Search or jump to... / Pull requests Issues Codespaces Marketplace Explore

tonsafe / tonsafe Private

Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights Settings

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: main ← compare: ton-connect-2.0 ✓ Able to merge. These branches can be automatically merged.

Ton connect 2.0

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Remember, contributions to this repository should follow our GitHub Community Guidelines.

Reviewers
No reviews

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Use Closing keywords in the description to automatically close issues

Helpful resources
GitHub Community Guidelines

7 commits 90 files changed 1 contributor

Commits on Jan 10, 2023

Ton Connect 2.0 is in progress. fa66f1f
codxer007 committed on Jan 10

Commits on Jan 12, 2023

Ton Connect 2.0 initial connection made to the demo-app that ton-conn... 3951518
codxer007 committed on Jan 12

Commits on Jan 19, 2023

Work in progress 13ca278
codxer007 committed last month

Commits on Jan 25, 2023

Initial functionality done 3f1ddc1
codxer007 committed 3 weeks ago

Commits on Feb 7, 2023

Ton Connect 2.0 released for TonSafe v2.2 26cbdec
codxer007 committed last week

Commits on Feb 14, 2023

Ton Connect 2.0 integration - Modifications done as per wallets.json ... 8332914
codxer007 committed 6 hours ago

Some minor improvements 6b762ac
codxer007 committed 2 minutes ago

Showing 90 changed files with 7,926 additions and 976 deletions. Split Unified

1 App.tsx

```
@@ -4,6 +4,7 @@ import {getCurrentLanguage} from "./app/utils/language";
4 4  global.Buffer = global.Buffer || require('buffer').Buffer;
5 5  import { polyfillWebCrypto } from 'expo-standard-web-crypto';
6 6  import { withPreventScreenshots } from 'react-native-prevent-screenshots';
7    +
7 8  polyfillWebCrypto();
8 9
9 10 // Navigation
```

2 VERSION_CODE

```
@@ -1 +1 @@
1   - 164
    + 173
```

12 android/app/build.gradle

```
@@ -359,6 +359,18 @@ dependencies {
359 359      implementation jscFlavor
360 360  }
361 361
362 +
363 +  // SMOKED IMPLEMENTATION - DUPLICATE CLASSES ISSUE - https://youtrack.jetbrains.com/issue/KT-54136/Duplicated-classes-cause-build-failure-if-a-dependency-to-kotlin-stdlib-specified-in-an-android-project
364 +  // bug, https://github.com/facebook/react-native/issues/35979#issuecomment-1405377512
365 +  constraints{
366 +      implementation("org.jetbrains.kotlin:kotlin-stdlib-jdk7:1.8.0"){
367 +          because("kotlin-stdlib-jdk7 is now  part of kotlin-stdlib")
368 +      }
369 +      implementation("org.jetbrains.kotlin:kotlin-stdlib-jdk8:1.8.0"){
370 +          because("kotlin-stdlib-jdk8 is now  part of kotlin-stdlib")
```

```
371 +         }
372 +     }
373 +
362 374       implementation platform('com.google.firebase:firebase-bom:29.0.4')
363 375       implementation 'com.google.firebase:firebase-messaging:23.0.0'
364 376       implementation 'com.google.firebase:firebase-installations:17.0.0'
```

19 android/app/src/debug/java/com/tonhub/wallet/ReactNativeFlipper.java

```
@@ -36,15 +36,16 @@ public static void initializeFlipper(Context context, ReactInstanceManager react
36 36              client.addPlugin(new SharedPreferencesFlipperPlugin(context));
37 37              client.addPlugin(CrashReporterPlugin.getInstance());
38 38
39    -             NetworkFlipperPlugin networkFlipperPlugin = new NetworkFlipperPlugin();
40    -             NetworkingModule.setCustomClientBuilder(
41    -                     new NetworkingModule.CustomClientBuilder() {
42    -                         @Override
43    -                         public void apply(OkHttpClient.Builder builder) {
44    -                             builder.addNetworkInterceptor(new FlipperOkhttpInterceptor(networkFlipperPlugin));
45    -                         }
46    -                     });
47    -             client.addPlugin(networkFlipperPlugin);
   39 + // todo commented because of this issue https://github.com/binaryminds/react-native-sse/issues/3 https://github.com/facebook/flipper/issues/2495
   40 + //             NetworkFlipperPlugin networkFlipperPlugin = new NetworkFlipperPlugin();
   41 + //             NetworkingModule.setCustomClientBuilder(
   42 + //                     new NetworkingModule.CustomClientBuilder() {
   43 + //                         @Override
   44 + //                         public void apply(OkHttpClient.Builder builder) {
   45 + //                             builder.addNetworkInterceptor(new FlipperOkhttpInterceptor(networkFlipperPlugin));
   46 + //                         }
   47 + //                     });
   48 + //             client.addPlugin(networkFlipperPlugin);
48 49              client.start();
49 50
50 51              // Fresco Plugin needs to ensure that ImagePipelineFactory is initialized
```

112 android/app/src/main/AndroidManifest.xml

```
@@ -1,67 +1,47 @@
1    - <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2    -     package="com.tonsafe.wallet">
3    -
4    -     <uses-permission android:name="android.permission.INTERNET" />
5    -     <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
6    -     <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
7    -     <uses-permission android:name="android.permission.VIBRATE" />
8    -     <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
9    -
10   -     <queries>
11   -         <intent>
12   -             <action android:name="android.intent.action.VIEW" />
13   -             <category android:name="android.intent.category.BROWSABLE" />
14   -             <data android:scheme="https" />
15   -         </intent>
16   -     </queries>
17   -     <application
18   -         android:name=".MainApplication"
19   -         android:allowBackup="false"
20   -         android:icon="@mipmap/ic_launcher"
21   -         android:label="@string/app_name"
22   -         android:roundIcon="@mipmap/ic_launcher_round"
23   -         android:theme="@style/AppTheme"
24   -         android:usesCleartextTraffic="true">
25   -         <meta-data
26   -             android:name="expo.modules.updates.ENABLED"
27   -             android:value="false" />
28   -         <meta-data
29   -             android:name="expo.modules.updates.EXPO_SDK_VERSION"
30   -             android:value="44.0.0" />
31   -         <meta-data
32   -             android:name="expo.modules.updates.EXPO_UPDATES_CHECK_ON_LAUNCH"
33   -             android:value="ALWAYS" />
34   -         <meta-data
35   -             android:name="expo.modules.updates.EXPO_UPDATES_LAUNCH_WAIT_MS"
36   -             android:value="0" />
37   -         <meta-data
38   -             android:name="expo.modules.updates.EXPO_UPDATE_URL"
39   -             android:value="https://exp.host/@ex3ndr/wallet" />
40   -
41   -         <activity
42   -             android:name=".MainActivity"
43   -             android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|screenSize|smallestScreenSize|uiMode"
44   -             android:exported="true"
45   -             android:label="@string/app_name"
46   -             android:launchMode="singleTask"
47   -             android:screenOrientation="portrait"
48   -             android:theme="@style/Theme.App.SplashScreen"
49   -             android:windowSoftInputMode="adjustResize">
50   -             <intent-filter>
51   -                 <action android:name="android.intent.action.MAIN" />
52   -                 <category android:name="android.intent.category.LAUNCHER" />
53   -             </intent-filter>
54   -             <intent-filter>
55   -                 <action android:name="android.intent.action.VIEW" />
56   -
57   -                 <category android:name="android.intent.category.DEFAULT" />
58   -                 <category android:name="android.intent.category.BROWSABLE" />
59   -
60   -                 <data android:scheme="com.tonsafe.wallet" />
61   -             </intent-filter>
62   -         </activity>
63   -         <activity
64   -             android:name="com.facebook.react.devsupport.DevSettingsActivity"
65   -             android:exported="false" />
66   -     </application>
   1 + <manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.tonsafe.wallet">
   2 +   <uses-permission android:name="android.permission.INTERNET"/>
   3 +   <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
   4 +   <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
   5 +   <uses-permission android:name="android.permission.VIBRATE"/>
   6 +   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
   7 +   <queries>
   8 +     <intent>
   9 +       <action android:name="android.intent.action.VIEW"/>
  10 +       <category android:name="android.intent.category.BROWSABLE"/>
  11 +       <data android:scheme="https"/>
  12 +     </intent>
  13 +   </queries>
  14 +   <application android:name=".MainApplication" android:allowBackup="false" android:icon="@mipmap/ic_launcher" android:label="@string/app_name"
       android:roundIcon="@mipmap/ic_launcher_round" android:theme="@style/AppTheme" android:usesCleartextTraffic="true">
  15 +     <meta-data android:name="expo.modules.updates.ENABLED" android:value="false"/>
  16 +     <meta-data android:name="expo.modules.updates.EXPO_SDK_VERSION" android:value="44.0.0"/>
  17 +     <meta-data android:name="expo.modules.updates.EXPO_UPDATES_CHECK_ON_LAUNCH" android:value="ALWAYS"/>
```

```
+    <meta-data android:name="expo.modules.updates.EXPO_UPDATES_LAUNCH_WAIT_MS" android:value="0"/>
+    <meta-data android:name="expo.modules.updates.EXPO_UPDATE_URL" android:value="https://exp.host/@ex3ndr/wallet"/>
+    <activity android:name=".MainActivity" android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|screenSize|smallestScreenSize|uiMode"
  android:exported="true" android:label="@string/app_name" android:launchMode="singleTask" android:screenOrientation="portrait"
  android:theme="@style/Theme.App.SplashScreen" android:windowSoftInputMode="adjustResize">
+      <intent-filter>
+        <action android:name="android.intent.action.MAIN"/>
+        <category android:name="android.intent.category.LAUNCHER"/>
+      </intent-filter>
+      <intent-filter>
+        <action android:name="android.intent.action.VIEW"/>
+        <category android:name="android.intent.category.DEFAULT"/>
+        <category android:name="android.intent.category.BROWSABLE"/>
+        <data android:scheme="ton"/>
+        <data android:scheme="ton-subscriptions"/>
+        <data android:scheme="tonkeeper"/>
+        <data android:scheme="tonsafe"/>
+        <data android:scheme="tc"/>
+      </intent-filter>
+      <intent-filter android:autoVerify="true">
+        <action android:name="android.intent.action.VIEW"/>
+        <category android:name="android.intent.category.LAUNCHER"/>
+        <category android:name="android.intent.category.DEFAULT"/>
+        <category android:name="android.intent.category.BROWSABLE"/>
+        <data android:scheme="https"/>
+        <data android:host="app.tonkeeper.com"/>
+        <data android:host="app.tonsafe.net"/>
+      </intent-filter>
+    </activity>
+    <activity android:name="com.facebook.react.devsupport.DevSettingsActivity" android:exported="false"/>
+  </application>
  </manifest>
```

3,601 android/app/src/main/assets/index.android.bundle

Load diff

Large diffs are not rendered by default.

1 android/gradle.properties

```
@@ -24,6 +24,7 @@ android.useAndroidX=true

 # Automatically convert third-party libraries to use AndroidX
 android.enableJetifier=true
+
 # Version of flipper SDK to use with React Native
 FLIPPER_VERSION=0.125.0
```

9 app.json

```
@@ -3,7 +3,7 @@
     "name": "TonSafe Wallet",
     "slug": "tonsafe-wallet",
     "privacy": "unlisted",
-    "version": "2.1",
+    "version": "2.2.1",
     "orientation": "portrait",
     "icon": "./assets/tonbase_splash_screen.png",
     "splash": {
@@ -31,7 +31,11 @@
     "web": {
       "favicon": "./assets/favicon.png"
     },
-    "jsEngine": "hermes"
+    "jsEngine": "hermes",
+    "extra": {
+      "eas": {
+        "projectId": "3829cab9-18f4-4f80-a1aa-5bc507f925d9"
+      }
+    }
   }
 }
```

42 app/Navigation.tsx

```
@@ -1,6 +1,6 @@
 import * as React from 'react';
 import {createNativeStackNavigator} from '@react-navigation/native-stack';
-import {Platform, View, Image} from 'react-native';
+import {View, Image} from 'react-native';
 import {WelcomeFragment} from './fragments/onboarding/WelcomeFragment';
 import {WalletImportFragment} from './fragments/onboarding/WalletImportFragment';
 import {WalletCreateFragment} from './fragments/onboarding/WalletCreateFragment';
@@ -33,17 +33,12 @@ import {registerForPushNotificationsAsync, registerPushToken} from './utils/regi
 import * as Notifications from 'expo-notifications';
 import {PermissionStatus} from 'expo-modules-core';
 import {t} from './i18n/t';
-import {AuthenticateFragment} from './fragments/secure/AuthenticateFragment';
 import {ConnectionsFragment} from './fragments/connections/ConnectionsFragment';
 import axios from 'axios';
-// import {NeocryptoFragment} from './fragments/integrations/NeocryptoFragment';
-// import {StakingTransferFragment} from './fragments/staking/StakingTransferFragment';
-// import {StakingFragment} from './fragments/staking/StakingFragment';
 import {SignFragment} from './fragments/secure/SignFragment';
 import {TransferFragment} from './fragments/secure/TransferFragment';
 import {createEngine} from './engine/createEngine';
 import {useRecoilCallback} from 'recoil';
-// import {AppFragment} from './fragments/apps/AppFragment';
 import {DevStorageFragment} from './fragments/dev/DevStorageFragment';
 import {getCurrenciesPrices, setDefaultCurrency, saveTONPriceAtStart} from "./utils/currency";
 import {LogoutFragment} from "./fragments/logout/logout";
@@ -67,8 +62,6 @@ import {SMSFragment} from "./fragments/sms/SMSFragment";

 const Stack = createNativeStackNavigator();

-// const Stack = Platform.OS === 'ios' ? createNativeStackNavigator() : createStackNavigator();
-
 function fullScreen(name: string, component: React.ComponentType<any>) {
     return (
         <Stack.Screen
@@ -84,6 +77,6 @@ function genericScreen(name: string, component: React.ComponentType<any>) {
     );
 }
```

```
- function formSheetScreen(name: string, component: React.ComponentType<any>) {
-     return (
-         <Stack.Screen
-             key={`formSheetScreen-${name}`}
-             name={name}
-             component={component}
-             options={{headerShown: false}}
-         />
-     );
- }
-
function modalScreen(name: string, component: React.ComponentType<any>) {
    return (
        <Stack.Screen
@@ -124,17 +106,6 @@ function lockedModalScreen(name: string, component: React.ComponentType<any>) {
    );
}

- // function fullScreenModal(name: string, component: React.ComponentType<any>) {
- //     return (
- //         <Stack.Screen
- //             key={`fullScreenModal-${name}`}
- //             name={name}
- //             component={component}
- //             options={{ presentation: 'fullScreenModal', headerShown: false }}
- //         />
- //     );
- // }
-
const navigation = [
    fullScreen('Welcome', WelcomeFragment),
    fullScreen('Home', HomeFragment),
@@ -170,7 +141,6 @@ const navigation = [
    modalScreen('CreateInvoice', CreateInvoiceFragment),
    modalScreen('CreateInvoiceLink', CreateInvoiceLinkFragment),
    modalScreen('Transaction', TransactionPreviewFragment),
-   modalScreen('Authenticate', AuthenticateFragment),
    modalScreen('Sign', SignFragment),
    modalScreen('Migration', MigrationFragment),
    modalScreen('SpamFilter', SpamFilterFragment),
@@ -181,17 +151,11 @@ const navigation = [
    lockedModalScreen('Scanner', ScannerFragment),
    genericScreen('DeveloperTools', DeveloperToolsFragment),
    genericScreen('DeveloperToolsStorage', DevStorageFragment),
-   // lockedModalScreen('Buy', NeocryptoFragment),
-   // fullScreen('Staking', StakingFragment),
-   // modalScreen('StakingTransfer', StakingTransferFragment),
-   // modalScreen('App', AppFragment),
];

export const Navigation = React.memo(() => {
    const safeArea = useSafeAreaInsets();
-
    const recoilUpdater = useRecoilCallback<[any, any], any>(({set}) => (node, value) => set(node, value));
-
    const engine = React.useMemo(() => {
        let state = getAppState();
        if (0 <= state.selected && state.selected < state.addresses.length) {
@@ -212,8 +176,8 @@ export const Navigation = React.memo(() => {
    const initial = React.useMemo(() => {
        const onboarding = resolveOnboarding(engine);
        setDefaultCurrency(); // If no currency is found then set a default currency.
-       getCurrenciesPrices(); //  Get currency prices for all supported currencies.
-       saveTONPriceAtStart(); // LOCK TON-USD price for static conversions.
+       getCurrenciesPrices().then(); //  Get currency prices for all supported currencies.
+       saveTONPriceAtStart().then(); // LOCK TON-USD price for static conversions.
        setSecretRemainderCounter(); // Secret reminder setup
        if (onboarding === 'backup') {
            return 'WalletCreated';
```

13 app/components/ConnectedAppButton.tsx

```
@@ -2,28 +2,23 @@ import React from "react";
import {View, Text, Pressable, Linking} from "react-native";
import { t } from "../i18n/t";
import { Theme } from "../Theme";
- import { useEngine } from "../engine/Engine";
import {WImage} from "./WImage";
- import {useTypedNavigation} from "../utils/useTypedNavigation";

export const ConnectedAppButton = React.memo((
    {
        name,
        url,
+       iconUrl,
        onRevoke,
        enableOpen
    }: {
        name: string,
        url: string,
+       iconUrl: string,
        onRevoke?: () => void,
        enableOpen?: boolean
    }
) => {
-
-   const engine = useEngine();
-   const app = engine.products.deApps.useAppData(url);
-   const navigation = useTypedNavigation();
-
    function openURL() {
        if (enableOpen) {
            Linking.openURL(url);
@@ -41,8 +36,8 @@ export const ConnectedAppButton = React.memo((
                <WImage
                    heigh={42}
                    width={42}
-                   src={app?.image?.preview256}
-                   blurhash={app?.image?.blurhash}
+                   src={iconUrl}
+                   blurhash={iconUrl}
                    style={{ marginRight: 10 }}
                    borderRadius={8}
                />
```

2 app/components/ItemButton.tsx

```
@@ -15,7 +15,7 @@ export const ItemButton = React.memo((props: {
        <Pressable style={({iProps}) => ({opacity: iProps.pressed ? 0.3 : 1, flexDirection: 'row', alignItems: 'center', width: (props.width) ? props.width : '100%'})} onPress={props.onPress}>
```

```
16  16              <View style={{height: 48, paddingLeft: props.leftIcon ? 8 : 16, paddingRight: 16, alignItems: 'center', justifyContent: 'center',
    flexDirection: 'row', flexGrow: 1, flexBasis: 0 }}>
17  17                  <View style={{ flexGrow: 1, flexShrink: 1, flexDirection: 'row', alignItems: 'center' }}>
18      -                   {props.leftIcon && (<Image style={{ height: 24, width: 24, tintColor: (props.disableTint) ? '' : Theme.tintColor }} source=
    {props.leftIcon} />)}
        18  +               {props.leftIcon && (<Image style={{ height: 24, width: 24, tintColor: (props.disableTint) ? undefined : Theme.tintColor }} source=
    {props.leftIcon} />)}
19  19                      <Text
20  20                          style={{
21  21                              fontSize: 17,
```

app/fragments/MoreSettingsFragment.tsx 24

```
@@ -176,18 +176,18 @@ export const MoreSettingsFragment = fragment(() => {
176 176                 </View>
177 177             </View>
178 178
179     -           <View style={{
180     -               marginBottom: 10, marginTop: 5,
181     -               backgroundColor: Theme.item,
182     -               borderRadius: Theme.borderRadius,
183     -               justifyContent: 'center',
184     -               alignItems: 'center',
185     -               flexShrink: 1,
186     -           }}>
187     -               <View style={{ marginHorizontal: 16, width: '100%' }}>
188     -                   <ItemButton leftIcon={require('../../assets/extensions.png')} title={t('home.extensions')} onPress={() =>
    navigation.navigate('Extensions')} />
189     -               </View>
190     -           </View>
        179 +           {/*<View style={*/}
        180 +           {/*    marginBottom: 10, marginTop: 5,*/}
        181 +           {/*    backgroundColor: Theme.item,*/}
        182 +           {/*    borderRadius: Theme.borderRadius,*/}
        183 +           {/*    justifyContent: 'center',*/}
        184 +           {/*    alignItems: 'center',*/}
        185 +           {/*    flexShrink: 1,*/}
        186 +           {/*}}>*/}
        187 +           {/*    <View style={{ marginHorizontal: 16, width: '100%' }}>*/}
        188 +           {/*        <ItemButton leftIcon={require('../../assets/extensions.png')} title={t('home.extensions')} onPress={() =>
    navigation.navigate('Extensions')} />*/}
        189 +           {/*    </View>*/}
        190 +           {/*</View>*/}
191 191
192 192               <View style={{
193 193                   marginBottom: 10, marginTop: 5,
```

app/fragments/PrimeHomeFragment.tsx 11

```
@@ -19,12 +19,13 @@ import BN from 'bn.js';
19  19  import {InformationFragment} from "./information/InformationFragment";
20  20  import {getDarkModeValue} from "../utils/misc";
21  21  import {SecretRemindersFragment} from "./secret-reminders/SecretRemindersFragment";
        22  + import ExternalAppConnection from "../tonconnect/connection-view/ExternalAppConnection";
        23  + import {TonConnectRemoteBridge} from "../tonconnect";
22  24
23  25  export const PrimeHomeFragment = fragment(() => {
24  26      const safeArea = useSafeAreaInsets();
25  27      const [tab, setTab] = React.useState(0);
26  28      const [infoTab, infoTabRefresher] = React.useState(0);
27      -   const [scrollToTop, setScrollToTop] = React.useState(0);
28  29      const navigation = useTypedNavigation();
29  30      const loader = useGlobalLoader()
30  31      const engine = useEngine();

@@ -111,12 +112,9 @@ export const PrimeHomeFragment = fragment(() => {
111 112                         });
112 113                     }
113 114                 }
114     -               if (resolved && resolved.type === 'connect') {
        115 +           if (resolved && resolved.type === 'ton-connect') {
115 116                     SplashScreen.hideAsync();
116     -                   navigation.navigate('Authenticate', {
117     -                       session: resolved.session,
118     -                       endpoint: resolved.endpoint
119     -                   });
        117 +               TonConnectRemoteBridge.handleConnectDeeplink(resolved.query).then();
120 118               }
121 119           }
122 120       });

@@ -126,6 +124,7 @@ export const PrimeHomeFragment = fragment(() => {
126 124       return (
127 125           <View style={{flexGrow: 1}}>
128 126               <SecretRemindersFragment></SecretRemindersFragment>
        127 +           <ExternalAppConnection></ExternalAppConnection>
129 128               <View style={{flexGrow: 1}}/>
130 129               <StatusBar style={(getDarkModeValue()) ? 'light' : 'dark'}/>
131 130               <View style={{position: 'absolute', top: 0, left: 0, right: 0, bottom: 0, opacity: tab === 0 ? 1 : 0}} pointerEvents={tab === 0 ? 'box-none' :
    'none'}>
```

app/fragments/connections/ConnectionsFragment.tsx 64

```
@@ -1,4 +1,3 @@
1       - import axios from 'axios';
2   1   import { StatusBar } from 'expo-status-bar';
3   2   import * as React from 'react';
4   3   import { Alert, Platform, Text, View } from 'react-native';

@@ -9,43 +8,36 @@ import { CloseButton } from '../../components/CloseButton';
9   8   import { ConnectedAppButton } from '../../components/ConnectedAppButton';
10  9   import { fragment } from '../../fragment';
11  10  import { t } from '../../i18n/t';
12      - import { addPendingRevoke, getConnectionReferences, removeConnectionReference, removePendingRevoke } from "../../storage/appState";
13  11  import { Theme } from '../../Theme';
14      - import { backoff } from '../../utils/time';
15  12  import { useTypedNavigation } from '../../utils/useTypedNavigation';
16      - import {getDarkModeValue} from "../../utils/misc";
        13  + import {getConnectedAppsReference, getDarkModeValue} from "../../utils/misc";
        14  + import {IConnectedApp, IConnectedAppConnection} from "../../store";
        15  + import {TonConnect} from "../../tonconnect";
        16  +
        17  + type connectedAppsType = { connectedApp: IConnectedApp | null; connection: IConnectedAppConnection | null; }[];
17  18
18  19  export const ConnectionsFragment = fragment(() => {
19  20      const safeArea = useSafeAreaInsets();
20  21      const navigation = useTypedNavigation();
21      -   let [apps, setApps] = React.useState(getConnectionReferences());
22      -   let disconnectApp = React.useCallback((src: string) => {
23      -
24      -       let refs = getConnectionReferences();
25      -       let ex = refs.find((v) => v.key === src);
```

```
-        if (!ex) {
-            return;
-        }
+    const [connectedApps, setConnectedApps] = React.useState<connectedAppsType>(getConnectedAppsReference());

-        Alert.alert(t('auth.revoke.title'), t('auth.revoke.message'), [
-            {
-                text: t('common.cancel')
-            },
-            {
-                text: t('auth.revoke.action'), style: 'destructive', onPress: () => {
-                    addPendingRevoke(src);
-                    removeConnectionReference(src);
-                    setApps((a) => a.filter((v) => v.key !== src));
-                    backoff('revoke', async () => {
-                        await axios.post('https://connect.tonhubapi.com/connect/revoke', { key: src }, { timeout: 5000 });
-                        removePendingRevoke(src);
-                    });
+    let disconnectApp = React.useCallback((url: string | undefined) => {
+        if (url) {
+            Alert.alert(t('auth.revoke.title'), t('auth.revoke.message'), [
+                {
+                    text: t('common.cancel')
+                },
+                {
+                    text: t('auth.revoke.action'), style: 'destructive', onPress: async () => {
+                        await TonConnect.disconnect(url);
+                        setConnectedApps(getConnectedAppsReference());
+                    }
                 }
-            }
-        ]);
+            ]);
+        }

    }, []);
-   if (apps.length === 0) {
+   if (connectedApps.length === 0) {
        return (
            <View style={{
                flexGrow: 1, flexBasis: 0,
```

