South Indian Bank	SOININ
slice small finance bank (North East)	NESFIN

My Investments section

```
deeplink-manager/fd/VIEW_INVESTMENTS
```

To open a specific tab in the My Investments section, you can pass the section as a query parameter:

- **Booked FDs tab:**

  ```
  deeplink-manager/fd/VIEW_INVESTMENTS?section=BOOKED
  ```

- **Pending journeys tab:**

  ```
  deeplink-manager/fd/VIEW_INVESTMENTS?section=PENDING
  ```

- **Matured FDs tab:**

  ```
  deeplink-manager/fd/VIEW_INVESTMENTS?section=CLOSED
  ```

- **View all transactions:**

  ```
  deeplink-manager/fd/VIEW_ALL_TRANSACTIONS
  ```

Help Center section

- **View tickets**

  ```
  deeplink-manager/fd/VIEW_TICKETS
  ```

- **Create new ticket flow**

  ```
  deeplink-manager/fd/CREATE_TICKET
  ```

- **Help center home**

  ```
  deeplink-manager/fd/OPEN_HELP_CENTER
  ```

Special FD Sections

There are special categories of FD which can be directly initialized as follows:

- **Senior Citizen FDs -** Opens the section to show senior citizen FD rates

  ```
  deeplink-manager/fd/VIEW_PLAN_CATEGORY?
  category=SENIOR_CITIZEN
  ```

- **Safest FDs -** Opens the section to display only Bank FDs where amount is insured upto INR 5 L

```
deeplink-manager/fd/VIEW_PLAN_CATEGORY?
category=BANKS_ONLY
```

- **High Return FDs -** Opens the section to display FDs which offer annual yield >9%

```
deeplink-manager/fd/VIEW_PLAN_CATEGORY?
category=YIELD_GREATER_THAN_9
```

- **Tax Saver FDs -** Opens the section for Tax Saver FD plans

```
deeplink-manager/fd/VIEW_PLAN_CATEGORY?
category=TAX_SAVER_FIXED_DEPOSIT
```

Implementation

Let's see this example to open help center home section via deep links

1. Define the help center deep link

```kotlin
val partnerCode = "ACME"

val helpCenterHomeDeeplink =
    "https://upswing.access.partner/${partnerCode}?
action=webview&redirect=deeplink-
manager/fd/OPEN_HELP_CENTER"
```

> ⓘ **NOTE**
>
> Do not forget to replace the Partner Code with the value assigned to your account. The deep link must follow the specified format and include the necessary parameters for redirection and identification.

2. Initialize the Upswing SDK

Initialize the Upswing SDK by creating an instance of `UpswingSdk.Builder` with the partner code provided to you. Finally, call `initializeSdkViaDeepLink` on the Upswing SDK instance, passing the help center home deep link as a URL object. This will launch the Upswing SDK and take the user directly to the Help Center home.

```kotlin
val upswingSdk = UpswingSdk.Builder(this, partnerCode,
PartnerInitCustomerImpl())
                        .setStatusBarColor(this.getColor(R.color.b
                        .setUIMode(UpswingTheme.SYSTEM)
                        .build()

upswingSdk.initializeSdkViaDeepLink(Uri.parse(helpCenterHomeDe
```