Moneytree LINK SDK v6 - how-to guides for the new features

Moneytree LINK SDK v6 offers many new features. Some of them require additional configuration to use. If you plan to utilize one of the following new features, please follow this document along with the README included with the SDK.

- Login Link
- LINK Kit

Login Link

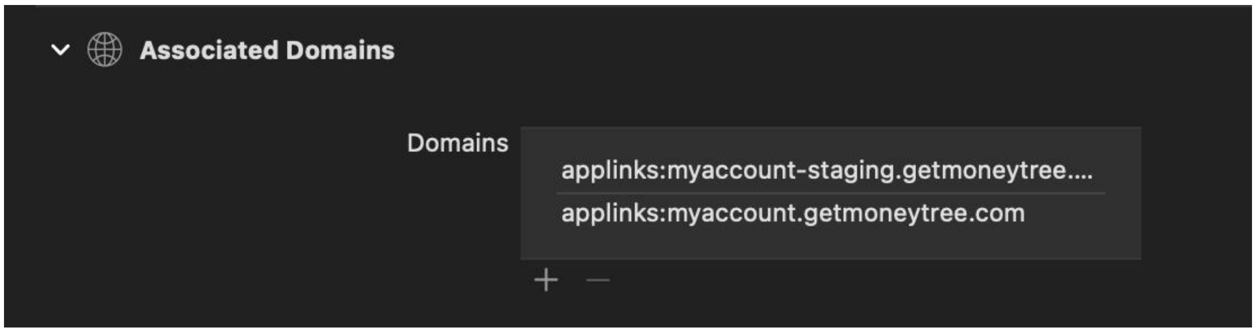
Login link works by taking users directly to your app from an email. In order to work properly, there are additional configuration steps for you and Moneytree.

iOS

In iOS, the Login Link feature is built on the universal link feature of iOS.

1. Add the following URLs to the *Associated Domains* section, under the *Signing and Capabilities* settings in your app target.

```
applinks:myaccount.getmoneytree.com
applinks:myaccount-staging.getmoneytree.com
```



applinks:myaccount-staging.getmoneytree....
applinks:myaccount.getmoneytree.com

2. Implement a method on *UIApplicationDelegate* to respond to a user action when they tap a link in an email. The following is an example from the sample application provided with the SDK. The point is to invoke *MTLApplicationDelegate.shared.application(_:userActivity:presentFrom:completion:)* in the function.

```
  func application(
    _ application: UIApplication,
    continue userActivity: NSUserActivity,
    restorationHandler: @escaping ([UIUserActivityRestoring]?) ->
Void
  ) -> Bool {
    guard userActivity.webpageURL?.host?.contains("getmoneytree.com")
?? false else {
      // If a given link doesn't have something to do with LINK SDK,
just ignore it for now.
      return false
    }
    return MTLApplicationDelegate.shared.application(
      application,
      userActivity: userActivity,
      presentFrom: viewController!
    ) { error in
      debugPrint(error)
    }
```

3. Share your App ID with the Moneytree Integration team because Moneytree must do some server configuration to support your app as well.

    - App ID
        - Consists of <*Your team id*>.<*Bundle identifier of your app*>. For example, `ABCDE12345.com.example.app`

4. That's it! The integration team will let you know after Moneytree's servers are updated.

See also
- https://developer.apple.com/documentation/xcode/supporting-associated-domains

Android

In Android, the Login Link feature is built on the *App Links* feature of Android. In order to pass the verification process, Moneytree's server should be able to verify your app using *Digital Asset Links*.

1. In the *AndroidManifest.xml*, please add the following *intent-filter* to your main Activity to enable App Links. Don't forget to define *autoVerify*.

   ```
   <!-- For production. This example assumes the first 5 characters
   of your production client ID are `abcde`. -->
   <intent-filter android:autoVerify="true">
      <action android:name="android.intent.action.VIEW" />
      <category android:name="android.intent.category.DEFAULT" />
      <category android:name="android.intent.category.BROWSABLE" />
      <data
          android:host="myaccount.getmoneytree.com"
          android:pathPrefix="/link/abcde"
          android:scheme="https" />
   </intent-filter>
   <!-- For staging. This example assumes the first 5 characters of
   your staging client ID is `hello`. -->
   <intent-filter android:autoVerify="true">
      <action android:name="android.intent.action.VIEW" />
      <category android:name="android.intent.category.DEFAULT" />
      <category android:name="android.intent.category.BROWSABLE" />
      <data
          android:host="myaccount-staging.getmoneytree.com"
          android:pathPrefix="/link/hello"
          android:scheme="https" />
   </intent-filter>
   ```

2. In the Activity you added above, override *onNewIntent* to consume the URL. The following is an example from the sample application provided with the SDK.