@@ -70,7 +62,8 @@ export const ConnectionsFragment = fragment(() => {

```
                    }, { textAlign: 'center' }]}>{t('auth.apps.title')}</Text>
                </View>
            )}
-           <Text style={{ fontSize: 24, textAlign: 'center', color: Theme.textSecondary }}>{t('auth.noApps')}</Text>
+           <Text style={{ fontSize: 24, textAlign: 'center', color: Theme.textSecondary }}>{t('auth.noApps') + '\n'}</Text>
+           <Text style={{ fontSize: 14, textAlign: 'center', color: Theme.textSecondary }}>{t('applications.infoMessage')}</Text>
            {Platform.OS === 'ios' && (
                <CloseButton
                    style={{ position: 'absolute', top: 12, right: 10 }}
```

@@ -112,16 +105,19 @@ export const ConnectionsFragment = fragment(() => {

```
                    alignItems: 'center',
                    flexShrink: 1,
                }}>
-                   {apps.map((app) => (
-                       <View key={`app-${app.key}`} style={{ marginHorizontal: 16, width: '100%', marginBottom: 8 }}>
+                   {connectedApps.map((app: { connectedApp: IConnectedApp | null; connection: IConnectedAppConnection | null; }) => (
+                       <View key={`app-${Math.random() * 1000}`} style={{ marginHorizontal: 16, width: '100%', marginBottom: 8 }}>
                            <ConnectedAppButton
-                               onRevoke={() => disconnectApp(app.key)}
-                               url={app.url}
-                               name={app.name}
+                               onRevoke={() => disconnectApp(app.connectedApp?.url)}
+                               url={(app.connectedApp) ? app.connectedApp.url : 'N/A'}
+                               name={(app.connectedApp) ? app.connectedApp.name : 'N/A'}
+                               iconUrl={(app.connectedApp) ? app.connectedApp.icon : ''}
+                               enableOpen={!!(app.connectedApp && app.connectedApp.url)}
                            />
                        </View>
                    ))}
                </View>
+               <Text style={{ fontSize: 14, textAlign: 'center', color: Theme.textSecondary, marginHorizontal: 10, marginVertical: 5 }}>{t('applications.infoMessage')}</Text>
            </ScrollView>
            {Platform.OS === 'ios' && (
                <CloseButton
```

app/fragments/extensions/ExtensionsFragment.tsx

@@ -84,7 +84,8 @@ export const ExtensionsFragment = systemFragment(() => {

```
                    }, { textAlign: 'center' }]}>{t('auth.apps.title')}</Text>
                </View>
            )}
-           <Text style={{ fontSize: 18, textAlign: 'center', color: Theme.textColor }}>{t('applications.noAppsTxt')}</Text>
+           <Text style={{ fontSize: 18, textAlign: 'center', color: Theme.textColor }}>{t('applications.noAppsTxt') + '\n'}</Text>
+           <Text style={{ fontSize: 14, textAlign: 'center', color: Theme.textColor }}>{t('applications.infoMessage')}</Text>
            {Platform.OS === 'ios' && (
                <CloseButton
                    style={{ position: 'absolute', top: 12, right: 10 }}
```

@@ -134,6 +135,7 @@ export const ExtensionsFragment = systemFragment(() => {

```
                            url={app.url}
                            name={app.name}
                            enableOpen={true}
+                           iconUrl={app.url}
                        />
                    </View>
                ))}
```

app/fragments/onboarding/WalletCreatedFragment.tsx

@@ -26,7 +26,6 @@ export const WalletCreatedFragment = systemFragment(() => {

```
    function finalize() {
        if (!(loose && share && responsibility && accountDeletion)) {
            if (scrollRef) {
-               console.log('SCROLLED');
                scrollRef.scrollToEnd({animated: true});
            }
        } else {
```

app/fragments/secure/AuthenticateFragment.tsx

Load diff

This file was deleted.

20 app/fragments/utils/ScannerFragment.tsx

```
@@ -13,6 +13,8 @@ import { t } from '../../i18n/t';
import { systemFragment } from '../../systemFragment';
import { RoundButton } from '../../components/RoundButton';
import {Theme} from "../../Theme";
+import {TonConnectRemoteBridge} from "../../tonconnect";
+import queryString from 'query-string';

export const ScannerFragment = systemFragment(() => {
    const safeArea = useSafeAreaInsets();
@@ -38,13 +40,23 @@ export const ScannerFragment = systemFragment(() => {

    useEffect(() => {
        if (barcodes && barcodes.length > 0 && barcodes[0]) {
+            console.log(barcodes, route);
            if (route && (route as any).callback) {
                setActive(false);

                setTimeout(() => {
                    navigation.goBack();
-                    if (barcodes[0].content.type === BarcodeValueType.URL) {
-                        (route as any).callback(barcodes[0].content.data.url);
+                    if ((barcodes[0].content.type === BarcodeValueType.URL) ||
+                        ((barcodes[0].content.type === BarcodeValueType.TEXT) && (barcodes[0].content.data.includes('tc://')))) {
+                        let endpoint: string | undefined = undefined;
+                        if (barcodes[0].content.type === BarcodeValueType.URL) {
+                            endpoint = barcodes[0].content.data.url;
+                        } else {
+                            endpoint = barcodes[0].content.data;
+                        }
+                        if (endpoint) {
+                            const query: any = queryString.parseUrl(endpoint).query;
+                            TonConnectRemoteBridge.handleConnectDeeplink(query).then();
+                        }
                    } else if (barcodes[0].content.type === BarcodeValueType.TEXT) {
                        (route as any).callback(barcodes[0].content.data);
                    } else if (barcodes[0].content.type === BarcodeValueType.UNKNOWN) {
@@ -178,7 +190,7 @@ export const ScannerFragment = systemFragment(() => {
                        opacity: props.pressed ? 0.5 : 1,
                    }
                })}
-                onPress={() => { setFlashOn(!flashOn); }}
+                        onPress={() => { setFlashOn(!flashOn); }}
                >
                    <Image style={{ height: 61, width: 61 }} source={flashOn ? require('../../../assets/ic_flash_on.png') : require('../../../assets/ic_flash.png')} />
                </Pressable>
```

24 app/fragments/wallet/WalletFragment.tsx

```
@@ -25,13 +25,15 @@ import {loading} from '../../components/LoadingIndicator';
import {log} from '../../utils/log';
import {Engine, useEngine} from '../../engine/Engine';
import {WalletState} from '../../engine/products/WalletProduct';
-import {useCallback, useState} from "react";
+import {useCallback, useEffect, useState} from "react";
import Clipboard from "@react-native-clipboard/clipboard";
import * as Haptics from "expo-haptics";
import * as Animatable from 'react-native-animatable';
import {TonPriceHistoryGraphFragment} from "../graph/TonPriceHistoryGraphFragment";
-import {AppMessageFragment} from "../app-message/AppMessageFragment";
+import {AppMessageFragment} from "../app-message/appMessageFragment";
import {I18nManager} from 'react-native';
+import VerifyTransactionModal from "../../tonconnect/transactions/VerifyTransactionModal";
+import {setWalletStateInit} from "../../utils/misc";

const WalletTransactions = React.memo((props: {
    txs: { id: string, time: number }[],
@@ -102,6 +104,9 @@ function WalletComponent(props: { wallet: WalletState }) {
    const engine = useEngine();
    const syncState = engine.state.use();
    const account = props.wallet;
+    useEffect(() => {
+        setWalletStateInit().then(); // Set wallet state init.
+    }, []);
    //
    // Transactions
    //
@@ -244,13 +249,7 @@ function WalletComponent(props: { wallet: WalletState }) {
                    });
                }
            }
-            if (res && res.type === 'connect') {
-                // if QR is valid navigate to sign fragment
-                navigation.navigate('Authenticate', {
-                    session: res.session,
-                    endpoint: res.endpoint
-                });
-            }
+            if (res && res.type === 'connect') {}
        } catch (error) {
            // Ignore
        }
@@ -315,6 +314,7 @@ function WalletComponent(props: { wallet: WalletState }) {

    return (
        <View style={{flexGrow: 1, paddingBottom: safeArea.bottom}}>
+            <VerifyTransactionModal></VerifyTransactionModal>
            <Animated.ScrollView
                contentContainerStyle={{
                    flexGrow: 1,
@@ -545,7 +545,7 @@ function WalletComponent(props: { wallet: WalletState }) {
                    </View>
                    <View style={{flexGrow: 1, flexBasis: 0, marginRight: 8, backgroundColor: Theme.item, borderRadius: Theme.borderRadius}}>
                        <TouchableHighlight underlayColor={Theme.selector} style={{borderRadius: Theme.borderRadius}} onPress={() => {
-                            navigation.navigate('Information', {fullHeight: true})
+                            navigation.navigate('Connections')
                        }}>
                            <View style={{
                                justifyContent: 'center',
@@ -557,8 +557,8 @@ function WalletComponent(props: { wallet: WalletState }) {
                                flexWrap: 'wrap',
                                paddingTop: 6
                            }}>
-                                <Image source={require('../../../assets/homepage-tonbase-icons.png')} style={{width: 20, height: 20, marginRight: 7, tintColor: Theme.tintColor}}></Image>
-                                <Text style={{fontSize: 14, color: Theme.accentText, textAlignVertical: 'center'}}>{t('common.infoTabTitle')}</Text>
+                                <Image source={require('../../../assets/homepage-tonbase-icons.png')} style={{width: 20, height: 20, marginRight: 7, tintColor: Theme.tintColor}}></Image>
+                                <Text style={{fontSize: 14, color: Theme.accentText, textAlignVertical: 'center'}}>{t('common.infoTabTitle')}</Text>
```

```
560 +                                <Image source={require('../../../assets/ic_import.png')} style={{width: 20, height: 20, marginRight: 7, tintColor:
    Theme.tintColor}}></Image>
561 +                                <Text style={{fontSize: 14, color: Theme.accentText, textAlignVertical: 'center'}}>{'dApps'}</Text>
562 562                            </View>
563 563                        </TouchableHighlight>
564 564                    </View>
```

107 app/fragments/wallet/products/ProductsComponent.tsx

```
... ... @@ -1,6 +1,6 @@
1 1   import BN from "bn.js"
2 2   import React from "react"
3   - import {View, Text, Alert, Pressable} from "react-native"
  3 + import {View} from "react-native"
4 4   import {ProductButton} from "./ProductButton"
5 5   import {useEngine} from "../../../engine/Engine"
6 6   import SignIcon from '../../../../assets/ic_sign.svg';
    @@ -9,71 +9,55 @@ import {useTypedNavigation} from "../../../utils/useTypedNavigation"
9 9   import {AppConfig} from "../../../AppConfig"
10 10  import {t} from "../../../i18n/t"
11 11  import {JettonProdcut} from "./JettonProduct";
12   + import {useRemoteBridge} from "../../../tonconnect";
12 13
13 14  export const ProductsComponent = React.memo(() => {
14 15      const navigation = useTypedNavigation();
15 16      const engine = useEngine();
16   -     const oldWalletsBalance = engine.products.legacy.useState();
17   -     // const pool = engine.products.whalesStakingPool.useState();
18 17      const currentJob = engine.products.apps.useState();
  18 +     useRemoteBridge();
19 19      const jettons = engine.products.main.useJettons().filter((j) => !j.disabled);
20   -     const nothingToShow = (oldWalletsBalance.lte(new BN(0))) && !currentJob && (jettons.length <= 0);
21   -
22   -     let removeExtension = React.useCallback((key: string) => {
23   -         Alert.alert(t('auth.apps.delete.title'), t('auth.apps.delete.message'), [{ text: t('common.cancel') }, {
24   -             text: t('common.delete'),
25   -             style: 'destructive',
26   -             onPress: () => {
27   -                 engine.products.deApps.removeApp(key);
28   -             }
29   -         }]);
30   -     }, []);
31 20
32 21      return (
33 22          <View style={{paddingTop: 8}}>
34   -             {/*{nothingToShow && (*/}
35   -             {/*    <View style={{display: 'flex', paddingHorizontal: 20}}>*/}
36   -             {/*        <Text style={{textAlign: 'center', textAlignVertical: 'center', fontSize: 14, color: Theme.accentText, letterSpacing: 1.5,
    fontWeight: '600'}}>{t('applications.noAppsTxt')}</Text>*/}
37   -             {/*    </View>*/}
23  +             {/*{currentJob && currentJob.job.type === 'transaction' && (*/}
24  +             {/*    <ProductButton*/}
25  +             {/*        name={t('products.transactionRequest.title')}*/}
26  +             {/*        subtitle={t('products.transactionRequest.subtitle')}*/}
27  +             {/*        icon={TransactionIcon}*/}
28  +             {/*        value={null}*/}
29  +             {/*        onPress={() => {*/}
30  +             {/*            if (currentJob.job.type === 'transaction') {*/}
31  +             {/*                navigation.navigateTransfer({*/}
32  +             {/*                    order: {*/}
33  +             {/*                        target: currentJob.job.target.toFriendly({testOnly: AppConfig.isTestnet}),*/}
34  +             {/*                        amount: currentJob.job.amount,*/}
35  +             {/*                        payload: currentJob.job.payload,*/}
36  +             {/*                        stateInit: currentJob.job.stateInit,*/}
37  +             {/*                        amountAll: false*/}
38  +             {/*                    },*/}
39  +             {/*                    job: currentJob.jobRaw,*/}
40  +             {/*                    text: currentJob.job.text*/}
41  +             {/*                });*/}
42  +             {/*            }*/}
43  +             {/*        }}*/}
44  +             {/*    />*/}
45  +             {/*)}*/}
46  +             {/*{currentJob && currentJob.job.type === 'sign' && (*/}
47  +             {/*    <ProductButton*/}
48  +             {/*        name={t('products.signatureRequest.title')}*/}
49  +             {/*        subtitle={t('products.signatureRequest.subtitle')}*/}
50  +             {/*        icon={SignIcon}*/}
51  +             {/*        value={null}*/}
52  +             {/*        onPress={() => {*/}
53  +             {/*            if (currentJob.job.type === 'sign') {*/}
54  +             {/*                navigation.navigate('Sign', {*/}
55  +             {/*                    job: currentJob.jobRaw*/}
56  +             {/*                });*/}
57  +             {/*            }*/}
58  +             {/*        }}*/}
59  +             {/*    />*/}
60  +             {/*)}*/}
38  +             {currentJob && currentJob.job.type === 'transaction' && (
39  -                 <ProductButton
40  -                     name={t('products.transactionRequest.title')}
41  -                     subtitle={t('products.transactionRequest.subtitle')}
42  -                     icon={TransactionIcon}
43  -                     value={null}
44  -                     onPress={() => {
45  -                         if (currentJob.job.type === 'transaction') {
46  -                             navigation.navigateTransfer({
47  -                                 order: {
48  -                                     target: currentJob.job.target.toFriendly({testOnly: AppConfig.isTestnet}),
49  -                                     amount: currentJob.job.amount,
50  -                                     payload: currentJob.job.payload,
51  -                                     stateInit: currentJob.job.stateInit,
52  -                                     amountAll: false
53  -                                 },
54  -                                 job: currentJob.jobRaw,
55  -                                 text: currentJob.job.text
56  -                             });
57  -                         }
58  -                     }}
59  -                 />
60  -             )}
61  -             {currentJob && currentJob.job.type === 'sign' && (
62  -                 <ProductButton
63  -                     name={t('products.signatureRequest.title')}
64  -                     subtitle={t('products.signatureRequest.subtitle')}
65  -                     icon={SignIcon}
66  -                     value={null}
67  -                     onPress={() => {
68  -                         if (currentJob.job.type === 'sign') {
69  -                             navigation.navigate('Sign', {
70  -                                 job: currentJob.jobRaw
71  -                             });
72  -                         }
73  -                     }}
74  -                 />
75  -             )}
```

```
75        -                />
76        -            ))}
77    61
78    62              {jettons.map((jt) => (
79    63                  (jt.balance.gt(new BN(0))) ? <JettonProdcut
      @@ -83,17 +67,6 @@ export const ProductsComponent = React.memo(() => {
83    67                      engine={engine}
84    68                  /> : <></>
85    69              ))}
86        -            {/*{pool && (<StakingProductComponent pool={pool} />)}*/}
87        -            {/*{oldWalletsBalance.gt(new BN(0)) && (*/}
88        -            {/*    <ProductButton*/}
89        -            {/*        name={t('products.oldWallets.title')}*/}
90        -            {/*        subtitle={t("products.oldWallets.subtitle")}*/}
91        -            {/*        icon={OldWalletIcon}*/}
92        -            {/*        value={oldWalletsBalance}*/}
93        -            {/*        onPress={() => navigation.navigate('Migration')}*/}
94        -            {/*        style={{marginVertical: 4}}*/}
95        -            {/*    />*/}
96        -            {/*)}*/}
97    70          </View>
98    71      )
99    72  })
```

app/i18n/i18n_ar.ts

```
      @@ -162,6 +162,8 @@ logout: {
162   162      },
163   163      applications: {
164   164          noAppsTxt: 'لا توجد تطبيقات متصلة.',
165       +        infoMessage: 'عند توصيل التطبيقات ، يرجى تحديد "TON Connect".',
166       +        cannotConnectMessage: 'موقع الويب المحدد يستخدم إصدار TON Connect قديم، وبالتالي لا يمكن إنشاء اتصال آمن.',
167   167      },
168   168      infoTabTitles: {
169   169          title: 'TON +',
      @@ -278,6 +280,7 @@ logout: {
278   280          expired: 'انتهت صلاحية طلب المصادقة هذا بالفعل',
279   281          failed: 'Invalid URL',
280   282          completed: 'اكتمل طلب المصادقة هذا بالفعل',
283       +        rejected: 'تم رفض طلب التفويض',
281   284          authorized: 'الموافقة على طلب التفويض',
282   285          authRequired: 'المصادقة مطلوبة',
283   286          disableAppAuth: 'عند البدء، تحقق الهوية',
```

app/i18n/i18n_en.ts

```
      @@ -162,6 +162,8 @@ const schema: PrepareSchema<LocalizationSchema, '' | '_plural'> = {
162   162      },
163   163      applications: {
164   164          noAppsTxt: 'No apps connected.',
165       +        infoMessage: 'When connecting dApps, please select TON Connect.',
166       +        cannotConnectMessage: 'The remote website uses an old TON Connect version, therefore a safe connection cannot be established.'
167   167      },
168   168      infoTabTitles: {
169   169          title: 'TON +',
      @@ -278,6 +280,7 @@ const schema: PrepareSchema<LocalizationSchema, '' | '_plural'> = {
278   280          expired: 'This authentication request already expired',
279   281          failed: 'Invalid URL',
280   282          completed: 'This authentication request already completed',
283       +        rejected: 'Authorization request rejected',
281   284          authorized: 'Authorization request approved',
282   285          authRequired: 'Authentication is required',
283   286          disableAppAuth: 'Startup security check',
```

app/i18n/i18n_es.ts

```
      @@ -162,6 +162,8 @@ const schema: PrepareSchema<LocalizationSchema, '' | '_plural'> = {
162   162      },
163   163      applications: {
164   164          noAppsTxt: 'No hay aplicaciones conectadas.',
165       +        infoMessage: 'Al conectar dApps, seleccione TON Connect.',
166       +        cannotConnectMessage: 'El sitio web remoto utiliza una versión antigua de TON Connect, por lo que no se puede establecer una conexión segura.'
167   167      },
168   168      infoTabTitles: {
169   169          title: 'TON +',
      @@ -278,6 +280,7 @@ const schema: PrepareSchema<LocalizationSchema, '' | '_plural'> = {
278   280          expired: 'Esta solicitud de autenticación ya ha caducado',
279   281          failed: 'Invalid URL',
280   282          completed: 'Esta solicitud de autenticación ya se ha completado',
283       +        rejected: 'Solicitud de autorización rechazada',
281   284          authorized: 'Solicitud de autorización aprobada',
282   285          authRequired: 'Se requiere autenticación',
283   286          disableAppAuth: 'En el inicio identidad',
```

app/i18n/i18n_ru.ts

```
      @@ -162,6 +162,8 @@ const schema: PrepareSchema<LocalizationSchema, '_0' | '_1' | '_2'> = {
162   162      },
163   163      applications: {
164   164          noAppsTxt: 'Нет подключенных приложений.',
165       +        infoMessage: 'При подключении dApps выберите TON Connect.',
166       +        cannotConnectMessage: 'Удаленный веб-сайт использует старую версию TON Connect, поэтому безопасное соединение не может быть установлено.'
167   167      },
168   168      infoTabTitles: {
169   169          title: 'TON +',
      @@ -278,6 +280,7 @@ const schema: PrepareSchema<LocalizationSchema, '_0' | '_1' | '_2'> = {
278   280          expired: 'Этот запрос на авторизацию уже истек',
279   281          failed: 'Неправильная ссылка',
280   282          completed: 'Этот запрос на авторизацию уже подтвержден',
283       +        rejected: 'Запрос на авторизацию отклонен',
281   284          authorized: 'Запрос на авторизацию одобрен',
282   285          authRequired: 'Требуется аутентификация',
283   286          disableAppAuth: 'Пароль при входе',
```

app/i18n/i18n_th.ts

```
      @@ -162,6 +162,8 @@ const schema: PrepareSchema<LocalizationSchema, '' | '_plural'> = {
162   162      },
163   163      applications: {
164   164          noAppsTxt: 'ไม่มีแอปพลิเคชันที่เชื่อมต่อ',
165       +        infoMessage: 'เมื่อเชื่อมต่อแอพพลิเค,โปรดเลือก TON Connect.',
166       +        cannotConnectMessage: 'เว็บไซต์ระยะไกลใช้เก่า TON Connect รุ่นดังนั้นการเชื่อมต่อที่ปลอดภัยไม่สามารถสร้าง'
167   167      },
```

```
166 168     infoTabTitles: {
167 169         title: 'TON +',
    @@ -278,6 +280,7 @@ const schema: PrepareSchema<LocalizationSchema, '' | '_plural'> = {
278 280         expired: 'คำขอการรับรองความถูกต้องนี้หมดอายุแล้ว',
279 281         failed: 'Invalid URL',
280 282         completed: 'การรับรองความถูกต้องของคำขอเสร็จสิ้นแล้ว',
    283 +       rejected: 'คำขอลายมือถูกปฏิเสธ',
281 284         authorized: 'คำขออนุญาตของคุณได้รับการอนุมัติ',
282 285         authRequired: 'จำเป็นต้องมีการตรวจสอบสิทธิ์',
283 286         disableAppAuth: 'เริ่มต้นตรวจสอบความปลอดภัย',
```

3 app/i18n/schema.ts

```
    @@ -94,6 +94,8 @@ export type LocalizationSchema = {
94 94     },
95 95     applications: {
96 96         noAppsTxt: string,
    97 +       infoMessage: string,
    98 +       cannotConnectMessage: string
97 99     },
98 100    settings: {
99 101        title: string,
    @@ -217,6 +219,7 @@ export type LocalizationSchema = {
217 219        expired: string,
218 220        failed: string,
219 221        completed: string,
    222 +       rejected: string,
220 223        noApps: string,
221 224        name: string,
222 225        yourWallet: string,
```

