

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

Unwatch 1 Fork 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.

6 commits 90 files changed 1 contributor

Commits on Jan 10, 2023

Ton Connect 2.0 is in progress. codexer007 committed on Jan 10

Commits on Jan 12, 2023

Ton Connect 2.0 initial connection made to the demo-app that ton-conn... codexer007 committed on Jan 12

Commits on Jan 19, 2023

Work in progress codexer007 committed last month

Commits on Jan 25, 2023

Initial functionality done codexer007 committed 3 weeks ago

Commits on Feb 7, 2023

Ton Connect 2.0 released for TonSafe v2.2 codexer007 committed last week

Commits on Feb 14, 2023

Ton Connect 2.0 integration - Modifications done as per wallets.json ... codexer007 committed 23 minutes ago

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

App.tsx

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

VERSION_CODE

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

```

android/app/build.gradle

```
@@ -359,6 +359,18 @@ dependencies {
359   implementation jsFlavor
360   }
361
362 +
363 + // SMOKE IMPLEMENTATION - DUPLICATE CLASSES ISSUE - https://youtrack.jetbrains.com/issue/KT-54136/Duplicated-classes-cause-build-failure-if-a-dependency-to-kotlin-stdlib-is-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.firebaseio:firebase-bom:29.0.4')
363 375     implementation 'com.google.firebaseio:firebase-messaging:23.0.0'
364 376     implementation 'com.google.firebaseio:firebase-installations:17.0.0'
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
975
```

```
        android:exported="true" android:label="@string/app_name" android:launchMode="singleTask" android:screenOrientation="portrait"
        android:theme="@style/Theme.App.SplashScreen" android:windowSoftInputMode="adjustResize">
    21 +        <intent-filter>
    22 +            <action android:name="android.intent.action.MAIN" />
    23 +            <category android:name="android.intent.category.LAUNCHER" />
    24 +        </intent-filter>
    25 +        <intent-filter>
    26 +            <action android:name="android.intent.action.VIEW" />
    27 +            <category android:name="android.intent.category.DEFAULT" />
    28 +            <category android:name="android.intent.category.BROWSABLE" />
    29 +            <data android:scheme="ton" />
    30 +            <data android:scheme="ton-subscriptions" />
    31 +            <data android:scheme="tonkeeper" />
    32 +            <data android:scheme="tonsafe" />
    33 +            <data android:scheme="tc" />
    34 +        </intent-filter>
    35 +        <intent-filter android:autoVerify="true">
    36 +            <action android:name="android.intent.action.VIEW" />
    37 +            <category android:name="android.intent.category.LAUNCHER" />
    38 +            <category android:name="android.intent.category.DEFAULT" />
    39 +            <category android:name="android.intent.category.BROWSABLE" />
    40 +            <data android:scheme="https" />
    41 +            <data android:host="app.tonkeeper.com" />
    42 +            <data android:host="app.tonsafe.net" />
    43 +        </intent-filter>
    44 +    </activity>
    45 +    <activity android:name="com.facebook.react.devsupport.DevSettingsActivity" android:exported="false" />
    46 +
</application>
    47 </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
24   24
25   25 # Automatically convert third-party libraries to use AndroidX
26   26 android.enableJetifier=true
27 + 27 +
28   28 # Version of Flipper SDK to use with React Native
29   29 FLIPPER_VERSION=0.125.0
30   30
```

⊕

▼ 9 ██████████ app.json []

...

```
@@ -3,7 +3,7 @@
  3   3     "name": "TonSafe Wallet",
  4   4     "slug": "tonsafe-wallet",
  5   5     "privacy": "unlisted",
  6   6     "version": "2.1",
  7   7     "version": "2.2.1",
  8   8     "orientation": "portrait",
  9   9     "icon": "./assets/tonbase_splash_screen.png",
  10 10     "splash": {
  11 11       "@@ -31,6 +31,11 @@
  31 31       "web": {
  32 32         "favicon": "./assets/favicon.png"
  33 33       },
  34 34     "jsEngine": "hermes"
  35 35     "jsEngine": "hermes",
  36 36     "extra": {
  37 37       "ear": {
  38 38         "projectId": "3829cab9-18f4-4f80-a1ea-5bc507f925d9"
  39 39       }
  40 40     }
  41 41   }
```

⊕

▼ 42 ██████████ app/Navigation.tsx []

...