   ```
   // In the Activity you added intent-filters above
   override fun onNewIntent(intent: Intent) {
     super.onNewIntent(intent)
     // Handle login link action
     intent.data?.also { uri ->
       MoneytreeLink.instance.consumeLoginLink(this, uri)
     }
   }
   ```

3. Prepare the keystores of your app for both staging and production.

4. Don’t forget to enable the keystore **even when testing on staging**. Otherwise you can’t test on Staging. See also the following example in *build.gradle.*

```
signingConfigs {
  debug {
      storeFile file('./debug.keystore')
      storePassword 'debugStorePassword'
      keyAlias 'debugKeyAlias'
      keyPassword 'debugKeyPassword'
  }
   release { … }
}
```

5. Share *the package name* of your app and *the fingerprint of your signing certificate* in the keystore for the app for each environment with the Moneytree Integration team because Moneytree must do some server configuration to support your app.

    a. *Package name* is defined in *AndroidManifest.xml* and uniquely represents your app.
    b. You can find the fingerprint of your signing certificate by running the *keytool*. The hash type should be SHA256. See also the link.

6. That’s it! The integration team will let you know after Moneytree’s servers are updated.

See also
- https://developer.android.com/training/app-links
- https://developers.google.com/digital-asset-links/v1/statements#syntax

LINK Kit

iOS

No additional steps are needed to set up LINK Kit.

Android

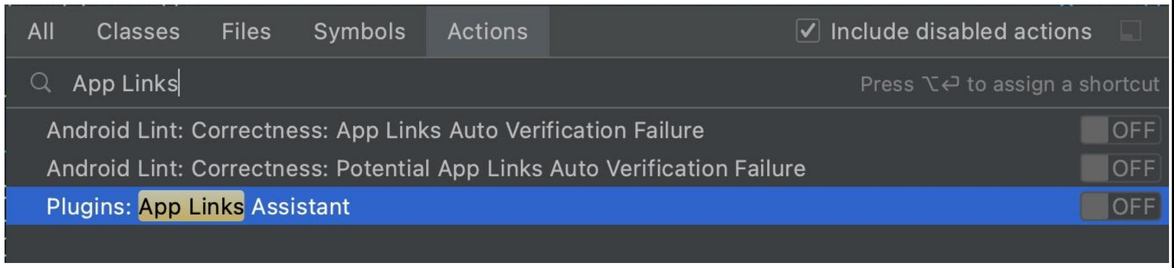
The LINK Kit should work in full-screen mode. In Android, it's built on a *Trusted Web Activity* (aka TWA).

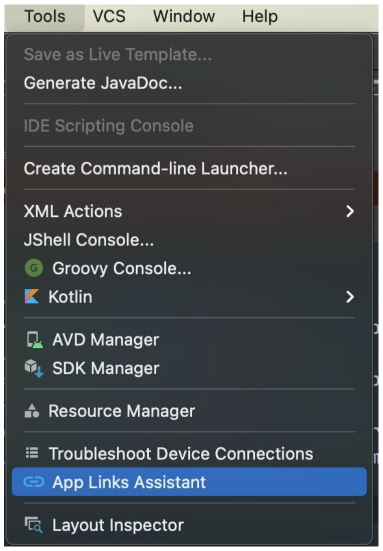
1. In order to use TWA, you and Moneytree have to set up *Digital Asset Links* first. Please follow the instructions above, **the sections from 3 to 6.**

2. Your Android project should have a statement like below. Note that it should have proper configuration per environment (production or staging).

```
<!-- res/value/strings.xml or anywhere to be activated properly -->
<!-- for staging env, subdomain should be linkkit-staging -->
<string name="asset_statements">
  [{
    \"relation\": [\"delegate_permission/common.handle_all_urls\"],
    \"target\": {
      \"namespace\": \"web\",
      \"site\": \"https://linkkit.getmoneytree.com\"}
  }]
</string>
```

You can generate the above xml using the *App Links Assistant* of Android Studio. In order to use it, follow the instructions below.

1	Check if your Android Studio enables it. If it's not enabled, like in the screenshot below, switch it to ON.  All Classes Files Symbols Actions ✓ Include disabled actions; App Links; Press ⌥⏎ to assign a shortcut; Android Lint: Correctness: App Links Auto Verification Failure OFF; Android Lint: Correctness: Potential App Links Auto Verification Failure OFF; Plugins: App Links Assistant OFF
2	(After you enable it from OFF in the previous step, you have to restart Android Studio)

2	Open *App Links Assistant* from the menu bar.	 Tools VCS Window Help Save as Live Template... Generate JavaDoc... IDE Scripting Console Create Command-line Launcher... XML Actions JShell Console... Groovy Console... Kotlin AVD Manager SDK Manager Resource Manager Troubleshoot Device Connections App Links Assistant Layout Inspector
3	Follow the instructions in the link and generate content. https://developer.android.com/studio/write/app-link-indexing#associatesite	

See also
- https://developer.chrome.com/docs/android/trusted-web-activity/quick-start/
- https://developers.google.com/digital-asset-links

Questions?

Configuring your app for these features will enable a better experience for your users. The Moneytree Integration team is here to help you with any technical questions you have.

When reporting an issue or asking a question, please share the following information so that we can quickly understand and respond to your request.

- Project configuration
    - In iOS, Build Settings in Xcode, or any information related to the steps described above.
    - In Android, AndroidManifest.xml or any information related to the steps above.
- What you achieved so far and what the issue is.
    - Expected behavior and actual behavior should be described.
    - How you debugged so far.

If possible, we can respond most effectively if you can reproduce your issue by configuring our sample app.