139 app/libs/Tonapi/Tonapi.ts

```
    @@ -0,0 +1,139 @@
  1 + import { getServerConfig } from '../../shared/constants';
  2 + import axios from 'axios';
  3 +
  4 +
  5 + export const bearerToken =
    'eyJhbGciOiJFZERTQSIsInR5cCI6IkpXVCJ9.eyJhdWQiOlsiQ29keGVyNDciXSwiZXhwIjoxODMxODc0ODI5LCJpc3MiOiJAdG9uYX8pX2JvdCIsImp0aSI6IkozNjZSUVlRVVhQQTVKSUNQVzRWU0FY
    UyIsInNjb3BlIjoic2VydmVyIiwic3ViIjoidG9uYX8pIn0.IC_ZzwCZD_6bewTHfAkqjsiPJV9GsR8UtxWOAi-eg0EfXpFRaAc3zDpsbtou_US08wpNqHIXzKTAqan9maLbDw';
  6 + export const tonapiIOEndpoint = 'https://tonapi.io';
  7 + const getBulkInfo = async (addresses: string[]) => {
  8 +   const endpoint = getServerConfig('tonapiIOEndpoint');
  9 +
 10 +   const resp = await axios.post(
 11 +     `${endpoint}/v1/account/getBulkInfo`,
 12 +     {
 13 +       addresses: addresses.join(','),
 14 +     },
 15 +     {
 16 +       headers: {
 17 +         Authorization: `Bearer ${getServerConfig('tonApiKey')}`,
 18 +       },
 19 +     },
 20 +   );
 21 + };
 22 +
 23 + const findByPubkey = async (pubkey: string) => {
 24 +   const endpoint = getServerConfig('tonapiIOEndpoint');
 25 +
 26 +   try {
 27 +     const resp = await axios.get(`${endpoint}/v1/wallet/findByPubkey`, {
 28 +       params: {
 29 +         public_key: pubkey,
 30 +       },
 31 +       headers: {
 32 +         Authorization: `Bearer ${getServerConfig('tonApiKey')}`,
 33 +       },
 34 +     });
 35 +
 36 +     if (!resp.data.wallets) {
 37 +       console.log(resp.data);
 38 +     }
 39 +
 40 +     return resp.data?.wallets ?? [];
 41 +   } catch (err) {
 42 +     console.log(err, err.response.data);
 43 +     return [];
 44 +   }
 45 + };
 46 +
 47 + async function getWalletInfo(address: string) {
 48 +   try {
 49 +     const endpoint = 'https://tonapi.io';
 50 +     const response: any = await axios.get(`${endpoint}/v1/account/getInfo`, {
 51 +       headers: {
 52 +         Authorization: `Bearer ${bearerToken}`,
 53 +       },
 54 +       params: {
 55 +         account: address,
 56 +       },
 57 +     });
 58 +     // console.log(endpoint, address, response);
 59 +     return response.data;
 60 +   } catch (e) {
 61 +     alert(e);
 62 +     console.log(e);
 63 +   }
 64 + }
 65 +
 66 +
 67 + async function getSeqno(address: string) {
 68 +   try {
 69 +     const endpoint = 'https://tonapi.io';
 70 +     const response: any = await axios.get(`${endpoint}/v1/wallet/getSeqno`, {
 71 +       headers: {
 72 +         Authorization: `Bearer ${bearerToken}`,
 73 +       },
 74 +       params: {
 75 +         account: address,
 76 +       },
 77 +     });
 78 +     // console.log(endpoint, address, response);
 79 +     return response.data;
 80 +   } catch (e) {
 81 +     alert(e);
 82 +     console.log(e);
 83 +   }
 84 + }
 85 +
```

```
+
+ type Balances = {
+   balance: number;
+   version: string;
+ };
+
+ async function resolveDns(domain: string) {
+   try {
+     const endpoint = getServerConfig('tonapiIOEndpoint');
+     const response: any = await axios.get(`${endpoint}/v1/dns/resolve`, {
+       headers: {
+         Authorization: `Bearer ${getServerConfig('tonApiKey')}`,
+       },
+       params: {
+         name: domain,
+       },
+     });
+     return response.data;
+   } catch (e) {
+     return false;
+   }
+ }
+
+ async function getBalances(pubkey: string) {
+   const wallets = await findByPubkey(pubkey);
+
+   const balances: Balances[] = [];
+   for (let wallet of wallets) {
+     const versions = ['wallet_v3R1', 'wallet_v3R2', 'wallet_v4R1', 'wallet_v4R2'];
+     const detectedVersion = wallet.interfaces.find((version: any) =>
+       versions.includes(version),
+     );
+     if (detectedVersion) {
+       if (wallet.balance > 0) {
+         const version = detectedVersion.replace('wallet_', '');
+         balances.push({
+           balance: wallet.balance,
+           version,
+         });
+       }
+     }
+   }
+
+   return balances;
+ }
+
+ export const Tonapi = {
+   getBulkInfo,
+   findByPubkey,
+   getSeqno,
+   getWalletInfo,
+   getBalances,
+   resolveDns,
+ };
```

app/libs/Tonapi/index.ts

```
@@ -0,0 +1 @@
+ export * from './Tonapi';
```

app/libs/deeplinking/DeepLinkingContext.ts

```
@@ -0,0 +1,4 @@
+ import React from 'react';
+ import { DeepLinkingContextValue } from './deeplinking.types';
+
+ export const DeepLinkingContext = React.createContext<DeepLinkingContextValue | null>(null);
```

app/libs/deeplinking/DeepLinkingProvider.tsx

```
@@ -0,0 +1,124 @@
+ import React from 'react';
+ import {
+   DeepLinkingContextValue,
+   DeeplinkingResolveOptions,
+   DeeplinkingResolver,
+   DeeplinkOrigin,
+ } from './deeplinking.types';
+ import { DeepLinkingContext } from './DeepLinkingContext';
+ import { useDeeplinkingListener } from './hooks/useDeeplinkingListener';
+ import { applyMiddleware, matchPath, Middleware } from './utils';
+
+ interface DeepLinkingProviderProps {
+   useIsReadyToListen?: () => boolean;
+   useResolvers: () => void;
+ }
+
+ export async function delay(ms: number) {
+   return new Promise((resolve) => setTimeout(resolve, ms));
+ }
+
+ export const DeepLinkingProvider: React.FC<DeepLinkingProviderProps> = (props) => {
+   const resolvers = React.useRef<Map<string, DeeplinkingResolver>>(new Map()).current;
+   const middleware = React.useRef<Middleware | undefined>();
+   const prefixes = React.useRef<string[]>([]);
+
+   const add = (pathname: string, resolver: DeeplinkingResolver) => {
+     if (!resolvers.has(pathname)) {
+       resolvers.set(pathname, resolver);
+     }
+   };
+
+   const getResolver = (url: string, options?: DeeplinkingResolveOptions) => {
+     try {
+       const prefix = prefixes.current.find((prefix) => {
+         return url.startsWith(prefix);
+       });
+
+       if (!prefix) {
+         return null;
+       }
+
+       let pathname = url.substring(prefix.length);
+       if (!pathname.startsWith('/')) {
+         pathname = `/${pathname}`;
+       }
+
+       for (let path of resolvers.keys()) {
+         const matchedPath = matchPath(path, pathname);
+         if (matchedPath) {
+           const resolver = resolvers.get(path)!;
+
+           return async () => {
```

```
+            if (options?.delay) {
+              await delay(options?.delay);
+            }
+
+            const middlewares: Middleware[] = [];
+            if (middleware.current) {
+              middlewares.push(middleware.current);
+            }
+
+            applyMiddleware(middlewares, () => {
+              return resolver({
+                origin: options?.origin || DeeplinkOrigin.DEEPLINK,
+                params: matchedPath.params,
+                // @ts-ignore
+                query: matchedPath.query,
+                resolveParams: options?.params ?? {},
+              });
+            });
+          };
+        }
+      }
+    } catch (err) {
+      console.log('[Deeplinking]: error parse', url, err);
+      // debugLog('[Deeplinking]: error parse', url, err);
+    }
+
+    return null;
+  };
+
+  const resolve = (url: string, options?: DeeplinkingResolveOptions) => {
+    try {
+      const resolver = getResolver(url, options);
+      if (resolver) {
+        resolver();
+      }
+    } catch (err) {
+      console.log('[Deeplinking]: error resolve', url, err);
+    }
+  };
+
+  const addMiddleware = (fn: Middleware) => {
+    middleware.current = fn;
+  };
+
+  const setPrefixes = (arr: string[]) => {
+    prefixes.current = arr;
+  };
+
+  const value: DeepLinkingContextValue = {
+    addMiddleware,
+    setPrefixes,
+    getResolver,
+    resolve,
+    add,
+  };
+
+  return (
+    <DeepLinkingContext.Provider value={value}>
+      <DeepLinkingListener
+        useIsReadyToListen={props.useIsReadyToListen}
+        useResolvers={props.useResolvers}
+      />
+      {props.children}
+    </DeepLinkingContext.Provider>
+  );
+};
+
+const DeepLinkingListener = (props: DeepLinkingProviderProps) => {
+  useDeeplinkingListener(props);
+
+  return null;
+};
```

44 ■■■■■ app/libs/deeplinking/deeplinking.types.ts

```
@@ -0,0 +1,44 @@
+import { Middleware } from './utils';
+
+export interface PathPattern<Path extends string = string> {
+  path: Path;
+  caseSensitive?: boolean;
+  end?: boolean;
+}
+
+export type Params<Key extends string = string> = {
+  readonly [key in Key]: string;
+};
+
+export type DeeplinkingResolverOptions = {
+  origin: DeeplinkOrigin;
+  query: Params;
+  params: Params;
+  resolveParams: Record<string, any>;
+};
+
+export type DeeplinkingResolver = (
+  options: DeeplinkingResolverOptions,
+) => Promise<void> | void;
+
+export type DeeplinkingResolveOptions = {
+  delay?: number;
+  origin?: DeeplinkOrigin;
+  params?: Record<string, any>;
+};
+
+export type DeepLinkingContextValue = {
+  getResolver: (
+    path: string,
+    options?: DeeplinkingResolveOptions,
+  ) => (() => Promise<void>) | null;
+  resolve: (path: string, options?: DeeplinkingResolveOptions) => void;
+  add: (path: string, resolver: DeeplinkingResolver) => void;
+  setPrefixes: (prefixes: string[]) => void;
+  addMiddleware: (middleware: Middleware) => void;
+};
+
+export enum DeeplinkOrigin {
+  DEEPLINK = 'DEEPLINK',
+  QR_CODE = 'QR_CODE',
+}
```

12 ■■■■■ app/libs/deeplinking/hooks/useDeeplinking.ts

```
@@ -0,0 +1,12 @@
+import React from 'react';
```

```
+ import { DeepLinkingContext } from '../DeepLinkingContext';
+
+ export const useDeeplinking = () => {
+   const deeplinking = React.useContext(DeepLinkingContext);
+
+   if (!deeplinking) {
+     throw new Error('No DeepLinkingProvider');
+   }
+
+   return deeplinking;
+ }
```

54 app/libs/deeplinking/hooks/useDeeplinkingListener.ts

```
@@ -0,0 +1,54 @@
+ import { Linking } from 'react-native';
+ import { useCallback, useEffect } from 'react';
+ import { useDeeplinking } from '../hooks/useDeeplinking';
+
+ const useIsReadyToListenDefault = () => true;
+ const useResolversDefault = () => {};
+
+ let IsInitialURLProcessed = false;
+
+ export function useDeeplinkingListener({
+   useIsReadyToListen = useIsReadyToListenDefault,
+   useResolvers = useResolversDefault
+ }) {
+   useResolvers();
+
+   const isReadyToListen = useIsReadyToListen();
+   const deeplinking = useDeeplinking();
+   deeplinking.setPrefixes([
+     'ton://',
+     'tonkeeper://',
+     'https://app.tonkeeper.com',
+     'https://tonhub.com',
+   ]);
+
+   const handleUrl = useCallback(({ url }) => {
+     Linking.canOpenURL(url).then((supported) => {
+       if (supported) {
+         deeplinking.resolve(url);
+       }
+     });
+   }, []);
+
+   useEffect(() => {
+     if (!isReadyToListen) {
+       return;
+     }
+
+     const linkingListener = Linking.addEventListener('url', handleUrl);
+     if (!IsInitialURLProcessed) {
+       IsInitialURLProcessed = true;
+       Linking.getInitialURL()
+         .then((url) => {
+           if (url) {
+             deeplinking.resolve(url);
+           }
+         })
+         .catch((err) => console.error('An error occurred', err));
+     }
+
+     return () => {
+       linkingListener.remove();
+     };
+   }, [isReadyToListen]);
+ }
```

3 app/libs/deeplinking/index.ts

```
@@ -0,0 +1,3 @@
+ export { DeepLinkingProvider } from './DeepLinkingProvider';
+ export { useDeeplinking } from './hooks/useDeeplinking';
+ export * from './deeplinking.types';
```

16 app/libs/deeplinking/utils/applyMiddleware.ts

```
@@ -0,0 +1,16 @@
+ export type MaybePromise<T> = T | Promise<T>;
+
+ export type Middleware = (next: () => void) => MaybePromise<any>;
+
+ export function applyMiddleware(middlewares: Middleware[], last: () => any) {
+   // @ts-ignore
+   async function dispatch(i: number) {
+     let fn = middlewares[i]
+     if (i === middlewares.length) {
+       return last();
+     }
+     return fn && fn(dispatch.bind(null, i + 1))
+   }
+
+   return dispatch(0);
+ }
```

33 app/libs/deeplinking/utils/compilPath.ts

```
@@ -0,0 +1,33 @@
+ export function compilePath(
+   path: string,
+   caseSensitive = false,
+   end = true
+ ): [RegExp, string[]] {
+   let paramNames: string[] = [];
+   let regexpSource =
+     "^" +
+     path
+       .replace(/\/*\*?$/, "")
+       .replace(/^\/*/, "/")
+       .replace(/[\\.*+^$?{}|()[\]]/g, "\\$&")
+       .replace(/:(\w+)/g, (_: string, paramName: string) => {
+         paramNames.push(paramName);
+         return "([^\\/]+)";
+       });
+
+   if (path.endsWith("*")) {
+     paramNames.push('path');
+     regexpSource +=
+       path === "*" || path === "/*"
+         ? "(.*)$"
+         : "(?:\\/(.+)|\\/*)$";
```

```
  } else {
    regexpSource += end
      ? "\\/*$"
      : "(?:(?=[.~-]|%[0-9A-F]{2})|\\b|\\/|$)";
  }

  let matcher = new RegExp(regexpSource, caseSensitive ? undefined : "i");

  return [matcher, paramNames];
}
```

3 app/libs/deeplinking/utils/index.ts

```
@@ -0,0 +1,3 @@
export * from './applyMiddleware';
export * from './compilPath';
export * from './matchPath';
```

61 app/libs/deeplinking/utils/matchPath.ts

```
@@ -0,0 +1,61 @@
import queryParser from 'query-string';
import { compilePath } from "./compilPath";
import {Params, PathPattern} from "../deeplinking.types";

export function matchPath<
  ParamKey extends string,
  Path extends string
>(
  pattern: PathPattern<Path> | Path,
  url: string
) {
  const parsedQuery = queryParser.parseUrl(url);
  const query = parsedQuery.query;
  const pathname = parsedQuery.url;

  if (typeof pattern === "string") {
    pattern = { path: pattern, caseSensitive: false, end: true };
  }

  let [matcher, paramNames] = compilePath(
    pattern.path,
    pattern.caseSensitive,
    pattern.end
  );

  let match = pathname.match(matcher);
  if (!match) return null;

  let matchedPathname = match[0];
  let pathnameBase = matchedPathname.replace(/(.)\/+$/, "$1");
  let captureGroups = match.slice(1);
  let params: Params = paramNames.reduce(
    (memo: any, paramName, index) => {
      let query = '';
      if (paramName === "path") {
        const rawQuery = url.split('?')[1];
        if (rawQuery) {
          query =  rawQuery ? `?${rawQuery}` : '';
        }

        const splatValue = captureGroups[index] ?? '';
        pathnameBase = matchedPathname
          .slice(0, matchedPathname.length - splatValue.length)
          .replace(/(.)\/+$/, "$1");
      }

      const paramValue = captureGroups[index];
      memo[paramName] = `${paramValue}${query}`;
      return memo;
    },
    {}
  );

  return {
    query,
    params,
    pathname: matchedPathname,
    pathnameBase,
    pattern,
  };
}
```

42 app/libs/index.ts

```
@@ -0,0 +1,42 @@
import { memo } from 'react';
export const chunk = (input: any[], size: number) => {
  return input.reduce((arr, item, idx) => {
    return idx % size === 0
      ? [...arr, [item]]
      : [...arr.slice(0, -1), [...arr.slice(-1)[0], item]];
  }, []);
};

export function shuffle(a: any[]) {
  for (let i = a.length - 1; i > 0; i--) {
    const j = Math.floor(Math.random() * (i + 1));
    [a[i], a[j]] = [a[j], a[i]];
  }
  return a;
}

export function getRandomInt(min: number, max: number) {
  return Math.floor(Math.random() * (max - min)) + min;
}

export function sleep(timeout: number): Promise<void> {
  return new Promise((resolve) => {
    setTimeout(() => resolve(), timeout);
  });
}

export const getStackTrace = function () {
  try {
    const obj: any = {};
    Error.captureStackTrace(obj, getStackTrace);
    return obj.stack;
  } catch (e) {
    return e.stack;
  }
};

```

```
38 + export function lowerCaseFirstLetter(string: string) {
39 +   return string.charAt(0).toLowerCase() + string.slice(1);
40 + }
41 +
42 + export const Memo: <T>(c: T) => T = memo; // Fix generic.
```

80 app/shared/constants/config.ts

```
... ... @@ -0,0 +1,80 @@
1 + export enum CryptoCurrencies {
2 +   Usdt = 'usdt',
3 +   Usdc = 'usdc',
4 +   Dai = 'dai',
5 +   Wbtc = 'wbtc',
6 + }
7 +
8 + export type CryptoCurrency = typeof CryptoCurrencies[keyof typeof CryptoCurrencies];
9 +
10 + export enum FiatCurrencies {
11 +   Usd = 'usd',
12 +   Eur = 'eur',
13 +   Rub = 'rub',
14 +   Aed = 'aed',
15 +   Gbp = 'gbp',
16 +   Chf = 'chf',
17 +   Cny = 'cny',
18 +   Krw = 'krw',
19 +   Idr = 'idr',
20 +   Inr = 'inr',
21 +   Jpy = 'jpy',
22 + }
23 +
24 + export type FiatCurrency = typeof FiatCurrencies[keyof typeof FiatCurrencies];
25 +
26 + export enum SelectableVersions {
27 +   V4R2 = 'v4R2',
28 +   V4R1 = 'v4R1',
29 +   V3R2 = 'v3R2',
30 +   V3R1 = 'v3R1',
31 + }
32 +
33 + export type SelectableVersion =
34 +   typeof SelectableVersions[keyof typeof SelectableVersions];
35 +
36 + export const TokenConfig: { [index: string]: any } = {
37 +   [CryptoCurrencies.Usdt]: {
38 +     address: '0xdac17f958d2ee523a2206206994597c13d831ec7',
39 +     blockchain: 'ethereum',
40 +     decimals: 18,
41 +   },
42 +   [CryptoCurrencies.Wbtc]: {
43 +     address: '0x2260fac5e5542a773aa44fbcfedf7c193bc2c599',
44 +     blockchain: 'ethereum',
45 +     decimals: 8,
46 +   },
47 +   [CryptoCurrencies.Usdc]: {
48 +     address: '0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48',
49 +     blockchain: 'ethereum',
50 +     decimals: 8,
51 +   },
52 +   [CryptoCurrencies.Dai]: {
53 +     address: '0x6B175474E89094C44Da98b954EedeAC495271d0F',
54 +     blockchain: 'ethereum',
55 +     decimals: 8,
56 +   },
57 + };
58 +
59 + export const TokenConfigTestnet: { [index: string]: any } = {
60 +   [CryptoCurrencies.Usdt]: {
61 +     address: '0x6ee856ae55b6e1a249f04cd3b947141bc146273c',
62 +     blockchain: 'ethereum',
63 +     decimals: 18,
64 +   },
65 +   [CryptoCurrencies.Wbtc]: {
66 +     address: '0x6ee856ae55b6e1a249f04cd3b947141bc146273c', // ToDo: change to correct
67 +     blockchain: 'ethereum',
68 +     decimals: 8,
69 +   },
70 +   [CryptoCurrencies.Usdc]: {
71 +     address: '0x6ee856ae55b6e1a249f04cd3b947141bc146273c', // ToDo: change to correct
72 +     blockchain: 'ethereum',
73 +     decimals: 8,
74 +   },
75 +   [CryptoCurrencies.Dai]: {
76 +     address: '0x6ee856ae55b6e1a249f04cd3b947141bc146273c', // ToDo: change to correct
77 +     blockchain: 'ethereum',
78 +     decimals: 8,
79 +   },
80 + };
```

2 app/shared/constants/index.ts

```
... ... @@ -0,0 +1,2 @@
1 + export * from './config';
2 + export * from './serverConfig';
```

84 app/shared/constants/serverConfig.ts

```
... ... @@ -0,0 +1,84 @@
1 + export interface ServerConfig {
2 +   _version: number;
3 +   tonsafeEndpoint: string;
4 +   tonEndpoint: string;
5 +   tonEndpointAPIKey: string;
6 +   tonApiEndpoint: string;
7 +   neocryptoWebView: string;
8 +   supportLink: string;
9 +   isExchangeEnabled: string;
10 +   exchangePostUrl: string;
11 +   mercuryoSecret: string;
12 +   appsflyerDevKey: string;
13 +   appsflyerAppId: string;
14 +   tonNFTsMarketplaceEndpoint: string;
15 +   tonapiIOEndpoint: string;
16 +   tonapiMainnetHost: string;
17 +   tonapiTestnetHost: string;
18 +   tonApiKey: string;
19 +   cachedMediaEndpoint: string;
20 +   cachedMediaKey: string;
21 +   cachedMediaSalt: string;
22 +   NFTOnExplorerUrl: string;
23 +   flags: Record<string, boolean>;
24 + }
25 +
26 + let config: ServerConfig | null = null;
```

```
+
+ export function setServerConfig(data: any, isTestnet: boolean) {
+   config = {
+     _version: 0,
+     tonsafeEndpoint: data.tonsafeEndpoint || 'https://api.tonsafe.net',
+     tonEndpoint: data.tonEndpoint || 'https://toncenter.com/api/v2/jsonRPC',
+     tonEndpointAPIKey: data.tonEndpointAPIKey,
+     tonApiEndpoint: data.tonApiEndpoint || 'https://toncenter.com/api/v2',
+     neocryptoWebView: data.neocryptoWebView,
+     supportLink: data.supportLink || 'mailto:ops@tonsafe.net',
+     isExchangeEnabled: data.isExchangeEnabled,
+     exchangePostUrl: data.exchangePostUrl,
+     mercuryoSecret: data.mercuryoSecret,
+     appsflyerDevKey: data.appsflyerDevKey,
+     appsflyerAppId: data.appsflyerAppId,
+     tonNFTsMarketplaceEndpoint: data.tonNFTsMarketplaceEndpoint,
+     tonapiIOEndpoint: data.tonapiIOEndpoint || 'https://tonapi.io',
+     tonApiKey: data.tonApiKey,
+     tonapiMainnetHost: data.tonapiMainnetHost || 'https://tonapi.io',
+     tonapiTestnetHost: data.tonapiTestnetHost || 'https://testnet.tonapi.io',
+     cachedMediaEndpoint: data.cachedMediaEndpoint,
+     cachedMediaKey: data.cachedMediaKey,
+     cachedMediaSalt: data.cachedMediaSalt,
+     NFTOnExplorerUrl: data.NFTOnExplorerUrl || 'https://tonscan.org/nft/%s',
+     flags: data.flags || {},
+   };
+ }
+
+ export function updateServerConfig(jsonConfig: any) {
+   if (!jsonConfig) return;
+   try {
+     Object.entries(JSON.parse(jsonConfig)).map(([key, value]) => {
+       if (config) {
+         // @ts-ignore
+         config[key] = value;
+       }
+     });
+   } catch (e) {
+   }
+ }
+
+ export function isServerConfigLoaded() {
+   return !!config;
+ }
+
+ export function getServerConfig<T extends keyof ServerConfig>(key: T): ServerConfig[T] {
+   if (!config) {
+     throw new Error('Config is not loaded');
+   }
+
+   return config[key];
+ }
+
+ export function getServerConfigSafe<T extends keyof ServerConfig>(
+   key: T,
+ ): ServerConfig[T] | 'none' {
+   return config ? config[key] : 'none';
+ }
```

14 ■■■■■ app/shared/dynamicConfig.ts

```
@@ -0,0 +1,14 @@
+ import { CryptoCurrency, TokenConfig } from '../shared/constants';
+
+ export function getTokenConfig(token: CryptoCurrency) {
+   const configs = TokenConfig;
+   return configs[token];
+ }
+
+ export function getWalletName(): string {
+   return `${getChainName()}_default`;
+ }
+
+ export function getChainName(): 'mainnet' | 'testnet' {
+   return 'mainnet';
+ }
```

5 ■■■■■ app/store/index.ts

```
@@ -0,0 +1,5 @@
+ import {createStore} from "zustand";
+ export const store = createStore(() => {});
+
+ // new store
+ export * from './zustand';
```

179 ■■■■■ app/store/models.ts

```
@@ -0,0 +1,179 @@
+ export type TransactionType =
+   | 'receive'
+   | 'sent'
+   | 'buy'
+   | 'subscription'
+   | 'unsubscription';
+
+ export interface Account {
+   address: string;
+   name?: string;
+   icon?: string;
+   isScam?: boolean;
+ }
+
+ export interface Jetton {
+   id: string;
+   name: string;
+   symbol?: string;
+   icon?: string;
+   divisibility?: number;
+ }
+
+ export interface TonTransfer {
+   amount: string; // nanocoins
+   recipient: Account; // address of the recipient
+   sender: Account; // address of the sender
+   payload?: string;
+   comment?: string;
+ }
+
+ export interface TonDeploy {
+   amount: string; // nanocoins
+   recipient: Account; // address of the recipient
+   sender: Account; // address of the sender
+   state_init: string; // recipient's StateInit
+   payload: string;
```

```
+   comment: string; // optional
+ }
+
+ export interface NftItemTransfer {
+   sender: Account; // address of the sender
+   recipient: Account; // address of the recipient
+   payload: string;
+   comment: string; // utf8 string from payload
+ }
+
+ export interface JettonTransfer {
+   jetton: Jetton;
+   amount: string; // base units for this jetton
+   sender: Account; // address of the sender
+   recipient: Account; // address of the recipient
+   payload: string;
+   comment: string; // utf8 string from payload
+ }
+
+ export interface ContractDeploy {
+   address: string;
+   deployer: Account; // address of the deployer
+   interfaces: string[];
+ }
+
+ export enum ActionType {
+   TonTransfer = 'tonTransfer',
+   JettonTransfer = 'jettonTransfer',
+   ContractDeploy = 'contractDeploy',
+   NftItemTransfer = 'nftItemTransfer',
+   Subscribe = 'subscribe',
+   UnSubscribe = 'unSubscribe',
+   Unknown = 'unknown',
+   AuctionBid = 'auctionBid',
+ }
+
+ export interface SubscriptionModel {
+   id?: string;
+   productName: string;
+   channelTgId: string;
+   amountNano: string;
+   intervalSec: number;
+   address: string;
+   status: string;
+   merchantName: string;
+   merchantPhoto: string;
+   returnUrl: string;
+   subscriptionId: number;
+   subscriptionAddress: string;
+   isActive?: boolean;
+   chargedAt: number;
+   fee: string;
+   userReturnUrl: string;
+ }
+
+ export interface InternalNotificationModel {
+   id: string;
+   title: string;
+   caption: string;
+   mode: 'warning';
+   action: {
+     type: 'open_link';
+     label: string;
+     url?: string;
+   };
+   isPersistenceHide?: boolean;
+ }
+
+ export interface ExchangeCategoryModel {
+   title: string;
+   subtitle: string;
+   items: string[];
+ }
+
+ export interface MarketplaceModel {
+   id: string;
+   title: string;
+   description: string;
+   icon_url: string;
+   marketplace_url: string;
+ }
+
+ export interface ExchangeMethodModel {
+   id: string;
+   title: string;
+   subtitle: string;
+   description: string;
+   disabled?: boolean;
+   badge: string;
+   badgeStyle: 'blue' | 'red';
+   features: {
+     text: string;
+     icon: string;
+   }[];
+   icon_url: string;
+   action_button: {
+     title: string;
+     url: string;
+   };
+   info_buttons: {
+     title: string;
+     url: string;
+   }[];
+   successUrlPattern?: {
+     pattern: string;
+     purchaseIdIndex: number;
+   };
+ }
+
+ export interface CollectionModel {
+   ownerAddressToDisplay?: string;
+   name: string;
+   description: string;
+   address: string;
+   addressRaw: string;
+   getGemsModerated?: boolean;
+ }
+
+ export interface TonDiamondMetadata {
+   animation_url: string;
+   image: string;
+   image_diamond: string;
+   lottie: string;
+   theme: {
+     main: string;
+   };
```

```
+ }
+
+ export interface JettonMetadata {
+   jettonAddress: string;
+   decimals: number;
+   symbol?: string;
+   image_data?: string;
+   image?: string;
+   description?: string;
+   name?: string;
+ }
+
+ export interface FavoriteModel {
+   name: string;
+   address: string;
+   domain?: string;
+ }
```

4 app/store/rootReducer.ts

```
@@ -0,0 +1,4 @@
+ // import { combineReducers } from '@reduxjs/toolkit';
+ // export const rootReducer = combineReducers({});
+ //
+ // export type RootState = ReturnType<typeof rootReducer>;
```

1 app/store/rootSaga.ts

```
@@ -0,0 +1 @@
+ export function* rootSaga() {}
```

2 app/store/zustand/appsList/index.ts

```
@@ -0,0 +1,2 @@
+ export * from './useAppsListStore';
+ export * from './types';
```

15 app/store/zustand/appsList/types.ts

```
@@ -0,0 +1,15 @@
+ export interface IAppMetadata {
+   name: string;
+   icon: string;
+   url: string;
+ }
+
+ export interface IAppsListStore {
+   fetching: boolean;
+   appsList: IAppMetadata[];
+   moreEnabled: boolean;
+   moreUrl: string;
+   actions: {
+     fetchPopularApps: () => void;
+   };
+ }
```

52 app/store/zustand/appsList/useAppsListStore.ts

```
@@ -0,0 +1,52 @@
+ import AsyncStorage from "@react-native-async-storage/async-storage";
+ import axios from 'axios';
+ import FastImage from 'react-native-fast-image';
+ import create from 'zustand';
+ import { persist } from 'zustand/middleware';
+ import { IAppsListStore } from './types';
+
+ const initialState: Omit<IAppsListStore, 'actions'> = {
+   fetching: true,
+   appsList: [],
+   moreEnabled: false,
+   moreUrl: '',
+ };
+
+ export const useAppsListStore = create(
+   persist<IAppsListStore>(
+     (set) => ({
+       ...initialState,
+       actions: {
+         fetchPopularApps: async () => {
+           set({ fetching: true });
+           try {
+             const response = await axios.get(
+               `https://api.tonkeeper.com/apps/popular`,
+             );
+
+             const { apps, moreEnabled, moreUrl } = response.data.data;
+
+             FastImage.preload(
+               apps.map((app: any) => ({
+                 uri: app.icon,
+               })),
+             );
+
+             set({ appsList: apps, moreEnabled, moreUrl });
+           } catch {
+           } finally {
+             set({ fetching: false });
+           }
+         },
+       },
+     }),
+     {
+       name: 'appsList',
+       getStorage: () => AsyncStorage,
+       partialize: ({ appsList, moreEnabled }) =>
+         ({ appsList, moreEnabled } as IAppsListStore),
+     },
+   ),
+ );
+
+ useAppsListStore.getState().actions.fetchPopularApps();
```

73 app/store/zustand/connectedApps/helpers.ts

```
@@ -0,0 +1,73 @@
+ import { getChainName } from '../../../shared/dynamicConfig';
+ import { getConnectedAppByUrl } from './selectors';
+ import { IConnectedAppConnection, IConnectedApp, TonConnectBridgeType } from './types';
+ import { useConnectedAppsStore } from './useConnectedAppsStore';
+ import {getCurrentAddress} from "../../../storage/appState";
+ import {AppConfig} from "../../../AppConfig";
+
+ export const saveAppConnection = (
+   walletAddress: string,
```

```
+   appData: IConnectedAppData | 'connection',
+   connection: IConnectedAppConnection,
+ ) => {
+   useConnectedAppsStore
+     .getState()
+     .actions.saveAppConnection(getChainName(), walletAddress, appData, connection);
+ };
+
+ export const removeConnectedApp = (url: string) => {
+   const currentWalletAddress = getCurrentAddress().address.toFriendly({testOnly: AppConfig.isTestnet});
+
+   useConnectedAppsStore
+     .getState()
+     .actions.removeApp(getChainName(), currentWalletAddress, url);
+ };
+
+ export const removeInjectedConnection = (url: string) => {
+   const currentWalletAddress = getCurrentAddress().address.toFriendly({testOnly: AppConfig.isTestnet});
+
+   useConnectedAppsStore
+     .getState()
+     .actions.removeInjectedConnection(getChainName(), currentWalletAddress, url);
+ };
+
+ export const findConnectedAppByUrl = (url: string): IConnectedApp | null => {
+   const currentWalletAddress = getCurrentAddress().address.toFriendly({testOnly: AppConfig.isTestnet});
+
+   return getConnectedAppByUrl(
+     currentWalletAddress,
+     url,
+     useConnectedAppsStore.getState(),
+   );
+ };
+
+ export const findConnectedAppByClientSessionId = (
+   clientSessionId: string,
+ ): { connectedApp: IConnectedApp | null; connection: IConnectedAppConnection | null } => {
+   const currentWalletAddress = getCurrentAddress().address.toFriendly({testOnly: AppConfig.isTestnet});
+
+   const connectedAppsList = Object.values(
+     useConnectedAppsStore.getState().connectedApps[getChainName()][
+       currentWalletAddress
+     ] || {},
+   );
+
+   let connection: IConnectedAppConnection | null = null;
+
+   const connectedApp = connectedAppsList.find((app) =>
+     app.connections.find((item) => {
+       if (
+         item.type === TonConnectBridgeType.Remote &&
+         item.clientSessionId === clientSessionId
+       ) {
+         connection = item;
+
+         return true;
+       }
+
+       return false;
+     }),
+   );
+
+   return { connectedApp: connectedApp ?? null, connection };
+ };
```

5 app/store/zustand/connectedApps/index.ts

```
@@ -0,0 +1,5 @@
+ export * from './types';
+ export * from './useConnectedAppsStore';
+ export * from './helpers';
+ export * from './selectors';
+ export * from './useConnectedAppsList';
```

24 app/store/zustand/connectedApps/selectors.ts

```
@@ -0,0 +1,24 @@
+ import { getChainName } from '../../../shared/dynamicConfig';
+ import {generateAppHashFromUrl} from "../../../utils/misc";
+ import concat from 'lodash/concat';
+ import { IConnectedApp, IConnectedAppsStore } from './types';
+
+ export const getConnectedAppByUrl = (
+   walletAddress: string,
+   url: string,
+   state: IConnectedAppsStore,
+ ): IConnectedApp | null => {
+   const hash = generateAppHashFromUrl(url);
+
+   const connectedApp = state.connectedApps[getChainName()][walletAddress]?.[hash];
+
+   return connectedApp ?? null;
+ };
+
+ export const getAllConnections = (state: IConnectedAppsStore, walletAddress: string) => {
+   const apps = Object.values(state.connectedApps[getChainName()][walletAddress] || {});
+
+   const connections = concat(...apps.map((app) => app.connections));
+
+   return connections;
+ };
```

56 app/store/zustand/connectedApps/types.ts

```
@@ -0,0 +1,56 @@
+ import { ConnectItemReply, KeyPair } from '@tonconnect/protocol';
+
+ export enum TonConnectBridgeType {
+   Remote = 'remote',
+   Injected = 'injected',
+ }
+
+ export interface IConnectedAppConnectionRemote {
+   type: TonConnectBridgeType.Remote;
+   sessionKeyPair: KeyPair;
+   clientSessionId: string;
+   replyItems: ConnectItemReply[];
+ }
+
+ export interface IConnectedAppConnectionInjected {
+   type: TonConnectBridgeType.Injected;
+   replyItems: ConnectItemReply[];
+ }
+
+ export type IConnectedAppConnection = IConnectedAppConnectionRemote | IConnectedAppConnectionInjected;
+
```

```ts
export interface IConnectedApp {
  name: string;
  url: string;
  icon: string;
  autoConnectDisabled?: boolean;
  connections: IConnectedAppConnection[];
}

export interface IConnectedAppsStore {
  connectedApps: {
    [chainName: string]: {
      [walletAddress: string]: {
        [domain: string]: IConnectedApp;
      };
    };
  };
  actions: {
    saveAppConnection: (
      chainName: 'mainnet' | 'testnet',
      walletAddress: string,
      appData: Omit<IConnectedApp, 'connections'>,
      connection: IConnectedAppConnection,
    ) => void;
    removeApp: (
      chainName: 'mainnet' | 'testnet',
      walletAddress: string,
      url: string,
    ) => void;
    removeInjectedConnection: (
      chainName: 'mainnet' | 'testnet',
      walletAddress: string,
      url: string,
    ) => void;
  };
}
```

18 app/store/zustand/connectedApps/useConnectedAppsList.ts

```ts
@@ -0,0 +1,18 @@
import { getChainName } from '../../../shared/dynamicConfig';
import { useCallback } from 'react';
import { IConnectedApp } from './types';
import { useConnectedAppsStore } from './useConnectedAppsStore';
import * as React from "react";
import {getCurrentAddress} from "../../../storage/appState";
import {AppConfig} from "../../../AppConfig";

export const useConnectedAppsList = (): IConnectedApp[] => {
  const address = React.useMemo(() => getCurrentAddress().address, []);

  return useConnectedAppsStore(
    useCallback(
      (s) => Object.values(s.connectedApps[getChainName()][address.toFriendly({testOnly: AppConfig.isTestnet})] ?? {}),
      [address.toFriendly({testOnly: AppConfig.isTestnet})],
    ),
  );
};
```

95 app/store/zustand/connectedApps/useConnectedAppsStore.ts

```ts
@@ -0,0 +1,95 @@
import {generateAppHashFromUrl} from "../../../utils/misc";
import AsyncStorage from '@react-native-async-storage/async-storage';
import create from 'zustand';
import { persist, subscribeWithSelector } from 'zustand/middleware';
import { IConnectedAppsStore, TonConnectBridgeType } from './types';

const initialState: Omit<IConnectedAppsStore, 'actions'> = {
  connectedApps: {
    mainnet: {},
    testnet: {},
  },
};

export const useConnectedAppsStore = create(
  subscribeWithSelector(
    persist<IConnectedAppsStore>(
      (set) => ({
        ...initialState,
        actions: {
          saveAppConnection: (chainName, walletAddress, appData, connection) => {
            set(({ connectedApps }) => {
              if (!connectedApps[chainName][walletAddress]) {
                connectedApps[chainName][walletAddress] = {};
              }

              const hash = generateAppHashFromUrl(appData.url);

              const alreadyConnectedApp = connectedApps[chainName][walletAddress][hash];

              if (alreadyConnectedApp) {
                connectedApps[chainName][walletAddress][hash] = {
                  ...alreadyConnectedApp,
                  ...appData,
                  icon: appData.icon || alreadyConnectedApp.icon,
                  autoConnectDisabled:
                    alreadyConnectedApp.autoConnectDisabled &&
                    connection?.type !== TonConnectBridgeType.Injected,
                  connections: connection
                    ? [...alreadyConnectedApp.connections, connection]
                    : alreadyConnectedApp.connections,
                };
              } else {
                connectedApps[chainName][walletAddress][hash] = {
                  ...appData,
                  connections: connection ? [connection] : [],
                };
              }

              return { connectedApps };
            });
          },
          removeInjectedConnection: (chainName, walletAddress, url) => {
            const hash = generateAppHashFromUrl(url);

            set(({ connectedApps }) => {
              if (connectedApps[chainName][walletAddress]?.[hash]) {
                connectedApps[chainName][walletAddress][hash].autoConnectDisabled = true;

                connectedApps[chainName][walletAddress][hash].connections = connectedApps[
                  chainName
                ][walletAddress][hash].connections.filter(
                  (connection) => connection.type !== TonConnectBridgeType.Injected,
                );

```

```
+              if (
+                connectedApps[chainName][walletAddress][hash].connections.length === 0
+              ) {
+                delete connectedApps[chainName][walletAddress][hash];
+              }
+            }
+
+            return { connectedApps };
+          });
+        },
+        removeApp: (chainName, walletAddress, url) => {
+          const hash = generateAppHashFromUrl(url);
+
+          set(({ connectedApps }) => {
+            if (connectedApps[chainName][walletAddress]?.[hash]) {
+              delete connectedApps[chainName][walletAddress][hash];
+            }
+
+            return { connectedApps };
+          });
+        },
+      },
+    }),
+    {
+      name: 'TCApps',
+      getStorage: () => AsyncStorage,
+      partialize: ({ connectedApps }) => ({ connectedApps } as IConnectedAppsStore),
+    },
+  ),
+ ),
+);
```

3 app/store/zustand/devFeaturesToggle/index.ts

```
@@ -0,0 +1,3 @@
+ export * from './useDevFeaturesToggle';
+ export * from './types';
+ export * from './useDevFeatureEnabled';
```

10 app/store/zustand/devFeaturesToggle/types.ts

```
@@ -0,0 +1,10 @@
+ export enum DevFeature {
+   Example = 'Example',
+ }
+
+ export interface IDevFeaturesToggleStore {
+   devFeatures: { [key in DevFeature]: boolean };
+   actions: {
+     toogleFeature: (feature: DevFeature) => void;
+   };
+ }
```

7 app/store/zustand/devFeaturesToggle/useDevFeatureEnabled.ts

```
@@ -0,0 +1,7 @@
+ import { DevFeature, useDevFeaturesToggle } from '../../../store';
+
+ export const useDevFeatureEnabled = (feature: DevFeature) => {
+   const { devFeatures } = useDevFeaturesToggle();
+
+   return devFeatures[feature];
+ };
```

33 app/store/zustand/devFeaturesToggle/useDevFeaturesToggle.ts

```
@@ -0,0 +1,33 @@
+ import AsyncStorage from '@react-native-async-storage/async-storage';
+ import create from 'zustand';
+ import { persist } from 'zustand/middleware';
+ import { DevFeature, IDevFeaturesToggleStore } from './types';
+
+ const initialState: Omit<IDevFeaturesToggleStore, 'actions'> = {
+   devFeatures: {
+     [DevFeature.Example]: false,
+   },
+ };
+
+ export const useDevFeaturesToggle = create(
+   persist<IDevFeaturesToggleStore>(
+     (set) => ({
+       ...initialState,
+       actions: {
+         toogleFeature: async (name: DevFeature) => {
+           set((state) => {
+             const devFeatures = state.devFeatures;
+             devFeatures[name] = !devFeatures[name];
+
+             return { devFeatures };
+           });
+         },
+       },
+     }),
+     {
+       name: 'devFeaturesToggle',
+       getStorage: () => AsyncStorage,
+       partialize: ({ devFeatures }) => ({ devFeatures } as IDevFeaturesToggleStore),
+     },
+   ),
+ );
```

3 app/store/zustand/index.ts

```
@@ -0,0 +1,3 @@
+ export * from './connectedApps';
+ export * from './appsList';
+ export * from './devFeaturesToggle';
```

17 app/tonconnect/ConnectEventError.ts

```
@@ -0,0 +1,17 @@
+ import {
+   ConnectEventError as IConnectEventError,
+   CONNECT_EVENT_ERROR_CODES,
+ } from '@tonconnect/protocol';
+
+ export class ConnectEventError implements IConnectEventError {
+   event: IConnectEventError['event'];
+   payload: IConnectEventError['payload'];
+
+   constructor(code = CONNECT_EVENT_ERROR_CODES.UNKNOWN_ERROR, message: string) {
+     this.event = 'connect_error';
+     this.payload = {
+       code,
```

```
+      message,
+    };
+  }
+}
```

142 app/tonconnect/ConnectReplyBuilder.ts

```
@@ -0,0 +1,142 @@
+import {
+  CHAIN,
+  ConnectItem,
+  ConnectItemReply,
+  ConnectRequest,
+  TonProofItemReply,
+} from '@tonconnect/protocol';
+import naclUtils from 'tweetnacl-util';
+import nacl from 'tweetnacl';
+import { Buffer } from 'buffer';
+import { getTimeSec } from '../utils/getTimeSec';
+import { Int64LE } from 'int64-buffer';
+import { CONNECT_EVENT_ERROR_CODES } from '@tonconnect/protocol';
+import { DAppManifest } from './models';
+import TonWeb from "tonweb";
+import Url from "url-parse";
+import {walletStateInit} from "../utils/misc";
+const { createHash } = require('react-native-crypto');
+
+export class ConnectReplyBuilder {
+  request: ConnectRequest;
+
+  manifest: DAppManifest;
+
+  constructor(request: ConnectRequest, manifest: DAppManifest) {
+    this.request = request;
+    this.manifest = manifest;
+  }
+
+  private static getNetwork() {
+    return CHAIN.MAINNET;
+  }
+
+  private createTonProofItem(
+    address: string,
+    secretKey: Uint8Array,
+    payload: string,
+  ): TonProofItemReply {
+    try {
+      const timestamp = getTimeSec();
+      const timestampBuffer = new Int64LE(timestamp).toBuffer();
+      const domain = new Url(this.manifest.url, true).host;
+      const domainBuffer = Buffer.from(domain);
+      const domainLengthBuffer = Buffer.allocUnsafe(4);
+      domainLengthBuffer.writeInt32LE(domainBuffer.byteLength);
+
+      const [workchain, addrHash] = address.split(':');
+
+      const addressWorkchainBuffer = Buffer.allocUnsafe(4);
+      addressWorkchainBuffer.writeInt32BE(Number(workchain));
+
+      const addressBuffer = Buffer.concat([
+        addressWorkchainBuffer,
+        Buffer.from(addrHash, 'hex'),
+      ]);
+
+      const messageBuffer = Buffer.concat([
+        Buffer.from('ton-proof-item-v2/'),
+        addressBuffer,
+        domainLengthBuffer,
+        domainBuffer,
+        timestampBuffer,
+        Buffer.from(payload),
+      ]);
+
+      const message = createHash('sha256').update(messageBuffer).digest();
+
+      const bufferToSign = Buffer.concat([
+        Buffer.from('ffff', 'hex'),
+        Buffer.from('ton-connect'),
+        message,
+      ]);
+
+      const signed = nacl.sign.detached(
+        createHash('sha256').update(bufferToSign).digest(),
+        secretKey,
+      );
+
+      const signature = naclUtils.encodeBase64(signed);
+
+      return {
+        name: 'ton_proof',
+        proof: {
+          timestamp,
+          domain: {
+            lengthBytes: domainBuffer.byteLength,
+            value: domain,
+          },
+          signature,
+          payload,
+        },
+      };
+    } catch (e) {
+      return {
+        name: 'ton_proof',
+        error: {
+          // @ts-ignore
+          code: CONNECT_EVENT_ERROR_CODES.UNKNOWN_ERROR,
+          message: `Wallet internal error: ${e.message}`,
+        },
+      };
+    }
+  }
+
+  createReplyItems(addr: string, privateKey: Uint8Array): ConnectItemReply[] {
+    const address = new TonWeb.utils.Address(addr).toString(false, true, true);
+    const replyItems = this.request.items.map((requestItem): ConnectItemReply => {
+      switch (requestItem.name) {
+        case 'ton_addr':
+          return {
+            name: 'ton_addr',
+            address,
+            network: ConnectReplyBuilder.getNetwork(),
+            walletStateInit: walletStateInit,
+          };
+
+        case 'ton_proof':
+          return this.createTonProofItem(address, privateKey, requestItem.payload);
```

```
+
+        default:
+          return {
+            name: (requestItem as ConnectItem).name,
+            error: { code: CONNECT_EVENT_ERROR_CODES.METHOD_NOT_SUPPORTED },
+          } as unknown as ConnectItemReply;
+      }
+    });
+
+    return replyItems;
+  }
+
+  static createAutoConnectReplyItems(addr: string): ConnectItemReply[] {
+    const address = new TonWeb.utils.Address(addr).toString(false, true, true);
+    return [
+      {
+        name: 'ton_addr',
+        address,
+        network: ConnectReplyBuilder.getNetwork(),
+        walletStateInit: walletStateInit,
+      },
+    ];
+  }
+}
```

23 ■■■■■ app/tonconnect/SendTransactionError.ts

```
@@ -0,0 +1,23 @@
+import {
+  SendTransactionRpcResponseError,
+  SEND_TRANSACTION_ERROR_CODES,
+} from '@tonconnect/protocol';
+
+export class SendTransactionError implements SendTransactionRpcResponseError {
+  id: SendTransactionRpcResponseError['id'];
+  error: SendTransactionRpcResponseError['error'];
+
+  constructor(
+    requestId: string,
+    code: SEND_TRANSACTION_ERROR_CODES,
+    message: string,
+    data?: any,
+  ) {
+    this.id = requestId;
+    this.error = {
+      code,
+      message,
+      data,
+    };
+  }
+}
```

345 ■■■■■ app/tonconnect/TonConnect.ts

```
@@ -0,0 +1,345 @@
+import {
+    findConnectedAppByClientSessionId,
+    findConnectedAppByUrl,
+    IConnectedApp,
+    saveAppConnection,
+    removeConnectedApp,
+    TonConnectBridgeType,
+    IConnectedAppConnectionRemote
+} from '../store';
+import {getTimeSec} from '../utils/getTimeSec';
+import {
+    AppRequest,
+    ConnectEvent,
+    ConnectRequest,
+    CONNECT_EVENT_ERROR_CODES,
+    RpcMethod,
+    SEND_TRANSACTION_ERROR_CODES,
+    SessionCrypto,
+    WalletResponse,
+} from '@tonconnect/protocol';
+import axios from 'axios';
+import FastImage from 'react-native-fast-image';
+import {MIN_PROTOCOL_VERSION, tonConnectDeviceInfo} from './config';
+import {ConnectEventError} from './ConnectEventError';
+import {ConnectReplyBuilder} from './ConnectReplyBuilder';
+import {DAppManifest, SignRawParams, TransactionModalParams} from './models';
+import {SendTransactionError} from './SendTransactionError';
+import {TonConnectRemoteBridge} from './TonConnectRemoteBridge';
+import {getCurrentAddress} from "../storage/appState";
+import {AppConfig} from "../AppConfig";
+import {getAppsConnectModalTriggerObject, getTransactionModalTriggerObject} from "./modalTrigger";
+
+class TonConnectService {
+
+    checkProtocolVersionCapability(protocolVersion: number) {
+        if (typeof protocolVersion !== 'number' || protocolVersion < MIN_PROTOCOL_VERSION) {
+            throw new ConnectEventError(
+                CONNECT_EVENT_ERROR_CODES.BAD_REQUEST_ERROR,
+                `Protocol version ${String(protocolVersion)} is not supported by the wallet app`,
+            );
+        }
+    }
+
+    verifyConnectRequest(request: ConnectRequest) {
+        if (!(request && request.manifestUrl && request.items?.length)) {
+            throw new ConnectEventError(
+                CONNECT_EVENT_ERROR_CODES.BAD_REQUEST_ERROR,
+                'Wrong request data',
+            );
+        }
+    }
+
+    async getManifest(request: ConnectRequest) {
+        try {
+            const {data: manifest} = await axios.get<DAppManifest>(request.manifestUrl);
+
+            const isValid =
+                manifest &&
+                typeof manifest.url === 'string' &&
+                typeof manifest.name === 'string' &&
+                typeof manifest.iconUrl === 'string';
+
+            if (!isValid) {
+                throw new ConnectEventError(
+                    CONNECT_EVENT_ERROR_CODES.MANIFEST_CONTENT_ERROR,
+                    'Manifest is not valid',
+                );
+            }
+
+            if (manifest.iconUrl) {
+                FastImage.preload([{uri: manifest.iconUrl}]);
```

```
+            }
+
+            return manifest;
+        } catch (error) {
+            if (axios.isAxiosError(error)) {
+                throw new ConnectEventError(
+                    CONNECT_EVENT_ERROR_CODES.MANIFEST_NOT_FOUND_ERROR,
+                    `Can't get ${request.manifestUrl}`,
+                );
+            }
+
+            throw error;
+        } finally {
+        }
+    }
+
+    async connect(
+        protocolVersion: number,
+        request: ConnectRequest,
+        sessionCrypto?: SessionCrypto,
+        clientSessionId?: string,
+        webViewUrl?: string,
+    ): Promise<ConnectEvent> {
+        try {
+            this.checkProtocolVersionCapability(protocolVersion);
+
+            this.verifyConnectRequest(request);
+
+            const manifest = await this.getManifest(request);
+
+            try {
+                const {address, replyItems} = await new Promise<any>(
+                    (resolve, reject) => {
+                        const connectedAppsModal = getAppsConnectModalTriggerObject();
+                        connectedAppsModal.subject.next({
+                            protocolVersion: protocolVersion as 2,
+                            manifest,
+                            replyBuilder: new ConnectReplyBuilder(request, manifest),
+                            requestPromise: {resolve, reject},
+                            hideImmediately: !!webViewUrl
+                        });
+                    }
+                );
+
+                saveAppConnection(
+                    address,
+                    {
+                        name: manifest.name,
+                        url: manifest.url,
+                        icon: manifest.iconUrl,
+                    },
+                    webViewUrl
+                        ? {type: TonConnectBridgeType.Injected, replyItems}
+                        : {
+                            type: TonConnectBridgeType.Remote,
+                            sessionKeyPair: sessionCrypto!.stringifyKeypair(),
+                            clientSessionId: clientSessionId!,
+                            replyItems,
+                            },
+                );
+
+                return {
+                    event: 'connect',
+                    payload: {
+                        items: replyItems,
+                        device: tonConnectDeviceInfo,
+                    },
+                };
+            } catch {
+                throw new ConnectEventError(
+                    CONNECT_EVENT_ERROR_CODES.USER_REJECTS_ERROR,
+                    'Wallet declined the request',
+                );
+            }
+        } catch (error) {
+            if (error instanceof ConnectEventError) {
+                return error;
+            }
+
+            return new ConnectEventError(
+                CONNECT_EVENT_ERROR_CODES.UNKNOWN_ERROR,
+                error?.message,
+            );
+        }
+    }
+
+    /**
+     * Only for injected ton-connect bridge
+     */
+    async autoConnect(webViewUrl: string): Promise<ConnectEvent> {
+        try {
+            const connectedApp = findConnectedAppByUrl(webViewUrl);
+
+            if (
+                !connectedApp ||
+                connectedApp.connections.length === 0 ||
+                connectedApp.autoConnectDisabled
+            ) {
+                throw new ConnectEventError(
+                    CONNECT_EVENT_ERROR_CODES.UNKNOWN_APP_ERROR,
+                    'Unknown app',
+                );
+            }
+
+            const currentWalletAddress = getCurrentAddress().address.toFriendly({testOnly: AppConfig.isTestnet});
+
+            const replyItems =
+                ConnectReplyBuilder.createAutoConnectReplyItems(currentWalletAddress);
+
+            return {
+                event: 'connect',
+                payload: {
+                    items: replyItems,
+                    device: tonConnectDeviceInfo,
+                },
+            };
+        } catch (error) {
+            if (error instanceof ConnectEventError) {
+                return error;
+            }
+
+            return new ConnectEventError(
+                CONNECT_EVENT_ERROR_CODES.UNKNOWN_ERROR,
+                error?.message,
+            );
+        }
```

```
+    }
+
+    async sendTransaction(
+        request: AppRequest<'sendTransaction'>,
+    ): Promise<WalletResponse<'sendTransaction'>> {
+        try {
+            const params = JSON.parse(request.params[0]) as any;
+
+            const isValidRequest =
+                params &&
+                typeof params.valid_until === 'number' &&
+                Array.isArray(params.messages) &&
+                params.messages.every((msg: any) => !!msg.address && !!msg.amount);
+
+            if (!isValidRequest) {
+                throw new SendTransactionError(
+                    request.id,
+                    SEND_TRANSACTION_ERROR_CODES.BAD_REQUEST_ERROR,
+                    'Bad request',
+                );
+            }
+
+            const {valid_until, messages} = params;
+
+            if (valid_until < getTimeSec()) {
+                throw new SendTransactionError(
+                    request.id,
+                    SEND_TRANSACTION_ERROR_CODES.BAD_REQUEST_ERROR,
+                    'Request timed out',
+                );
+            }
+
+            const currentWalletAddress = getCurrentAddress().address.toFriendly({testOnly: AppConfig.isTestnet});
+
+            const txParams: SignRawParams = {
+                valid_until,
+                messages,
+                source: currentWalletAddress,
+            };
+
+            const boc = await new Promise<string>(async (resolve, reject) => {
+                const transactionModalObject = getTransactionModalTriggerObject();
+                const dataForTransactionModal: TransactionModalParams = {
+                    params: txParams,
+                    onSuccess: resolve,
+                    onDismiss: () =>
+                        reject(
+                            new SendTransactionError(
+                                request.id,
+                                SEND_TRANSACTION_ERROR_CODES.USER_REJECTS_ERROR,
+                                'Wallet declined the request',
+                            ),
+                        ),
+                    options: {
+                        expires_sec: valid_until,
+                        response_options: {
+                            broadcast: false,
+                        }
+                    }
+                }
+                transactionModalObject.subject.next({
+                    params: dataForTransactionModal.params,
+                    options: dataForTransactionModal.options,
+                    tResolve: dataForTransactionModal.onSuccess,
+                    tReject: dataForTransactionModal.onDismiss
+                });
+            });
+
+            return {
+                result: boc,
+                id: request.id,
+            };
+        } catch (error) {
+            if (error instanceof SendTransactionError) {
+                return error;
+            }
+
+            return new SendTransactionError(
+                request.id,
+                SEND_TRANSACTION_ERROR_CODES.UNKNOWN_ERROR,
+                error?.message,
+            );
+        }
+    }
+
+    private async handleRequest<T extends RpcMethod>(
+        request: AppRequest<T>,
+        connectedApp: IConnectedApp | null,
+    ): Promise<WalletResponse<T>> {
+        if (!connectedApp) {
+            return {
+                error: {
+                    code: SEND_TRANSACTION_ERROR_CODES.UNKNOWN_APP_ERROR,
+                    message: 'Unknown app',
+                },
+                id: request.id,
+            };
+        }
+        if (request.method === 'sendTransaction') {
+            return this.sendTransaction(request);
+        }
+
+        return {
+            error: {
+                code: SEND_TRANSACTION_ERROR_CODES.BAD_REQUEST_ERROR,
+                message: `Method "${request.method}" does not supported by the wallet app`,
+            },
+            id: request.id,
+        };
+    }
+
+    async handleRequestFromInjectedBridge<T extends RpcMethod>(
+        request: AppRequest<T>,
+        webViewUrl: string,
+    ): Promise<WalletResponse<T>> {
+        const connectedApp = findConnectedAppByUrl(webViewUrl);
+        console.log('HERE ... 1');
+        return this.handleRequest(request, connectedApp);
+    }
+
+    async handleRequestFromRemoteBridge<T extends RpcMethod>(
+        request: AppRequest<T>,
+        clientSessionId: string,
+    ): Promise<WalletResponse<T>> {
+        const {connectedApp} = findConnectedAppByClientSessionId(clientSessionId);
+        return this.handleRequest(request, connectedApp);
```

```
+    }
+
+    async disconnect(url: string) {
+        const connectedApp = findConnectedAppByUrl(url);
+
+        if (!connectedApp) {
+            return;
+        }
+
+        const remoteConnections = connectedApp.connections.filter(
+            (connection) => connection.type === TonConnectBridgeType.Remote,
+        ) as IConnectedAppConnectionRemote[];
+
+        remoteConnections.forEach((connection) =>
+            TonConnectRemoteBridge.sendDisconnectEvent(connection),
+        );
+
+        removeConnectedApp(url);
+    }
+ }
+
+ export const TonConnect = new TonConnectService();
```

239 app/tonconnect/TonConnectRemoteBridge.ts

```
@@ -0,0 +1,239 @@
+ import {
+   IConnectedAppConnection,
+   IConnectedAppConnectionRemote,
+   TonConnectBridgeType,
+ } from '../store';
+ import {
+   AppRequest,
+   Base64,
+   ConnectEvent,
+   ConnectRequest,
+   DisconnectEvent,
+   hexToByteArray,
+   RpcMethod,
+   SEND_TRANSACTION_ERROR_CODES,
+   SessionCrypto,
+   WalletResponse,
+ } from '@tonconnect/protocol';
+ import debounce from "lodash/debounce";
+ import { TonConnect } from './TonConnect';
+ import { IConnectQrQuery } from './models';
+ import EventSource, { EventSourceListener, MessageEvent } from 'react-native-sse';
+ import {storage as AsyncStorage} from "../storage/storage";
+ import { DeeplinkOrigin } from '../libs/deeplinking';
+ import Minimizer from 'react-native-minimizer';
+ import {getCurrentAddress} from "../storage/appState";
+ import {Tonapi} from "../libs/Tonapi";
+ import {t} from "../i18n/t";
+
+ class TonConnectRemoteBridgeService {
+   private readonly storeKey = 'ton-connect-http-bridge-lastEventId';
+
+   private readonly bridgeUrl = 'https://bridge.tonapi.io/bridge';
+
+   private readonly defaultTtl = 300;
+
+   private eventSource: EventSource | null = null;
+
+   private connections: IConnectedAppConnectionRemote[] = [];
+
+   private activeRequests: { [from: string]: AppRequest<RpcMethod> } = {};
+
+   private origin: DeeplinkOrigin | null = null;
+
+   public setOrigin(origin: DeeplinkOrigin) {
+     this.origin = origin;
+   }
+
+   async open(connections: IConnectedAppConnection[]) {
+     this.close();
+
+     this.connections = connections.filter(
+       (item) => item.type === TonConnectBridgeType.Remote,
+     ) as IConnectedAppConnectionRemote[];
+
+     if (this.connections.length === 0) {
+       return;
+     }
+
+     const walletSessionIds = this.connections
+       .map((item) => new SessionCrypto(item.sessionKeyPair).sessionId)
+       .join(',');
+
+     let url = `${this.bridgeUrl}/events?client_id=${walletSessionIds}`;
+
+     const lastEventId = await this.getLastEventId();
+
+     if (lastEventId) {
+       url += `&last_event_id=${lastEventId}`;
+     }
+
+     console.log('sse connect', url);
+
+     this.eventSource = new EventSource(url);
+
+     this.eventSource.addEventListener(
+       'message',
+       debounce(this.handleMessage.bind(this), 200) as EventSourceListener,
+     );
+
+     this.eventSource.addEventListener('open', () => {
+       console.log('sse connect: opened');
+     });
+
+     this.eventSource.addEventListener('error', (event) => {
+       console.log('sse connect: error', event);
+     });
+   }
+
+   close() {
+     if (this.eventSource) {
+       this.eventSource.removeAllEventListeners();
+       this.eventSource.close();
+       this.eventSource = null;
+       console.log('sse close');
+     }
+   }
+
+   private async setLastEventId(lastEventId: string) {
+     try {
```

```
+      await AsyncStorage.set(this.storeKey, lastEventId);
+    } catch {}
+  }
+
+  private async getLastEventId() {
+    try {
+      return await AsyncStorage.getString(this.storeKey);
+    } catch {
+      return null;
+    }
+  }
+
+  private async send<T extends RpcMethod>(
+    response: WalletResponse<T> | ConnectEvent | DisconnectEvent,
+    sessionCrypto: SessionCrypto,
+    clientSessionId: string,
+    ttl?: number,
+  ): Promise<void> {
+    try {
+      const url = `${this.bridgeUrl}/message?client_id=${
+        sessionCrypto.sessionId
+      }&to=${clientSessionId}&ttl=${ttl || this.defaultTtl}`;
+
+      const encodedResponse = sessionCrypto.encrypt(
+        JSON.stringify(response),
+        hexToByteArray(clientSessionId),
+      );
+
+      await fetch(url, {
+        body: Base64.encode(encodedResponse),
+        method: 'POST',
+      });
+    } catch (e) {
+      console.log('send fail', e);
+    }
+  }
+
+  private async handleMessage(event: MessageEvent) {
+    try {
+      if (event.lastEventId) {
+        this.setLastEventId(event.lastEventId);
+      }
+
+      const { from, message } = JSON.parse(event.data!);
+
+      console.log('handleMessage', from);
+
+      const connection = this.connections.find((item) => item.clientSessionId === from);
+
+      if (!connection) {
+        console.log(`connection with clientId "${from}" not found!`);
+        return;
+      }
+
+      const sessionCrypto = new SessionCrypto(connection.sessionKeyPair);
+      const request: AppRequest<RpcMethod> = JSON.parse(
+        sessionCrypto.decrypt(
+          Base64.decode(message).toUint8Array(),
+          hexToByteArray(from),
+        ),
+      );
+
+      if (this.activeRequests[from]) {
+        await this.send(
+          {
+            error: {
+              code: SEND_TRANSACTION_ERROR_CODES.USER_REJECTS_ERROR,
+              message: 'User has already opened the previous request',
+            },
+            id: request.id,
+          },
+          sessionCrypto,
+          from,
+        );
+
+        return;
+      }
+
+      this.activeRequests[from] = request;
+
+      console.log('handleMessage request', request);
+
+      const response = await TonConnect.handleRequestFromRemoteBridge(request, from);
+
+      delete this.activeRequests[from];
+
+      console.log('handleMessage response', response);
+
+      await this.send(response, sessionCrypto, from);
+
+      this.redirectIfNeeded();
+    } catch (e) {
+      console.log('handleMessage error');
+      console.error(e);
+    }
+  }
+
+  private redirectIfNeeded() {
+    if (this.origin === DeeplinkOrigin.DEEPLINK) {
+      Minimizer.goBack();
+    }
+
+    this.origin = null;
+  }
+
+  async handleConnectDeeplink(query: IConnectQrQuery) {
+    // if ((typeof query.v === 'undefined')) {
+    //   alert(t('applications.cannotConnectMessage'));
+    //   return;
+    // }
+    try {
+      const protocolVersion = Number(query.v);
+      const request = JSON.parse(decodeURIComponent(query.r)) as ConnectRequest;
+      const clientSessionId = query.id;
+      console.log('handleConnectDeeplink request', request);
+      const sessionCrypto = new SessionCrypto();
+      const response = await TonConnect.connect(
+        protocolVersion,
+        request,
+        sessionCrypto,
+        clientSessionId,
+      );
+      console.log('handleConnectDeeplink response', response.event);
+      await this.send(response, sessionCrypto, clientSessionId);
+      // this.redirectIfNeeded();
+    } catch (err) {
```

```
+      console.log('handleConnectDeeplink error', err);
+    }
+  }
+
+  sendDisconnectEvent(connection: IConnectedAppConnectionRemote) {
+    const sessionCrypto = new SessionCrypto(connection.sessionKeyPair);
+
+    const event: DisconnectEvent = { event: 'disconnect', payload: {} };
+
+    this.send(event, sessionCrypto, connection.clientSessionId);
+  }
+}
+
+export const TonConnectRemoteBridge = new TonConnectRemoteBridgeService();
```

23 ■■■■■ app/tonconnect/config.ts

```
@@ -0,0 +1,23 @@
+ import { DeviceInfo } from '@tonconnect/protocol';
+ import RNDeviceInfo from 'react-native-device-info';
+ import { Platform } from 'react-native';
+
+ export const CURRENT_PROTOCOL_VERSION = 2;
+
+ export const MIN_PROTOCOL_VERSION = 2;
+
+ const getPlatform = (): DeviceInfo['platform'] => {
+   if (Platform.OS === 'ios') {
+     return RNDeviceInfo.isTablet() ? 'ipad' : 'iphone';
+   }
+
+   return Platform.OS as DeviceInfo['platform'];
+ };
+
+ export const tonConnectDeviceInfo: DeviceInfo = {
+   platform: getPlatform(),
+   appName: RNDeviceInfo.getApplicationName(),
+   appVersion: RNDeviceInfo.getReadableVersion(),
+   maxProtocolVersion: CURRENT_PROTOCOL_VERSION,
+   features: ['SendTransaction'],
+ };
```

275 ■■■■■ app/tonconnect/connection-view/ExternalAppConnection.tsx

```
@@ -0,0 +1,275 @@
+ import React, {useState} from "react";
+ import {Text, View} from "react-native";
+ import Modal from "react-native-modal";
+ import {Theme} from "../../Theme";
+ import LottieView from "lottie-react-native";
+ import {t, tStyled} from "../../i18n/t";
+ import {RoundButton} from "../../components/RoundButton";
+ import {WImage} from "../../components/WImage";
+ import {getAppsConnectModalTriggerObject} from "../modalTrigger";
+ import {getDarkModeValue} from "../../utils/misc";
+ import ProtectedIcon from "../../../assets/ic_protected.svg";
+ import {DAppManifest, TonConnectResponseModel} from "../models";
+ import {ConnectReplyBuilder} from "../ConnectReplyBuilder";
+ import {contractFromPublicKey} from "../../engine/contractFromPublicKey";
+ import {getCurrentAddress} from "../../storage/appState";
+ import {loadWalletKeys, WalletKeys} from "../../storage/walletKeys";
+ import {AppConfig} from "../../AppConfig";
+ import {warn} from '../../utils/log';
+
+ type stateType = {
+     appIconUrl: string, appTitle: string, protocolVersion: number, manifest: DAppManifest, replyBuilder: ConnectReplyBuilder, requestPromise: {
+         resolve: (response: TonConnectResponseModel) => void;
+         reject: () => void;
+     }, hideImmediately: boolean
+ } | null;
+
+ function ExternalAppConnection() {
+     const connectedAppsModalObject = getAppsConnectModalTriggerObject();
+     const [state, setState] = useState<stateType>(null);
+     const [currentAction, setCurrentAction] = useState({action: 'requesting', message: ''}); // requesting - approved - rejected - expired
+     React.useEffect(() => {
+         const connectedAppsModalSubscriber = connectedAppsModalObject.observable.subscribe((data: any) => {
+             // if (data.protocolVersion !== 1) {
+             //     data.requestPromise.reject();
+             // }
+             const collectedData: stateType = {
+                 hideImmediately: false,
+                 appIconUrl: data.manifest.iconUrl,
+                 appTitle: data.manifest.name,
+                 protocolVersion: data.protocolVersion,
+                 manifest: data.manifest,
+                 replyBuilder: data.replyBuilder,
+                 requestPromise: data.requestPromise
+             }
+             setState(collectedData);
+         });
+         return function cleanup() {
+             connectedAppsModalSubscriber.unsubscribe();
+         }
+     }, [connectedAppsModalObject]);
+
+     const acc = React.useMemo(() => getCurrentAddress(), []);
+     let active = React.useRef(true);
+     React.useEffect(() => {
+         return () => {
+             active.current = false;
+         };
+     }, []);
+
+     const createResponse = React.useCallback(async () => {
+         try {
+             if (state) {
+                 const contract = await contractFromPublicKey(acc.publicKey);
+                 let address = contract.address.toFriendly({testOnly: AppConfig.isTestnet});
+                 let walletKeys: WalletKeys;
+                 try {
+                     walletKeys = await loadWalletKeys(acc.secretKeyEnc);
+                 } catch (e) {
+                     warn(e);
+                     return;
+                 }
+
+                 const protocolVersion = state?.protocolVersion;
+                 if (protocolVersion !== 1) {
+                     const { replyBuilder, requestPromise } = state;
+                     const replyItems = replyBuilder.createReplyItems(address, walletKeys.keyPair.secretKey);
+                     requestPromise.resolve({ address, replyItems });
+                 }
+                 setCurrentAction({action: 'approved', message: t('auth.authorized')});
+             }
+         } catch (error) {
```

```
+        } catch (error) {
+            let message = error?.message;
+            setCurrentAction({action: 'rejected', message: message});
+        }
+    }, [state]);
+
+    const cancelRequest = React.useCallback(async () => {
+        if (state && state.requestPromise) {
+            state.requestPromise.reject();
+            setCurrentAction({action: 'rejected', message: t('auth.rejected')});
+        }
+    }, [state]);
+
+    const closeModal = React.useCallback(async () => {
+        setState(null);
+        setCurrentAction({action: 'requesting', message: ''});
+    }, []);
+
+    return (
+        <Modal isVisible={!!state}>
+            <View style={{width: '100%', backgroundColor: Theme.item, paddingHorizontal: 5, paddingVertical: 10}}>
+                <Text style={{
+                    fontSize: 19,
+                    fontWeight: '600',
+                    color: Theme.textColor,
+                    textAlign: 'center',
+                    textAlignVertical: 'center'
+                }}>{t('auth.title')}</Text>
+            </View>
+            <View style={{backgroundColor: Theme.item}}>
+                <View style={{
+                    flexDirection: 'row',
+                    width: '100%',
+                    height: 120,
+                    backgroundColor: Theme.background,
+                    alignContent: 'center',
+                    alignItems: 'center'
+                }}>
+                    <View style={{width: '30%', alignContent: 'center', alignItems: 'center'}}>
+                        <WImage
+                            heigh={55}
+                            width={55}
+                            style={{marginBottom: 0}}
+                            src={require('../../../assets/user.png')}
+                            blurhash={require('../../../assets/user.png')}
+                            borderRadius={16}
+                        />
+                        <Text style={{fontSize: 11, color: Theme.textColor}}>TonSafe</Text>
+                    </View>
+                    <View style={{width: '40%'}}>
+                        <LottieView
+                            source={require('../../../assets/animations/link-app.json')}
+                            style={{width: '100%', transform: [{rotate: '22deg'}]}}
+                            autoPlay={true}
+                        />
+                    </View>
+                    <View style={{width: '30%', alignContent: 'center', alignItems: 'center'}}>
+                        <WImage
+                            heigh={55}
+                            width={55}
+                            style={{marginBottom: 0}}
+                            src={state?.appIconUrl}
+                            blurhash={state?.appIconUrl}
+                            borderRadius={16}
+                        />
+                        <Text style={{fontSize: 11, color: Theme.textColor}}>{state?.appTitle}</Text>
+                    </View>
+                </View>
+                {currentAction.action === 'requesting' && (
+                    <View style={{
+                        flexDirection: 'column',
+                        marginTop: 5,
+                        justifyContent: 'space-evenly',
+                        alignContent: 'stretch'
+                    }}>
+                        <View style={{
+                            backgroundColor: Theme.lightGreen,
+                            paddingLeft: 15,
+                            paddingRight: 15,
+                            paddingTop: 10,
+                            paddingBottom: 10,
+                            marginBottom: 5
+                        }}>
+                            <Text
+                                style={{
+                                    fontSize: 16,
+                                    textAlign: 'center',
+                                    color: Theme.textColor,
+                                    fontWeight: '400'
+                                }}
+                            >
+                                {tStyled('auth.message', {name: state?.appTitle})}
+                            </Text>
+                            <View style={{flexDirection: 'row', marginTop: 15, marginBottom: 5}}>
+                                <ProtectedIcon height={26} width={26} style={{marginRight: 5}}/>
+                                <Text
+                                    style={{
+                                        fontSize: 13,
+                                        fontWeight: '400',
+                                        color: Theme.textColor,
+                                        opacity: 0.6
+                                    }}
+                                >{
+                                    t('auth.hint')}
+                                </Text>
+                            </View>
+                        </View>
+                        <View style={{paddingHorizontal: 16, paddingBottom: 16, marginTop: 10}}>
+                            <RoundButton
+                                title={t('auth.action')}
+                                display={getDarkModeValue() ? 'default' : 'outline'}
+                                action={createResponse}
+                            />
+                            <RoundButton
+                                title={t('common.cancel')}
+                                display={getDarkModeValue() ? 'outline' : 'default'}
+                                style={{marginTop: 10}}
+                                action={cancelRequest}
+                            />
+                        </View>
+                    </View>
+                )}
+
+                {currentAction.action === 'approved' && (
+                    <View style={{
+                        flexGrow: 1,
+                        width: '100%',
```

```
+                        alignContent: 'center',
+                        alignItems: 'center',
+                        paddingBottom: 20
+                    }}>
+                        <LottieView
+                            source={require('../../../assets/animations/green-check.json')}
+                            style={{width: 140}}
+                            autoPlay={true}
+                            loop={true}
+                        />
+                        <Text style={{
+                            marginTop: 5,
+                            fontSize: 16,
+                            fontWeight: '600',
+                            color: Theme.primeGreen,
+                            textAlign: 'center'
+                        }}>{currentAction.message}</Text>
+                        <Text style={{
+                            marginTop: 5,
+                            fontSize: 12,
+                            fontWeight: '400',
+                            color: Theme.textColor,
+                            textAlign: 'left'
+                        }}>{t('auth.authorizedDescription')}</Text>
+                        <RoundButton
+                            title={t('common.close')}
+                            display={getDarkModeValue() ? 'outline' : 'default'}
+                            style={{marginTop: 25, width: '60%'}}
+                            action={closeModal}
+                        />
+                    </View>
+                )}
+
+                {currentAction.action === 'rejected' && (
+                    <View style={{
+                        flexGrow: 1,
+                        width: '100%',
+                        alignContent: 'center',
+                        alignItems: 'center',
+                        paddingBottom: 20
+                    }}>
+                        <LottieView
+                            source={require('../../../assets/animations/red-cross.json')}
+                            style={{width: 140}}
+                            autoPlay={true}
+                            loop={true}
+                        />
+                        <Text style={{
+                            marginTop: 5,
+                            fontSize: 16,
+                            fontWeight: '600',
+                            color: Theme.primeRed
+                        }}>{currentAction.message}</Text>
+                        <RoundButton
+                            title={t('common.close')}
+                            display={getDarkModeValue() ? 'outline' : 'default'}
+                            style={{marginTop: 25, width: '60%'}}
+                            action={closeModal}
+                        />
+                    </View>
+                )}
+
+            </View>
+        </Modal>
+    );
+ }
+
+ export default ExternalAppConnection;
```

1 app/tonconnect/hooks/index.ts

```
@@ -0,0 +1 @@
+ export * from './useRemoteBridge';
```

26 app/tonconnect/hooks/useRemoteBridge.ts

```
@@ -0,0 +1,26 @@
+ import { getAllConnections, useConnectedAppsStore } from '../../store';
+ import { useEffect } from 'react';
+ import { TonConnectRemoteBridge } from '../TonConnectRemoteBridge';
+ import {getCurrentAddress} from "../../storage/appState";
+ import {AppConfig} from "../../AppConfig";
+
+ export const useRemoteBridge = () => {
+   const { address } = getCurrentAddress();
+   useEffect(() => {
+     const initialConnections = getAllConnections(
+       useConnectedAppsStore.getState(),
+       address.toFriendly({testOnly: AppConfig.isTestnet}),
+     );
+     TonConnectRemoteBridge.open(initialConnections);
+     const unsubscribe = useConnectedAppsStore.subscribe(
+       (s: any) => getAllConnections(s, address.toFriendly({testOnly: AppConfig.isTestnet})),
+       (connections: any) => {
+         TonConnectRemoteBridge.open(connections);
+       },
+     );
+     return () => {
+       unsubscribe();
+       TonConnectRemoteBridge.close();
+     };
+   }, [address.toFriendly({testOnly: AppConfig.isTestnet})]);
+ };
```

6 app/tonconnect/index.ts

```
@@ -0,0 +1,6 @@
+ export * from './config';
+ export * from './TonConnect';
+ export * from './ConnectReplyBuilder';
+ export * from './TonConnectRemoteBridge';
+ export * from './hooks';
+ export * from './models';
```

12 app/tonconnect/modalTrigger.ts

```
@@ -0,0 +1,12 @@
+ import {Subject} from 'rxjs';
+
+ const modalTriggerSubject = new Subject();
+ const transactionModalTriggerSubject = new Subject();
+
+ export function getAppsConnectModalTriggerObject() {
```

```
+     return {subject: modalTriggerSubject, observable: modalTriggerSubject.asObservable()};
+ }
+
+ export function getTransactionModalTriggerObject() {
+     return {subject: transactionModalTriggerSubject, observable: transactionModalTriggerSubject.asObservable()};
+ }
```

77 app/tonconnect/models.ts

```
@@ -0,0 +1,77 @@
+ import {AuthRequestBody, TonLoginClient} from "@tonapps/tonlogin-client";
+ import {ConnectReplyBuilder} from "./ConnectReplyBuilder";
+ import {ConnectItemReply} from "@tonconnect/protocol";
+
+ export interface IConnectQrQuery {
+   v: string;
+   r: string;
+   id: string;
+ }
+
+ export interface DAppManifest {
+   url: string;
+   name: string;
+   iconUrl: string;
+   termsOfUseUrl?: string;
+   privacyPolicyUrl?: string;
+ }
+
+ export interface TonConnectResponseModel {
+   address: string;
+   replyItems: ConnectItemReply[];
+ }
+
+ export type TonConnectModel = {
+   protocolVersion: 2;
+   manifest: DAppManifest;
+   replyBuilder: ConnectReplyBuilder;
+   requestPromise: {
+     resolve: (response: TonConnectResponseModel) => void;
+     reject: () => void;
+   };
+   hideImmediately: boolean;
+ };
+
+ export interface SignRawMessage {
+   address: string;
+   amount: string; // (decimal string): number of nanocoins to send.
+   payload?: string; // (string base64, optional): raw one-cell BoC encoded in Base64.
+   stateInit?: string; // (string base64, optional): raw once-cell BoC encoded in Base64.
+ }
+
+ export type TransactionModalParams = {
+   params: SignRawParams,
+   options: TxBodyOptions,
+   onSuccess?: (boc: string) => void,
+   onDismiss?: () => void,
+ } | null;
+
+ export type SignRawParams = {
+   source: string;
+   valid_until: number;
+   messages: SignRawMessage[];
+ };
+
+ export type TxBodyOptions = Omit<TxRequestBody, 'params' | 'type'>;
+
+ export type TxRequestBody<TParams = TxParams> = {
+   type: TxTypes;
+   expires_sec: number;
+   response_options: TxResponseOptions;
+   params: TParams;
+ };
+
+ export type TxParams =
+     | SignRawParams
+     | SignRawParams;
+
+ export type TxTypes =
+     | 'sign-raw-payload'
+     | 'deploy';
+
+ export type TxResponseOptions = {
+   broadcast: boolean;
+   return_url?: string;
+   callback_url?: string;
+   onDone?: () => void;
+ };
```

331 app/tonconnect/transactions/VerifyTransactionModal.tsx

```
@@ -0,0 +1,331 @@
+ import React, {useState} from "react";
+ import {Platform, Text, View} from "react-native";
+ import Modal from "react-native-modal";
+ import {Theme} from "../../Theme";
+ import LottieView from "lottie-react-native";
+ import {t} from "../../i18n/t";
+ import {RoundButton} from "../../components/RoundButton";
+ import {getTransactionModalTriggerObject} from "../modalTrigger";
+ import {base64ToCell, encodeBytes, getDarkModeValue} from "../../utils/misc";
+ import {contractFromPublicKey} from "../../engine/contractFromPublicKey";
+ import {getCurrentAddress} from "../../storage/appState";
+ import {loadWalletKeys, WalletKeys} from "../../storage/walletKeys";
+ import { Configuration, SendApi } from 'tonapi-sdk-js';
+ import {fromNano, toNano} from "ton";
+ import TonWeb from "tonweb";
+ import BN from "bn.js";
+ import {useEngine} from "../../engine/Engine";
+ import {useItem} from "../../engine/persistence/PersistedItem";
+ import {bearerToken, tonapiIOEndpoint} from "../../libs/Tonapi";
+ import {warn} from "../../utils/log";
+
+ type stateType = { action: string, message: string }; // 'requesting, rejected, approved'
+ type verifyTransactionView = any;
+
+ function VerifyTransactionModal() {
+     const engine = useEngine();
+     const account = useItem(engine.model.wallet(engine.address));
+     const acc = React.useMemo(() => getCurrentAddress(), []);
+     const balance = React.useMemo(() => {
+         return account.balance;
+     }, [account.balance]);
+     const transactionModalObject = getTransactionModalTriggerObject();
+     const [transactionObject, setTransactionObject] = useState<verifyTransactionView>(null);
+     const [currentAction, setCurrentAction] = useState<stateType>({action: 'requesting', message: ''}); // 'requesting, rejected, approved'
```

```
+    React.useEffect(() => {
+        const transactionModalSubscriber = transactionModalObject.observable.subscribe(async (transaction: any) => {
+            if (transaction) {
+                const messages = transaction.params.messages;
+                const target = new TonWeb.Address(messages[messages.length - 1].address).toString(true, true, true, false);
+                const amount = fromNano(messages[messages.length - 1].amount);
+                const resolver = transaction.tResolve;
+                const rejection = transaction.tReject;
+                setTransactionObject({target, amount, messages, resolve: resolver, reject: rejection});
+            }
+        });
+        return function cleanup() {
+            transactionModalSubscriber.unsubscribe();
+        }
+    }, [transactionModalObject]);
+
+    async function getWalletMethods(p_messages: any) {
+        const ton = new TonWeb();
+        const address = getCurrentAddress();
+        const wallet = await new ton.wallet.all['v4R2'](ton.provider, {
+            publicKey: address.publicKey,
+            wc: 0,
+        });
+        const seqno = await wallet.methods.seqno().call();
+        const sendMode = 3;
+        const signingMessage = (wallet as any).createSigningMessage(seqno);
+        let walletKeys: WalletKeys;
+        try {
+            walletKeys = await loadWalletKeys(acc.secretKeyEnc);
+        } catch (e) {
+            warn(e);
+            return;
+        }
+        const secretKey: Uint8Array = walletKeys.keyPair.secretKey;
+        const messages = p_messages.splice(0, 4);
+        for (let message of messages) {
+            const add = new TonWeb.utils.Address(message.address);
+            const order = TonWeb.Contract.createCommonMsgInfo(
+                TonWeb.Contract.createInternalMessageHeader(
+                    new TonWeb.Address(add),
+                    new TonWeb.utils.BN(message.amount),
+                ),
+                // @ts-ignore
+                base64ToCell(message.stateInit),
+                base64ToCell(message.payload),
+            );
+            signingMessage.bits.writeUint8(sendMode);
+            signingMessage.refs.push(order);
+        }
+
+        return TonWeb.Contract.createMethod(
+            wallet.provider,
+            (wallet as any).createExternalMessage(
+                signingMessage,
+                secretKey,
+                seqno,
+                !secretKey,
+            ),
+        );
+    }
+
+    const doSend = React.useCallback(async () => {
+        let address = transactionObject?.target;
+        let value: BN;
+        let errorMsg: string | null = null;
+
+        if (address) {
+            try {
+                if (transactionObject?.amount) {
+                    const validAmount = transactionObject?.amount.replace(',', '.');
+                    value = toNano(validAmount);
+                } else {
+                    errorMsg = t('transfer.error.invalidAmount');
+                    setCurrentAction({action: 'rejected', message: errorMsg});
+                    return;
+                }
+            } catch (e) {
+                errorMsg = t('transfer.error.invalidAmount');
+                setCurrentAction({action: 'rejected', message: errorMsg});
+                return;
+            }
+
+            // Load contract
+            const contract = await contractFromPublicKey(acc.publicKey);
+
+            // Check if same address
+            if (address && address === (contract.address.toFriendly({testOnly: false}))) {
+                errorMsg = t('transfer.error.sendingToYourself');
+            }
+
+            // Check amount
+            if (!value.eq(balance) && balance.lt(value)) {
+                errorMsg = t('transfer.error.insufficientAmountTitle'), t('transfer.error.notEnoughCoins');
+            }
+            if (value.eq(new BN(0))) {
+                errorMsg = t('transfer.error.zeroCoins');
+            }

+            if (errorMsg) {
+                setCurrentAction({action: 'rejected', message: errorMsg});
+                transactionObject.reject();
+                return;
+            } else {
+                const messages = transactionObject.messages;
+                const methods = await getWalletMethods(messages);
+                if (methods) {
+                    const queryMsg = await methods.getQuery();
+                    const boc = encodeBytes(await queryMsg.toBoc(false));
+                    const response = await sendAPI().sendBoc({sendBocRequest:{ boc }});
+                    transactionObject.resolve(boc);
+                    setCurrentAction({action: 'approved', message: 'Transaction approved!'});
+                    return response;
+                } else {
+                    setCurrentAction({action: 'rejected', message: 'Something went wrong.\nPlease try again.'});
+                    transactionObject.reject();
+                    return;
+                }
+            }
+        } else {
+            setCurrentAction({action: 'rejected', message: t('transfer.error.invalidAddress')});
+            transactionObject.reject();
+            return;
+        }

+    }, [transactionObject, acc]);
+
```

```
+    function sendAPI() {
+        return new SendApi(
+            new Configuration({
+                basePath: tonapiIOEndpoint,
+                headers: {
+                    Authorization: `Bearer ${bearerToken}`,
+                },
+            }),
+        )
+    }
+
+    const cancelRequest = React.useCallback(async () => {
+        if (transactionObject ) {
+            transactionObject.reject();
+            setCurrentAction({action: 'rejected', message: 'User cancelled the the request.'});
+        }
+    }, [transactionObject]);
+
+    const closeModal = React.useCallback(async () => {
+        setTransactionObject(null);
+        setCurrentAction({action: 'requesting', message: ''});
+    }, []);
+
+    return (
+        <Modal isVisible={!!transactionObject}>
+            <View style={{width: '100%', backgroundColor: Theme.item, paddingHorizontal: 5, paddingVertical: 10}}>
+                <Text style={{
+                    fontSize: 19,
+                    fontWeight: '600',
+                    color: Theme.textColor,
+                    textAlign: 'center',
+                    textAlignVertical: 'center'
+                }}>{t('products.transactionRequest.title')}</Text>
+            </View>
+            <View style={{backgroundColor: Theme.item}}>
+                <View style={{
+                    flexDirection: 'row',
+                    width: '100%',
+                    height: 145,
+                    backgroundColor: Theme.background,
+                    alignContent: 'center',
+                    alignItems: 'center'
+                }}>
+                    <View style={{width: '100%', alignItems: 'center', alignContent: 'center'}}>
+                        <LottieView
+                            source={require('../../../assets/animations/transaction-waiting.json')}
+                            style={{width: 140}}
+                            autoPlay={true}
+                        />
+                    </View>
+                </View>
+                {currentAction.action === 'requesting' && (
+                    <View style={{
+                        flexDirection: 'column',
+                        marginTop: 5,
+                        justifyContent: 'space-evenly',
+                        alignContent: 'stretch'
+                    }}>
+                        <View style={{
+                            backgroundColor: Theme.lightGreen,
+                            paddingLeft: 15,
+                            paddingRight: 15,
+                            paddingTop: 10,
+                            paddingBottom: 10,
+                            marginBottom: 5
+                        }}>
+                            <Text
+                                style={{
+                                    fontSize: 16,
+                                    textAlign: 'center',
+                                    color: Theme.textColor,
+                                    fontWeight: '400'
+                                }}
+                            >
+                                {`${t('common.amount')}: ${transactionObject?.amount} TON`}
+                            </Text>
+                            <Text
+                                style={{
+                                    fontSize: 16,
+                                    textAlign: 'center',
+                                    color: Theme.textColor,
+                                    fontWeight: '400',
+                                    fontVariant: ['tabular-nums'],
+                                    fontFamily: Platform.OS === 'ios' ? 'Menlo-Bold' : 'monospace'
+                                }}
+                            >
+                                {`${transactionObject?.target}`}
+                            </Text>
+                        </View>
+                        <View style={{paddingHorizontal: 16, paddingBottom: 16, marginTop: 10}}>
+                            <RoundButton
+                                title={t('common.confirm')}
+                                display={getDarkModeValue() ? 'default' : 'outline'}
+                                action={doSend}
+
+                            />
+                            <RoundButton
+                                title={t('common.cancel')}
+                                display={getDarkModeValue() ? 'outline' : 'default'}
+                                style={{marginTop: 10}}
+                                action={cancelRequest}
+                            />
+                        </View>
+                    </View>
+                )}
+
+                {currentAction.action === 'approved' && (
+                    <View style={{
+                        flexGrow: 1,
+                        width: '100%',
+                        alignContent: 'center',
+                        alignItems: 'center',
+                        paddingBottom: 20
+                    }}>
+                        <LottieView
+                            source={require('../../../assets/animations/green-check.json')}
+                            style={{width: 140}}
+                            autoPlay={true}
+                            loop={true}
+                        />
+                        <Text style={{
+                            marginTop: 5,
+                            fontSize: 16,
+                            fontWeight: '600',
+                            color: Theme.primeGreen,
+                            textAlign: 'center'
```

```
+                        }}>{currentAction.message}</Text>
+                        <RoundButton
+                            title={t('common.close')}
+                            display={getDarkModeValue() ? 'outline' : 'default'}
+                            style={{marginTop: 25, width: '60%'}}
+                            action={closeModal}
+                        />
+                    </View>
+                )}
+
+                {currentAction.action === 'rejected' && (
+                    <View style={{
+                        flexGrow: 1,
+                        width: '100%',
+                        alignContent: 'center',
+                        alignItems: 'center',
+                        paddingBottom: 20
+                    }}>
+                        <LottieView
+                            source={require('../../../assets/animations/red-cross.json')}
+                            style={{width: 140}}
+                            autoPlay={true}
+                            loop={true}
+                        />
+                        <Text style={{
+                            marginTop: 5,
+                            fontSize: 16,
+                            fontWeight: '600',
+                            color: Theme.primeRed
+                        }}>{currentAction.message}</Text>
+                        <RoundButton
+                            title={t('common.close')}
+                            display={getDarkModeValue() ? 'outline' : 'default'}
+                            style={{marginTop: 25, width: '60%'}}
+                            action={closeModal}
+                        />
+                    </View>
+                )}
+
+            </View>
+        </Modal>
+    );
+ }
+
+ export default VerifyTransactionModal;
```

app/utils/SupportedDomains.ts

```
@@ -4,6 +4,8 @@ export const SupportedDomains = [
     'tonsafe.com',
     'test.tonsafe.com',
     'app.tonkeeper.com',
+    'app.tonsafe.net',
+    'www.app.tonsafe.net',
     'www.tonhub.com',
     'www.test.tonhub.com',
     'www.app.tonkeeper.com',
```

app/utils/getTimeSec.ts

```
@@ -0,0 +1,3 @@
+ export function getTimeSec() {
+   return Math.floor(Date.now() / 1000);
+ }
```

app/utils/misc.ts

```
@@ -2,9 +2,31 @@ import {storage} from "../storage/storage";
 import {fetchPriceHistory} from "../engine/api/fetchPrice";
 import {getCurrentCurrencyObject, getTON2USDValue} from "./currency";
 import {formatDate} from "./dates";
+ import queryParser from 'query-string';
+ import {createHash} from "crypto";
+ import {findConnectedAppByClientSessionId, getAllConnections, IConnectedApp, IConnectedAppConnection,
+     useConnectedAppsStore
+ } from "../store";
+ import {AppConfig} from "../AppConfig";
+ import {getCurrentAddress} from "../storage/appState";
+ import naclUtils from "tweetnacl-util";
+ import {Cell} from "ton";
+ import TonWeb from "tonweb";

 export const minHideBalanceValue = 0.05;
-export let graphDetails = {data: [], hideIndices: [], dataAvailable: false, min: 0, max: 0, start: '', end: '', startPrice: null, endPrice: null,
fromCache: false, apiCallEnded: false};
+ export let graphDetails = {
+     data: [],
+     hideIndices: [],
+     dataAvailable: false,
+     min: 0,
+     max: 0,
+     start: '',
+     end: '',
+     startPrice: null,
+     endPrice: null,
+     fromCache: false,
+     apiCallEnded: false
+ };

 export function setHideBalanceValue(value: boolean) {
     storage.set('settings:hideSmallBalances', value ? 'true' : 'false');
@@ -120,7 +142,13 @@ export function parseHistoryData(history: any) {
         graphDataArr.push(parseFloat(price.toString().substring(0, 4)));
         graphDateArr.push(formatDate((new Date().getTime() / 1000)));
     }
-    return {data: graphDataArr, date: graphDateArr, hideIndices, min: Math.min.apply(Math, graphDataArr), max: Math.max.apply(Math, graphDataArr)};
+    return {
+        data: graphDataArr,
+        date: graphDateArr,
+        hideIndices,
+        min: Math.min.apply(Math, graphDataArr),
+        max: Math.max.apply(Math, graphDataArr)
+    };
 }

 export function getPriceHistory() {
@@ -143,11 +171,35 @@ export function getPriceHistory() {
                 };
                 resolve(graphDetails);
             } else {
-                graphDetails = {data: [], hideIndices: [], dataAvailable: false, min: 0, max: 0, start: '', end: '', startPrice: null, endPrice: null,
fromCache: false, apiCallEnded: true};
```

```
fromCache: false, apiCallEnded: true};
                graphDetails = {
                    data: [],
                    hideIndices: [],
                    dataAvailable: false,
                    min: 0,
                    max: 0,
                    start: '',
                    end: '',
                    startPrice: null,
                    endPrice: null,
                    fromCache: false,
                    apiCallEnded: true
                };
                resolve(graphDetails);
            }
        }, (err) => {
-           graphDetails = {data: [], hideIndices: [], dataAvailable: false, min: 0, max: 0, start: '', end: '', startPrice: null, endPrice: null,
  fromCache: false, apiCallEnded: true}; // Load cached results
+           graphDetails = {
+               data: [],
+               hideIndices: [],
+               dataAvailable: false,
+               min: 0,
+               max: 0,
+               start: '',
+               end: '',
+               startPrice: null,
+               endPrice: null,
+               fromCache: false,
+               apiCallEnded: true
+           }; // Load cached results
            resolve(graphDetails);
        });
    });
@@ -157,4 +209,97 @@ export function getFreshGraphData() {
    return graphDetails;
}

+ export const generateAppHashFromUrl = (url: string) => {
+     // get url without query
+     const {url: parsedUrl} = queryParser.parseUrl(url);
+
+     // remove last slash if it exists
+     const fixedUrl = parsedUrl.replace(/\/$/, '');
+
+     const hash = createHash('sha256').update(Buffer.from(fixedUrl)).digest('hex');
+
+     return hash;
+ };
+
+ export const getConnectedAppsReference = () => {
+     const {address} = getCurrentAddress();
+     const initialConnections = getAllConnections(
+         useConnectedAppsStore.getState(),
+         address.toFriendly({testOnly: AppConfig.isTestnet}),
+     );
+     if (initialConnections.length > 0) {
+         const apps: { connectedApp: IConnectedApp | null; connection: IConnectedAppConnection | null; }[] = [];
+         initialConnections.forEach((connection: any) => {
+             apps.push(findConnectedAppByClientSessionId(connection.clientSessionId));
+         });
+         return apps;
+     }
+     return [];
+ }
+
+ export function Utf8ArrayToString(array: Uint8Array): string {
+     let out = '';
+     let len = array.length;
+     let i = 0;
+     let c: any = null;
+     while (i < len) {
+         c = array[i++];
+         switch (c >> 4) {
+             case 0:
+             case 1:
+             case 2:
+             case 3:
+             case 4:
+             case 5:
+             case 6:
+             case 7:
+                 // 0xxxxxxx
+                 out += String.fromCharCode(c);
+                 break;
+             case 12:
+             case 13:
+                 // 110x xxxx   10xx xxxx
+                 let char = array[i++];
+                 out += String.fromCharCode(((c & 0x1f) << 6) | (char & 0x3f));
+                 break;
+             case 14:
+                 // 1110 xxxx  10xx xxxx  10xx xxxx
+                 let a = array[i++];
+                 let b = array[i++];
+                 out += String.fromCharCode(
+                     ((c & 0x0f) << 12) | ((a & 0x3f) << 6) | ((b & 0x3f) << 0),
+                 );
+                 break;
+         }
+     }
+     return out;
+ }
+
+ export function encodeBytes(bytes: Uint8Array) {
+     return naclUtils.encodeBase64(bytes);
+ }
+
+ export function base64ToCell(base64?: string): Cell | null {
+     if (base64) {
+         const bytes = new Uint8Array(Buffer.from(base64, 'base64'));
+         // @ts-ignore
+         return TonWeb.boc.Cell.oneFromBoc(bytes);
+     } else {
+         return null;
+     }
+ }
+
+ export let walletStateInit: string = '';
+
+ export async function setWalletStateInit () {
+     const ton = new TonWeb();
+     const address = getCurrentAddress();
+     const wallet = await new ton.wallet.all['v4R2'](ton.provider, {
+         publicKey: address.publicKey,
```

```
+        wc: 0,
+    });
+    const { stateInit } = await wallet.createStateInit();
+    walletStateInit = ton.utils.bytesToBase64(await stateInit.toBoc(false));
+ }
+
 export type AddressContact = { name: string, fields?: { key: string, value: string | null | undefined }[] | null } | null;
```

app/utils/resolveUrl.ts

```
@@ -15,6 +15,9 @@ type ResolvedUrl = {
     type: 'connect',
     session: string,
     endpoint: string | null
+ } | {
+    type: 'ton-connect',
+    query: any
 } | null;

 export function resolveUrl(src: string): ResolvedUrl {
@@ -73,20 +76,10 @@ export function resolveUrl(src: string): ResolvedUrl {
         }

         // ton url connect
-        if ((url.protocol.toLowerCase() === 'ton:' || url.protocol.toLowerCase() === 'ton-test:') && url.host.toLowerCase() === 'connect' && url.pathname.startsWith('/')) {
-            let session = url.pathname.slice(1);
-            let endpoint: string | null = null;
-            if (url.query) {
-                for (let key in url.query) {
-                    if (key.toLowerCase() === 'endpoint') {
-                        endpoint = url.query[key]!;
-                    }
-                }
-            }
+        if (((url.protocol.toLowerCase() === 'ton:' || url.protocol.toLowerCase() === 'ton-test:') && url.host.toLowerCase() === 'ton-connect') || (url.protocol.toLowerCase() === 'tc:')) {
             return {
-                type: 'connect',
-                session,
-                endpoint
+                type: 'ton-connect',
+                query: url.query
             }
         }

@@ -160,23 +153,13 @@ export function resolveUrl(src: string): ResolvedUrl {
             }
         }

-        // HTTP(s) Sign Url
+        // HTTP(s) Sign Url for ton-connect
         if ((url.protocol.toLowerCase() === 'http:' || url.protocol.toLowerCase() === 'https:')
             && (SupportedDomains.find((d) => d === url.host.toLowerCase()))
-            && (url.pathname.toLowerCase().startsWith('/connect/'))) {
-            let session = url.pathname.slice('/connect/'.length);
-            let endpoint: string | null = null;
-            if (url.query) {
-                for (let key in url.query) {
-                    if (key.toLowerCase() === 'endpoint') {
-                        endpoint = url.query[key]!;
-                    }
-                }
-            }
+            && (url.pathname.toLowerCase().includes('ton-connect'))) {
             return {
-                type: 'connect',
-                session,
-                endpoint
+                type: 'ton-connect',
+                query: url.query
             }
         }
```

assets/animations/connection.json

```
@@ -0,0 +1 @@
+ {"nm":"steps file","mn":"","layers":[{"ty":4,"nm":"Shape Layer 7","mn":"","sr":1,"st":0,"op":90,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[-121.284,1.716,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[201.886,50,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Ellipse 1","ix":1,"cix":2,"np":3,"it":[{"ty":"el","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Ellipse","nm":"Ellipse Path 1","d":1,"p":{"a":0,"k":[0,0],"ix":3},"s":{"a":0,"k":[21.432,21.432],"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke 1","lc":1,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":0,"ix":5},"d":[],"c":{"a":0,"k":[0,0.498,1],"ix":3}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0.898,0.3922,0.3333],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[65.353,65.353],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[-121.284,1.716],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]}],"ind":1},{"ty":4,"nm":"Shape Layer 6","mn":"","sr":1,"st":0,"op":90,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[-122.944,0.782,0],"ix":1},"s":{"a":0,"k":[111.817,111.817,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[201.886,50,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Ellipse 1","ix":1,"cix":2,"np":3,"it":[{"ty":"el","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Ellipse","nm":"Ellipse Path 1","d":1,"p":{"a":0,"k":[0,0],"ix":3},"s":{"a":0,"k":[68.408,68.408],"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke 1","lc":1,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":0,"ix":5},"d":[],"c":{"a":0,"k":[0,0.498,1],"ix":3}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0.9294,0.9294,0.9294],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[49.48,49.48],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[-122.944,0.782],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]}],"ind":2},{"ty":4,"nm":"Shape Layer 5","mn":"","sr":1,"st":0,"op":90,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[-121.284,1.716,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[356.886,50,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Ellipse 1","ix":1,"cix":2,"np":3,"it":[{"ty":"el","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Ellipse","nm":"Ellipse Path 1","d":1,"p":{"a":0,"k":[0,0],"ix":3},"s":{"a":0,"k":[21.432,21.432],"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke 1","lc":1,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":0,"ix":5},"d":[],"c":{"a":0,"k":[0,0.498,1],"ix":3}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0,0.7686,0.3725],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[65.353,65.353],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[-121.284,1.716],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]}],"ind":3},{"ty":4,"nm":"Shape Layer 4","mn":"","sr":1,"st":0,"op":90,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[-122.944,0.782,0],"ix":1},"s":{"a":0,"k":[111.817,111.817,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[356.886,50,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Ellipse 1","ix":1,"cix":2,"np":3,"it":[{"ty":"el","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Ellipse","nm":"Ellipse Path 1","d":1,"p":{"a":0,"k":[0,0],"ix":3},"s":{"a":0,"k":[68.408,68.408],"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke 1","lc":1,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":0,"ix":5},"d":[],"c":{"a":0,"k":[0,0.498,1],"ix":3}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0.9294,0.9294,0.9294],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[49.48,49.48],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[-122.944,0.782],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]}],"ind":4},{"ty":4,"nm":"Shape Layer 3","mn":"","sr":1,"st":0,"op":90,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[-121.284,1.716,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[41.886,50,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Ellipse 1","ix":1,"cix":2,"np":3,"it":[{"ty":"el","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Ellipse","nm":"Ellipse Path 1","d":1,"p":{"a":0,"k":[0,0],"ix":3},"s":{"a":0,"k":[21.432,21.432],"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke 1","lc":1,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":0,"ix":5},"d":[],"c":{"a":0,"k":[0,0.498,1],"ix":3}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0.9294,0.9294,0.9294],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[49.48,49.48],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[-122.944,0.782],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]}],"ind":5},{"ty":4,"nm":"Shape Layer 2","mn":"","sr":1,"st":0,"op":90,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[-121.284,1.716,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[201.886,50,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Ellipse 1","ix":1,"cix":2,"np":3,"it":[{"ty":"el","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Ellipse","nm":"Ellipse Path 1","d":1,"p":{"a":0,"k":[0,0],"ix":3},"s":{"a":0,"k":[21.432,21.432],"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke 1","lc":1,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":0,"ix":5},"d":[],"c":{"a":0,"k":[0,0.498,1],"ix":3}},
```

{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0,0.2549,0.4157],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[65.353,65.353],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[-121.284,1.716],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},{"ty":4,"nm":"Shape Layer 2","mn":"","sr":1,"st":0,"op":90,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[-122.944,0.782,0],"ix":1},"s":{"a":0,"k":[111.817,111.817,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[41.886,50,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Ellipse 1","ix":1,"cix":2,"np":3,"it":[{"ty":"el","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Ellipse","nm":"Ellipse Path 1","d":1,"p":{"a":0,"k":[0,0],"ix":3},"s":{"a":0,"k":[68.408,68.408],"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke 1","lc":1,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":0,"ix":5},"d":[],"c":{"a":0,"k":[0,0.498,1],"ix":3}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0.9294,0.9294,0.9294],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[49.48,49.48],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[-122.944,0.782],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},"ind":6},{"ty":4,"nm":"Shape Layer 1","mn":"","sr":1,"st":0,"op":90,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[-423.781,12.549,0],"ix":1},"s":{"a":0,"k":[49.758,21.985,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[41.77,50.5,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Rectangle 1","ix":1,"cix":2,"np":3,"it":[{"ty":"rc","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Rect","nm":"Rectangle Path 1","d":1,"p":{"a":0,"k":[0,0],"ix":3},"r":{"a":0,"k":0,"ix":4},"s":{"a":0,"k":[55.711,8.973],"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke 1","lc":1,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":0,"ix":5},"d":[],"c":{"a":0,"k":[0,0.2549,0.4157],"ix":3}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":1,"k":[{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[0,0.2549,0.4157],"t":0},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[0,0.7686,0.3725],"t":45},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[0.898,0.3922,0.3333],"t":68},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[0,0.2549,0.4157],"t":90}],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[86.786,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":1,"k":[{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[-434.781,12.549],"t":0},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[-102.781,12.549],"t":23},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[218.219,12.549],"t":45},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[-102.907,12.549],"t":68},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[-423.787,12.549],"t":90}],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},"ind":7},{"ty":4,"nm":"dashed line Outlines","mn":"","sr":1,"st":0,"op":90,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[297,7,0],"ix":1},"s":{"a":0,"k":[55.533,47.727,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[199,50,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Group 1","ix":1,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":{"c":false,"i":[[0,0],[0,0]],"o":[[0,0],[0,0]],"v":[[23,8],[580,8]]},"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke 1","lc":1,"lj":1,"ml":10,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":2,"ix":5},"d":[{"nm":"dash","mn":"","n":"d","v":{"a":0,"k":12,"ix":1}},{"nm":"gap","mn":"","n":"g","v":{"a":0,"k":6,"ix":2}},{"nm":"offset","mn":"","n":"o","v":{"a":0,"k":0,"ix":7}}],"c":{"a":0,"k":[0.8,0.8,0.8],"ix":3}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[0,0],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]}],"ind":8}],"ddd":0,"h":100,"w":400,"meta":{"a":"","k":"","d":"","g":"@lottiefiles/toolkit-js 0.21.3","tc":"#000000"},"v":"5.6.10","fr":30,"op":90,"ip":0,"assets":[]}

1 assets/animations/green-check.json

```
@@ -0,0 +1 @@
```


+ {"nm":"Comp 1","mn":"","layers":
[{"ty":0,"nm":"trait","mn":"","sr":1,"st":0,"op":150.000006109625,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[256,256,0],"ix":1},"s":{"a":0,"k":[15,15,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[263.334,471.109,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":90,"ix":10}},"ef":[],"w":512,"h":512,"refId":"comp_0","ind":1},
{"ty":0,"nm":"trait","mn":"","sr":1,"st":0,"op":150.000006109625,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[256,256,0],"ix":1},"s":{"a":0,"k":[15,15,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[51.641,253.275,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":-180,"ix":10}},"ef":[],"w":512,"h":512,"refId":"comp_0","ind":2},
{"ty":0,"nm":"trait","mn":"","sr":1,"st":0,"op":150.000006109625,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[256,256,0],"ix":1},"s":{"a":0,"k":[15,15,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[266.322,44.315,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":-90,"ix":10}},"ef":[],"w":512,"h":512,"refId":"comp_0","ind":3},
{"ty":0,"nm":"trait","mn":"","sr":1,"st":0,"op":150.000006109625,"ip":150.000006109625,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[256,256,0],"ix":1},"s":{"a":0,"k":[15,15,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[469.91,258.792,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"w":512,"h":512,"refId":"comp_0","ind":4},
{"ty":0,"nm":"firefly","mn":"","sr":1,"st":0,"op":150.000006109625,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[256,256,0],"ix":1},"s":{"a":0,"k":[20,20,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[400.635,189.708,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":-18.097,"ix":10}},"ef":[],"w":512,"h":512,"refId":"comp_1","ind":5},
{"ty":0,"nm":"firefly","mn":"","sr":1,"st":0,"op":150.000006109625,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[256,256,0],"ix":1},"s":{"a":0,"k":[20,20,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[359.413,150.912,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":-72.471,"ix":10}},"ef":[],"w":512,"h":512,"refId":"comp_1","ind":6},
{"ty":0,"nm":"firefly","mn":"","sr":1,"st":0,"op":150.000006109625,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[256,256,0],"ix":1},"s":{"a":0,"k":[30,30,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[396.894,150.961,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":-45.707,"ix":10}},"ef":[],"w":512,"h":512,"refId":"comp_1","ind":7},{"ty":0,"nm":"trait
2","mn":"","sr":1,"st":0,"op":150.000006109625,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[256,256,0],"ix":1},"s":{"a":0,"k":[-19.512,19.512,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[410.865,406.53,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":-135.205,"ix":10}},"ef":[],"w":512,"h":512,"refId":"comp_2","ind":8},{"ty":0,"nm":"trait
2","mn":"","sr":1,"st":0,"op":150.000006109625,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[256,256,0],"ix":1},"s":{"a":0,"k":[-19.512,19.512,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[105.535,402.598,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":-45.606,"ix":10}},"ef":[],"w":512,"h":512,"refId":"comp_2","ind":9},{"ty":0,"nm":"trait
2","mn":"","sr":1,"st":0,"op":150.000006109625,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[256,256,0],"ix":1},"s":{"a":0,"k":[19.512,19.512,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[104.864,111.71,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":-135.205,"ix":10}},"ef":[],"w":512,"h":512,"refId":"comp_2","ind":10},{"ty":0,"nm":"trait
2","mn":"","sr":1,"st":0,"op":150.000006109625,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[256,256,0],"ix":1},"s":{"a":0,"k":[19.512,19.512,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[416.722,113.206,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":-45.606,"ix":10}},"ef":[],"w":512,"h":512,"refId":"comp_2","ind":11},{"ty":4,"nm":"Shape Layer 2","mn":"","sr":1,"st":0,"op":150.000006109625,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[0,0,0],"ix":1},"s":{"a":0,"k":[69.59,69.59,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[236.888,240.258,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Shape 1","ix":1,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":{"c":false,"i":[[0,0],[0,0],[0,0]],"o":[[0,0],[0,0],[0,0]],"v":[[-76.426,37.999],[12.056,114.074],[169.991,-68.635]]},"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke 1","lc":2,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":35,"ix":5},"d":[],"c":{"a":0,"k":[1,1,1],"ix":3}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[-7,11],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},
{"ty":"tm","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Filter - Trim","nm":"Trim Paths 1","ix":2,"e":{"a":1,"k":[{"o":{"x":0.167,"y":0},"i":{"x":0.833,"y":-2.986},"s":[0],"t":0},{"o":{"x":0.167,"y":0.042},"i":{"x":0,"y":0.973},"s":[0],"t":14.791},{"o":{"x":0.828,"y":0.011},"i":{"x":0.833,"y":0.833},"s":[32],"t":19.791},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[100],"t":24.791251009768}],"ix":2},"o":{"a":0,"k":0,"ix":3},"s":{"a":0,"k":0,"ix":1},"m":1}],"ind":12},{"ty":4,"nm":"Shape Layer 1","mn":"","sr":1,"st":0,"op":150.000006109625,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[0,0,0],"ix":1},"s":{"a":0,"k":[80,80,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[256,256,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Ellipse 1","ix":1,"cix":2,"np":2,"it":[{"ty":"el","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Ellipse","nm":"Ellipse Path 1","d":1,"p":{"a":0,"k":[0,0],"ix":3},"s":{"a":1,"k":[{"o":{"x":0.654,"y":0.007},"i":{"x":0.976,"y":0.968},"s":[0,0],"t":0},{"o":{"x":0.346,"y":-4.83},"i":{"x":0.468,"y":1.057},"s":[401.025,401.025],"t":7},{"o":{"x":0.364,"y":0.031},"i":{"x":0.375,"y":1.543},"s":[372.7,372.7],"t":12},{"o":{"x":0.327,"y":-8.038},"i":{"x":0.833,"y":1},"s":[401.025,401.025],"t":16},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[401.025,401.025],"t":20.000000814616 7}],"ix":2}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0,0.7686,0.3725],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[5.992,3.49],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]}],"ind":13}],"ddd":0,"h":512,"w":512,"meta":{"a":"","k":"","d":"","g":"@lottiefiles/toolkit-js 0.21.3","tc":"#000000"},"v":"5.0.1","fr":29.970001220703 1,"op":45.000001832887 6,"ip":0,"assets":[{"nm":"","mn":"","layers":[{"ty":4,"nm":"Shape Layer 1","mn":"","sr":1,"st":0,"op":150.000006109625,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[0,0,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[256,256,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":1,"k":[{"o":{"x":0.167,"y":0.167},"i":{"x":0.572,"y":0.556},"s":[100],"t":7},{"o":{"x":0.641,"y":0.056},"i":{"x":0.833,"y":0.833},"s":[92.154],"t":13},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[30],"t":17.000000692424 2}],"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Shape 1","ix":1,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":{"c":false,"i":[[0,0],[0,0]],"o":[[0,0],[0,0]],"v":[[-230.4,[214.4]]},"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke 1","lc":2,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":70,"ix":5},"d":[],"c":{"a":0,"k":[0,0.7686,0.3725],"ix":3}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[0,0],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]}],"ind":13}]},{"id":"comp_0","fr":30},{"nm":"","mn":"","layers":[{"ty":4,"nm":"Shape Layer 1","mn":"","sr":1,"st":0,"op":150.000006109625,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[0,0,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[256,256,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Ellipse 1","ix":1,"cix":2,"np":2,"it":[{"ty":"el","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Ellipse","nm":"Ellipse Path 1","d":1,"p":{"a":0,"k":[0,0],"ix":3},"s":{"a":1,"k":[{"o":{"x":0.167,"y":0.167},"i":{"x":0,"y":0.997},"s":[40,40],"t":24},{"o":{"x":0.574,"y":-0.004},"i":{"x":0.833,"y":0.833},"s":[90,90],"t":27},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[18.394,18.394],"t":38.000001547771 7}],"ix":2}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[1,1,1],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":1,"k":[{"o":{"x":0.167,"y":0.167},"i":{"x":0.667,"y":1},"s":[-181.074,-5.414],"t":24},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[200,-5.414],"t":38.000001547771 7}],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":1,"k":[{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[0],"t":24},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[100],"t":25},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[100],"t":29},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":

```
[0],"t":38.000001547771}],"ix":7}}}}],"ind":1}],"id":"comp_1","fr":30},{"nm":"","mn":"","layers":[{"ty":4,"nm":"Shape Layer
1","mn":"","sr":1,"st":0,"op":150.000006109625,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":
[0,0,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[256,256,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":1,"k":[{"o":
{"x":0.167,"y":0.167},"i":{"x":0.572,"y":0.556},"s":[100],"t":10},{"o":{"x":0.641,"y":0.056},"i":{"x":0.833,"y":0.833},"s":[92.154],"t":16},{"o":
{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[30],"t":20.0000008146167}],"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":
[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Shape 1","ix":1,"cix":2,"np":2,"it":
[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":{"c":false,"i":[[0,0],
[0,0]],"o":[[0,0],[0,0]],"v":[[-230,4],[214,4]]},"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke
1","lc":2,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":70,"ix":5},"d":[],"c":{"a":0,"k":[0,0.7686,0.3725],"ix":3}},{"ty":"tr","a":{"a":0,"k":
[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[0,0],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":
{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},{"ty":"tm","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Filter - Trim","nm":"Trim Paths
1","ix":2,"e":{"a":1,"k":[{"o":{"x":0.457,"y":0.063},"i":{"x":0.566,"y":0.999},"s":[0],"t":10},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":
[100],"t":19.0000007738859}],"ix":2},"o":{"a":0,"k":0,"ix":3},"s":{"a":1,"k":[{"o":{"x":0.167,"y":0.167},"i":{"x":0.588,"y":-51709.363},"s":[0],"t":10},
{"o":{"x":0.509,"y":0.003},"i":{"x":0.696,"y":0.999},"s":[0],"t":13},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":
[100],"t":19.0000007738859}],"ix":1},"m":1}],"ind":1}],"id":"comp_2","fr":30}]}
```


1 assets/animations/link-app.json

```
@@ -0,0 +1 @@
```


```
+ {"nm":"Link","mn":"","layers":
[{"ty":3,"nm":"linkTopController","mn":"","sr":1,"st":0,"op":75,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":
{"a":0,"k":[50,50,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":1,"k":[{"o":{"x":0.596,"y":0},"i":{"x":0.238,"y":1},"s":
[220.5,50,0],"t":0},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[155.5,120,0],"t":50}],"ix":2},"sa":{"a":0,"k":0},"o":
{"a":0,"k":0,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"ind":1},
{"ty":3,"nm":"linkBottomController","mn":"","sr":1,"st":0,"op":75,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":
{"a":0,"k":[50,50,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":1,"k":[{"o":{"x":0.596,"y":0},"i":{"x":0.238,"y":1},"s":
[80.5,250.5,0],"t":0},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[144.5,186.5,0],"t":50}],"ix":2},"sa":{"a":0,"k":0},"o":
{"a":0,"k":0,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"ind":2},{"ty":4,"nm":"linkTop
3","mn":"","sr":1,"st":10,"op":75,"ip":16,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":
[5.463,-29.199,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[60.963,-15.699,0],"ix":2},"sa":{"a":0,"k":0},"o":
{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Shape
1","ix":1,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":
{"c":false,"i":[[0,0],[-16.5,12.5],[0,0],[11,12.5],[0,0],[0,0]],"o":[[0,0],[16.5,-12.5],[0,0],[-11,-12.5],[0,0],[0,0]],"v":[[-9,11],[17.5,12.5],
[51,-20.5],[49.25,-47],[25.5,-49],[5.75,-30]]},"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke
1","lc":2,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":11,"ix":5},"d":[],"c":{"a":0,"k":[0.8392,0.9922,0.5137],"ix":3}},{"ty":"tr","a":
{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[0,0],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":
{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},{"ty":"tm","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Filter - Trim","nm":"Trim Paths
1","ix":2,"e":{"a":1,"k":[{"o":{"x":0.333,"y":0},"i":{"x":0.252,"y":1},"s":[0],"t":16},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":
[100],"t":61}],"ix":2},"o":{"a":0,"k":0,"ix":3},"s":{"a":0,"k":0,"ix":1},"m":1}],"ind":3,"parent":2},{"ty":4,"nm":"linkTop
2","mn":"","sr":1,"st":6,"op":75,"ip":12,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":
[6.056,-29.584,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[61.556,-16.084,0],"ix":2},"sa":{"a":0,"k":0},"o":
{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Shape
1","ix":1,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":
{"c":false,"i":[[0,0],[-16.5,12.5],[0,0],[11,12.5],[0,0],[0,0]],"o":[[0,0],[16.5,-12.5],[0,0],[-11,-12.5],[0,0],[0,0]],"v":[[-9,11],[17.5,12.5],
[51,-20.5],[49.25,-47],[25.5,-49],[5.75,-30]]},"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke
1","lc":2,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":11,"ix":5},"d":[],"c":{"a":0,"k":[0.8392,0.9922,0.5137],"ix":3}},{"ty":"tr","a":
{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[0,0],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":
{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},{"ty":"tm","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Filter - Trim","nm":"Trim Paths
1","ix":2,"e":{"a":1,"k":[{"o":{"x":0.333,"y":0},"i":{"x":0.252,"y":1},"s":[0],"t":12},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":
[100],"t":57}],"ix":2},"o":{"a":0,"k":0,"ix":3},"s":{"a":0,"k":0,"ix":1},"m":1}],"ind":4,"parent":2},
{"ty":4,"nm":"linkTop","mn":"","sr":1,"st":2,"op":75,"ip":8,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":
[5.567,-29.688,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[61.067,-16.188,0],"ix":2},"sa":{"a":0,"k":0},"o":
{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Shape
1","ix":1,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":
{"c":false,"i":[[0,0],[-16.5,12.5],[0,0],[11,12.5],[0,0],[0,0]],"o":[[0,0],[16.5,-12.5],[0,0],[-11,-12.5],[0,0],[0,0]],"v":[[-9,11],[17.5,12.5],
[51,-20.5],[49.25,-47],[25.5,-49],[5.75,-30]]},"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke
1","lc":2,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":11,"ix":5},"d":[],"c":{"a":0,"k":[0.8392,0.9922,0.5137],"ix":3}},{"ty":"tr","a":
{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[0,0],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":
{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},{"ty":"tm","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Filter - Trim","nm":"Trim Paths
1","ix":2,"e":{"a":1,"k":[{"o":{"x":0.333,"y":0},"i":{"x":0.252,"y":1},"s":[0],"t":8},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":
[100],"t":53}],"ix":2},"o":{"a":0,"k":0,"ix":3},"s":{"a":0,"k":0,"ix":1},"m":1}],"ind":5,"parent":2},{"ty":4,"nm":"linkBottom
2","mn":"","sr":1,"st":8,"op":75,"ip":8,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":
[7.536,-34.104,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[40.127,116.008,0],"ix":2},"sa":{"a":0,"k":0},"o":
{"a":0,"k":100,"ix":11},"r":{"a":0,"k":-180,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Shape
1","ix":1,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":
{"c":false,"i":[[0,0],[0,0],[-11,-12.5],[0,0],[16.963,-11.864],[1.162,3.655]],"o":[[0,0],[0,0],[11,12.5],[0,0],[-10.588,7.405],[-1.162,-3.655]],"v":
[[8.5,-34],[26.5,-50.5],[50,-48],[51,-20.5],[18.5,10.75],[-4,7]]},"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic -
Stroke","nm":"Stroke 1","lc":2,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":12,"ix":5},"d":[],"c":{"a":0,"k":[0.7686,0.3725],"ix":3}},
{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[0,0],"ix":2},"r":
{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},{"ty":"tm","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Filter -
Trim","nm":"Trim Paths 1","ix":2,"e":{"a":1,"k":[{"o":{"x":0.333,"y":0},"i":{"x":0.252,"y":1},"s":[0],"t":8},{"o":{"x":0.167,"y":0.167},"i":
{"x":0.833,"y":0.833},"s":[100],"t":53}],"ix":2},"o":{"a":0,"k":0,"ix":3},"s":{"a":0,"k":0,"ix":1},"m":1}],"ind":6,"parent":1},{"ty":4,"nm":"linkBottom
3","mn":"","sr":1,"st":4,"op":75,"ip":4,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":
[8.243,-34.126,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[39.419,116.03,0],"ix":2},"sa":{"a":0,"k":0},"o":
{"a":0,"k":100,"ix":11},"r":{"a":0,"k":-180,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Shape
1","ix":1,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":
{"c":false,"i":[[0,0],[0,0],[-11,-12.5],[0,0],[16.963,-11.864],[1.162,3.655]],"o":[[0,0],[0,0],[11,12.5],[0,0],[-10.588,7.405],[-1.162,-3.655]],"v":
[[8.5,+34],[26.5,-50.5],[50,-48],[51,-20.5],[18.5,10.75],[-4,7]]},"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic -
Stroke","nm":"Stroke 1","lc":2,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":11,"ix":5},"d":[],"c":{"a":0,"k":[0.0392,0.9922,0.5137],"ix":3}},
{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[0,0],"ix":2},"r":
{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},{"ty":"tm","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Filter -
Trim","nm":"Trim Paths 1","ix":2,"e":{"a":1,"k":[{"o":{"x":0.333,"y":0},"i":{"x":0.252,"y":1},"s":[0],"t":4},{"o":{"x":0.167,"y":0.167},"i":
{"x":0.833,"y":0.833},"s":[100],"t":49}],"ix":2},"o":{"a":0,"k":0,"ix":3},"s":{"a":0,"k":0,"ix":1},"m":1}],"ind":7,"parent":1},
{"ty":4,"nm":"linkBottom","mn":"","sr":1,"st":0,"op":75,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":
[7.65,-34.308,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[40.012,116.213,0],"ix":2},"sa":{"a":0,"k":0},"o":
{"a":0,"k":100,"ix":11},"r":{"a":0,"k":-180,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Shape
1","ix":1,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":
{"c":false,"i":[[0,0],[0,0],[-11,-12.5],[0,0],[16.963,-11.864],[1.162,3.655]],"o":[[0,0],[0,0],[11,12.5],[0,0],[-10.588,7.405],[-1.162,-3.655]],"v":
[[8.5,-34],[26.5,-50.5],[50,-48],[51,-20.5],[18.5,10.75],[-4,7]]},"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic -
Stroke","nm":"Stroke 1","lc":2,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":11,"ix":5},"d":[],"c":{"a":0,"k":[0.0392,0.9922,0.5137],"ix":3}},
{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[0,0],"ix":2},"r":
{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},{"ty":"tm","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Filter -
Trim","nm":"Trim Paths 1","ix":2,"e":{"a":1,"k":[{"o":{"x":0.333,"y":0},"i":{"x":0.252,"y":1},"s":[0],"t":0},{"o":{"x":0.167,"y":0.167},"i":
{"x":0.833,"y":0.833},"s":[100],"t":45}],"ix":2},"o":{"a":0,"k":0,"ix":3},"s":
{"a":0,"k":0,"ix":1},"m":1}],"ind":8,"parent":1}],"ddd":0,"h":300,"w":300,"meta":{"a":"","k":"","d":"","g":"LottieFiles AE
0.1.20","tc":"#000000"},"v":"5.5.7","fr":25,"op":75,"ip":0,"assets":[]}
```


1 assets/animations/red-cross.json

```
@@ -0,0 +1 @@
```


```
+ {"nm":"Cancel JSON","mn":"","layers":[{"ty":4,"nm":"cancel line
1","mn":"","sr":1,"st":89,"op":14489,"ip":89,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":
[12.09,12.76,0],"ix":1},"s":{"a":0,"k":[87.717,87.717,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[0.956,-1.526,0],"ix":2},"sa":{"a":0,"k":0},"o":
{"a":0,"k":100,"ix":11},"r":{"a":0,"k":90,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Group
1","ix":1,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":
{"c":false,"i":[[0,0],[0,0]],"o":[[0,0],[0,0]],"v":[[24.18,0],[0,25.52]]},"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic
- Stroke","nm":"Stroke 1","lc":2,"lj":2,"ml":1,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":6.091,"ix":5},"d":[],"c":{"a":0,"k":[1,1,1],"ix":3}},
{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[0,0],"ix":2},"r":
{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},{"ty":4,"nm":"cancel line
2","mn":"","sr":1,"st":89,"op":14489,"ip":89,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":
[12.09,12.76,0],"ix":1},"s":{"a":0,"k":[87.717,87.717,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[0.956,-1.526,0],"ix":2},"sa":{"a":0,"k":0},"o":
{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Group
1","ix":1,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":
{"c":false,"i":[[0,0],[0,0]],"o":[[0,0],[0,0]],"v":[[24.18,0],[0,25.52]]},"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic
- Stroke","nm":"Stroke 1","lc":2,"lj":2,"ml":1,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":6.091,"ix":5},"d":[],"c":{"a":0,"k":[1,1,1],"ix":3}},
{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[0,0],"ix":2},"r":
{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]}],"ind":2,"parent":3},{"ty":3,"nm":"pending symbol
ctrl","mn":"","sr":1,"st":89,"op":14489,"ip":89,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[0,0,0],"ix":1},"s":
{"a":1,"k":[{"o":{"x":0.333,"y":0},"i":{"x":0.667,"y":1},"s":[0,0,100],"t":89},{"o":{"x":0.333,"y":0},"i":{"x":0.667,"y":1},"s":[130,130,100],"t":99},
{"o":{"x":0.333,"y":0},"i":{"x":0.667,"y":1},"s":[80,80,100],"t":109},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":
[100,100,100],"t":119}],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[0,0,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":
{"a":0,"k":0,"ix":10}},"ef":[],"ind":3,"parent":4},{"ty":4,"nm":"pending
yellow","mn":"","sr":1,"st":89,"op":14489,"ip":89,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":
```

{"x":0.333,"y":0},"i":{"x":0,"y":0.989},"s":[0],"t":89},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[100],"t":109}],"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Mask","ix":1,"cix":2,"np":2,"it":[{"ty":"el","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Ellipse","nm":"Ellipse Path 1","d":1,"p":{"a":0,"k":[0,0],"ix":3},"s":{"a":0,"k":[96,96],"ix":2}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0.898,0.3922,0.3333],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":100,"ix":5},"p":{"a":0,"k":0,"ix":4},"s":{"a":0,"k":100,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},"ind":4},{"ty":4,"nm":"green circle 3","mn":"","sr":1,"st":39,"op":120,"ip":79,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[0,0,0],"ix":1},"s":{"a":1,"k":[{"o":{"x":0.333,"y":0},"i":{"x":0,"y":0.994},"s":[0,0,100],"t":79},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[130,130,100],"t":119}],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[65,65,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":1,"k":[{"o":{"x":0.333,"y":0},"i":{"x":0,"y":1.008},"s":[100],"t":79},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[0],"t":119}],"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Mask","ix":1,"cix":2,"np":3,"it":[{"ty":"el","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Ellipse","nm":"Ellipse Path 1","d":1,"p":{"a":0,"k":[0,0],"ix":3},"s":{"a":0,"k":[96,96],"ix":2}},{"ty":"tm","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Filter - Trim","nm":"Trim Paths 1","ix":2,"e":{"a":0,"k":100,"ix":2},"o":{"a":0,"k":0,"ix":3},"s":{"a":0,"k":0,"ix":1},"m":1},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke 1","lc":1,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":4,"ix":5},"d":[],"c":{"a":0,"k":[0.898,0.3922,0.3333],"ix":3}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[0,0],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]}],"ind":5}],"ddd":0,"h":130,"w":130,"meta":{"a":"","k":"","d":"","g":"@lottiefiles/toolkit-js 0.21.3","tc":"#000000"},"v":"5.3.4","fr":34,"op":130,"ip":79,"assets":[]}

1 assets/animations/transaction-waiting.json

@@ -0,0 +1 @@

+ {"nm":"Comp 1","mn":"","layers":[{"ty":4,"nm":"dompet Outlines","mn":"","sr":1,"st":0,"op":300.00001221925,"ip":0,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[250,250,0],"ix":1},"s":{"a":1,"k":[{"o":{"x":0.167,"y":0.167},"i":{"x":0.362,"y":1},"s":[0,0,100],"t":0},{"o":{"x":0.333,"y":0},"i":{"x":0.833,"y":1},"s":[80,80,100],"t":10},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[76,76,100],"t":14.0000005702317}],"ix":6},"sk":{"a":0,"k":0},"p":{"a":0,"k":[250,250,0],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":0,"k":0,"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Group 1","ix":1,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":{"c":true,"i":[[0,0],[0,0],[0,0],[0,0]],"o":[[0,0],[0,0],[0,0],[0,0]],"v":[[22.5,76],[-22.5,76],[-22.5,-76],[22.5,-76]]},"ix":2}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0.0,0.7686,0.3725],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[176.5,261],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Group 2","ix":2,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":{"c":true,"i":[[0,0],[0,0],[0,0],[0,0]],"o":[[0,0],[0,0],[0,0],[0,0]],"v":[[22.5,76],[-22.5,76],[-22.5,-76],[22.5,-76]]},"ix":2}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0.0,0.8196,0.4118],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[176.5,240],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Group 3","ix":3,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":{"c":true,"i":[[13.393,0],[0,0],[0,13.393],[0,0],[-13.393,0],[0,-13.393],[0,0]],"o":[[0,0],[-13.393,0],[0,0],[0,-13.393],[13.393,0],[0,0],[0,13.393]],"v":[[0,76],[0,76],[-24.25,51.75],[-24.25,-51.75],[0,-76],[24.25,-51.75],[24.25,51.75]]},"ix":2}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0.0,0.7294,0.3647],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[154.25,261],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Group 4","ix":4,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":{"c":true,"i":[[13.393,0],[0,0],[0,13.393],[0,0],[-13.393,0],[0,-13.393],[0,0]],"o":[[0,0],[-13.393,0],[0,0],[0,-13.393],[13.393,0],[0,0],[0,13.393]],"v":[[0,76],[0,76],[-24.25,51.75],[-24.25,-51.75],[0,-76],[24.25,-51.75],[24.25,51.75]]},"ix":2}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0.0,0.851,0.4275],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[154.25,240],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Group 5","ix":5,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":{"c":true,"i":[[0,-10.112],[10.113,0],[0,10.112],[-10.111,0]],"o":[[0,10.112],[-10.111,0],[0,-10.112],[10.113,0]],"v":[[18.309,-0.001],[-0.001,18.309],[-18.309,-0.001],[-0.001,-18.311]]},"ix":2}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0.6235,1,0.8118],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[295.619,258.186],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Group 6","ix":6,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":{"c":true,"i":[[11.598,0],[0,0],[0,11.599],[0,0],[-11.598,0],[0,0],[0,-11.598],[0,0]],"o":[[0,0],[-11.598,0],[0,0],[0,-11.598],[0,0],[11.598,0],[0,0],[0,11.599]],"v":[[34.5,32.613],[-34.5,32.613],[-55.5,11.613],[-55.5,-11.613],[-34.5,-32.613],[34.5,-32.613],[55.5,-11.613],[55.5,11.613]]},"ix":2}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0.0392,0.9922,0.5137],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[314.5,257.613],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Group 7","ix":7,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":{"c":true,"i":[[13.393,0],[0,0],[0,13.393],[0,0],[-13.393,0],[0,-13.393],[0,0]],"o":[[0,0],[-13.393,0],[0,0],[0,-13.393],[0,0],[13.393,0],[0,0],[0,13.393]],"v":[[84.5,76],[-84.5,76],[-108.75,51.75],[-108.75,-51.75],[-84.5,-76],[84.5,-76],[108.75,-51.75],[108.75,51.75]]},"ix":2}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0.0,0.7686,0.3725],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[238.75,261],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Group 8","ix":8,"cix":2,"np":2,"it":[{"ty":"sh","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Group","nm":"Path 1","ix":1,"d":1,"ks":{"a":0,"k":{"c":true,"i":[[13.393,0],[0,0],[0,13.393],[0,0],[-13.393,0],[0,-13.393],[0,0]],"o":[[0,0],[-13.393,0],[0,0],[0,-13.393],[0,0],[13.393,0],[0,0],[0,13.393]],"v":[[84.25,76.325],[-84.25,76.325],[-108.5,52.075],[-108.5,-52.075],[-84.25,-76.325],[84.25,-76.325],[108.5,-52.075],[108.5,52.075]]},"ix":2}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[0.0,0.8196,0.4118],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[100,100],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[238.5,240.325],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]}],"ind":1},{"ty":4,"nm":"Shape Layer 4","mn":"","sr":1,"st":21.0000008553475,"op":310.000012626559,"ip":21.0000008553475,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[0,0,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":1,"k":[{"o":{"x":0.282,"y":0.278},"i":{"x":0.395,"y":1},"s":[250,415,0],"t":21},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[250,250,0],"t":34.0000013848484}],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":1,"k":[{"o":{"x":0.335,"y":0},"i":{"x":0.667,"y":1},"s":[0],"t":23},{"o":{"x":0.167,"y":0},"i":{"x":0.667,"y":1},"s":[360],"t":76},{"o":{"x":0.167,"y":0},"i":{"x":0.667,"y":1},"s":[720],"t":128},{"o":{"x":0.167,"y":0},"i":{"x":0.667,"y":1},"s":[1080],"t":180},{"o":{"x":0.167,"y":0},"i":{"x":0.667,"y":1},"s":[1440],"t":232},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[1800],"t":284.000011567557}],"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Ellipse 1","ix":1,"cix":2,"np":3,"it":[{"ty":"el","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Ellipse","nm":"Ellipse Path 1","d":1,"p":{"a":0,"k":[0,0],"ix":3},"s":{"a":0,"k":[67,67],"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke 1","lc":1,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":0,"ix":5},"d":[],"c":{"a":0,"k":[0.8078,0.0431,0.0431],"ix":3}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[1,0.7373,0],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[77.596,77.596],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[-3.5,-183.5],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]}],"ind":2},{"ty":4,"nm":"Shape Layer 3","mn":"","sr":1,"st":18.0000007331555,"op":307.000012504366,"ip":18.0000007331555,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[0,0,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":1,"k":[{"o":{"x":0.282,"y":0.278},"i":{"x":0.395,"y":1},"s":[250,415,0],"t":18},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[250,250,0],"t":31.0000012626559}],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":1,"k":[{"o":{"x":0.335,"y":0},"i":{"x":0.667,"y":1},"s":[0],"t":20},{"o":{"x":0.167,"y":0},"i":{"x":0.667,"y":1},"s":[360],"t":73},{"o":{"x":0.167,"y":0},"i":{"x":0.667,"y":1},"s":[720],"t":125},{"o":{"x":0.167,"y":0},"i":{"x":0.667,"y":1},"s":[1080],"t":177},{"o":{"x":0.167,"y":0},"i":{"x":0.667,"y":1},"s":[1440],"t":229},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[1800],"t":281.000011445365}],"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Ellipse 1","ix":1,"cix":2,"np":3,"it":[{"ty":"el","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Ellipse","nm":"Ellipse Path 1","d":1,"p":{"a":0,"k":[0,0],"ix":3},"s":{"a":0,"k":[67,67],"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke 1","lc":1,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":0,"ix":5},"d":[],"c":{"a":0,"k":[0.8078,0.0431,0.0431],"ix":3}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[1,0.7373,0],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[77.596,77.596],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[-3.5,-183.5],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]}],"ind":3},{"ty":4,"nm":"Shape Layer 2","mn":"","sr":1,"st":15.0000006109625,"op":304.000012382174,"ip":15.0000006109625,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[0,0,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":1,"k":[{"o":{"x":0.282,"y":0.278},"i":{"x":0.395,"y":1},"s":[250,415,0],"t":15},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[250,250,0],"t":28.0000011404634}],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":1,"k":[{"o":{"x":0.335,"y":0},"i":{"x":0.667,"y":1},"s":[0],"t":17},{"o":{"x":0.167,"y":0},"i":{"x":0.667,"y":1},"s":[360],"t":70},{"o":{"x":0.167,"y":0},"i":{"x":0.667,"y":1},"s":[720],"t":122},{"o":{"x":0.167,"y":0},"i":{"x":0.667,"y":1},"s":[1080],"t":174},{"o":{"x":0.167,"y":0},"i":{"x":0.667,"y":1},"s":[1440],"t":226},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[1800],"t":278.000011323172}],"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Ellipse 1","ix":1,"cix":2,"np":3,"it":[{"ty":"el","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Ellipse","nm":"Ellipse Path 1","d":1,"p":{"a":0,"k":[0,0],"ix":3},"s":{"a":0,"k":[67,67],"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke 1","lc":1,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":0,"ix":5},"d":[],"c":{"a":0,"k":[0.8078,0.0431,0.0431],"ix":3}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[1,0.7373,0],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[77.596,77.596],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[-3.5,-183.5],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]}],"ind":4},{"ty":4,"nm":"Shape Layer 1","mn":"","sr":1,"st":12.00000048877,"op":301.000012259981,"ip":12.00000048877,"hd":false,"cl":"","ln":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":{"a":0,"k":[0,0,0],"ix":1},"s":{"a":0,"k":[100,100,100],"ix":6},"sk":{"a":0,"k":0},"p":{"a":1,"k":[{"o":{"x":0.282,"y":0.278},"i":{"x":0.395,"y":1},"s":[250,415,0],"t":12},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[250,250,0],"t":25.0000010182709}],"ix":2},"sa":{"a":0,"k":0},"o":{"a":0,"k":100,"ix":11},"r":{"a":1,"k":[{"o":{"x":0.335,"y":0},"i":{"x":0.667,"y":1},"s":[0],"t":14},{"o":{"x":0.167,"y":0},"i":{"x":0.667,"y":1},"s":[360],"t":67},{"o":{"x":0.167,"y":0},"i":{"x":0.667,"y":1},"s":[720],"t":119},{"o":{"x":0.167,"y":0},"i":{"x":0.667,"y":1},"s":[1080],"t":171},{"o":{"x":0.167,"y":0},"i":{"x":0.667,"y":1},"s":[1440],"t":223},{"o":{"x":0.167,"y":0.167},"i":{"x":0.833,"y":0.833},"s":[1800],"t":275.000011200985}],"ix":10}},"ef":[],"shapes":[{"ty":"gr","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Group","nm":"Ellipse

1","ix":1,"cix":2,"np":3,"it":[{"ty":"el","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Shape - Ellipse","nm":"Ellipse Path 1","d":1,"p":{"a":0,"k":[0,0],"ix":3},"s":{"a":0,"k":[67,67],"ix":2}},{"ty":"st","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Stroke","nm":"Stroke 1","lc":1,"lj":1,"ml":4,"o":{"a":0,"k":100,"ix":4},"w":{"a":0,"k":0,"ix":5},"d":[],"c":{"a":0,"k":[0.8078,0.0431,0.0431],"ix":3}},{"ty":"fl","bm":0,"cl":"","ln":"","hd":false,"mn":"ADBE Vector Graphic - Fill","nm":"Fill 1","c":{"a":0,"k":[1,0.7373,0],"ix":4},"r":1,"o":{"a":0,"k":100,"ix":5}},{"ty":"tr","a":{"a":0,"k":[0,0],"ix":1},"s":{"a":0,"k":[77.596,77.596],"ix":3},"sk":{"a":0,"k":0,"ix":4},"p":{"a":0,"k":[-3.5,-183.5],"ix":2},"r":{"a":0,"k":0,"ix":6},"sa":{"a":0,"k":0,"ix":5},"o":{"a":0,"k":100,"ix":7}}]},"ind":5}],"ddd":0,"h":500,"w":500,"meta":{"a":"","k":"","d":"","g":"@lottiefiles/toolkit-js 0.21.3","tc":"#000000"},"v":"5.5.7","fr":29.9700012207031,"op":300.00001221925,"ip":0,"assets":[]}

4 global.js

```
@@ -0,0 +1,4 @@
+ /* eslint-disable no-undef */
+ // Inject node globals into React Native global scope.
+ global.Buffer = global.Buffer || require('buffer').Buffer;
+ global.process = require('process');
```

4 index.js

```
@@ -1,5 +1,7 @@
- global.Buffer = global.Buffer || require('buffer').Buffer;
+ import './global';
+ import 'react-native-get-random-values';
  import 'react-native-url-polyfill/auto';
+ import 'text-encoding-polyfill';

  // Disable warnings
  import { LogBox } from 'react-native';
```

206 ios/wallet/Info.plist

Load diff

9 metro.config.js

Load diff

37 package.json

Load diff

32 shim.js

Load diff

945 yarn.lock

Load diff