```
@@ -1,6 +1,6 @@
  1   1 import * as React from 'react';
  2   2 import {createNativeStackNavigator} from '@react-navigation/native-stack';
  3   3 - import {Platform, View, Image} from 'react-native';
  4   4 + import {View, Image} from 'react-native';
  5   5 import {WelcomeFragment} from './fragments/onboarding/WelcomeFragment';
  6   6 import {WalletImportFragment} from './fragments/onboarding/WalletImportFragment';
  7   7 import {WalletCreateFragment} from './fragments/onboarding/WalletCreateFragment';
  8   8
  9   9     "@@ -33,17 +33,12 @@ import {registerForPushNotificationsAsync, registerPushToken} from './utils/regi
 33 33 import * as Notifications from 'expo-notifications';
 34 34 import {PermissionStatus} from 'expo-modules-core';
 35 35 import {t} from './i18n/t';
 36 36 - import {AuthenticatorFragment} from './fragments/secure/AuthenticatorFragment';
 37 37 import {ConnectionsFragment} from './fragments/connections/ConnectionsFragment';
 38 38 import axios from 'axios';
 39 39 // import {NeocryptoFragment} from './fragments/integrations/NeocryptoFragment';
 40 40 // import {StakingTransferFragment} from './fragments/staking/StakingTransferFragment';
 41 41 // import {StakingFragment} from './fragments/staking/StakingFragment';
 42 42 import {SignFragment} from './fragments/secure/SignFragment';
 43 43 import {TransferFragment} from './fragments/secure/TransferFragment';
 44 44 import {createEngine} from './engine/createEngine';
 45 45 import {useRecallCallback} from 'recoil';
 46 46 - // import {Appfragment} from './fragments/apps/AppFragment';
 47 47 import {DevStorageFragment} from './fragments/dev/DevStorageFragment';
 48 48 import {getCurrencyPrices, setDefaultCurrency, saveTONPriceAtStart} from './utils/currency';
 49 49 import {LogoutFragment} from './fragments/logout/logout';
 50 50
 51 51     "@@ -67,8 +62,6 @@ import {SMSFragment} from './fragments/sms/SMSFragment';
 52 52
 53 53 const Stack = createNativeStackNavigator();
 54 54
 55 55 - // const Stack = Platform.OS === 'ios' ? createNativeStackNavigator() : createStackNavigator();
 56 56
 57 57 function fullScreen(name: string, component: React.ComponentType<any>) {
 58 58   return (
 59 59     <Stack.Screen
 60 60       @@@ -91,17 +84,6 @@ function genericScreen(name: string, component: React.ComponentType<any>) {
 61 61       );
 62 62     }
 63 63   }
 64 64
 65 65   function formSheetScreen(name: string, component: React.ComponentType<any>) {
 66 66     return (
 67 67       <Stack.Screen
 68 68       @@@ -94,17 +84,6 @@ function formSheetScreen(name: string, component: React.ComponentType<any>) {
 69 69       );
 70 70     }
 71 71   }
 72 72
 73 73   function fullSheetScreen(name: string, component: React.ComponentType<any>) {
 74 74     return (
 75 75       <Stack.Screen
 76 76       @@@ -95,17 +84,6 @@ function fullSheetScreen(name: string, component: React.ComponentType<any>) {
 77 77       );
 78 78     }
 79 79   }
 80 80
 81 81   function fullFormSheetScreen(name: string, component: React.ComponentType<any>) {
 82 82     return (
 83 83       <Stack.Screen
 84 84       @@@ -96,17 +84,6 @@ function fullFormSheetScreen(name: string, component: React.ComponentType<any>) {
 85 85       );
 86 86     }
 87 87   }
 88 88
 89 89   function fullFormSheetSheetScreen(name: string, component: React.ComponentType<any>) {
 90 90     return (
 91 91       <Stack.Screen
 92 92       @@@ -97,17 +84,6 @@ function fullFormSheetSheetScreen(name: string, component: React.ComponentType<any>) {
 93 93       );
 94 94     }
 95 95   }
 96 96 }
```

```

97      -           key={'formSheetScreen-${name}'}
98      -           name={name}
99      -           component={component}
100     -           options={{headerShown: false}}
101    -         />
102    -       );
103   -     }
104   -
105  87   function modalsScreen(name: string, component: React.ComponentType<any>) {
106  88     return (
107  89       <Stack.Screen
108  90         @@ -124,17 +106,6 @@
109  91         );
110  92       }
111  93     )
112  94   -
113  95   const navigation = [
114  96     fullScreen('Welcome', WelcomeFragment),
115  97     fullScreen('Home', HomeFragment),
116  98   -
117  99     @@ -178,7 +141,6 @@
118 100   const navigation = [
119 101     modalScreen('CreateInvoice', CreateInvoiceFragment),
120 102     modalScreen('CreateInvoiceLink', CreateInvoiceLinkFragment),
121 103     modalScreen('Transaction', TransactionPreviewFragment),
122 104   -
123 105     modalScreen('Authenticate', AuthenticateFragment),
124 106     modalScreen('Sign', SignFragment),
125 107     modalScreen('Migration', MigrationFragment),
126 108     modalScreen('SpamFilter', SpamfilterFragment),
127 109   -
128 110     @@ -181,17 +151,11 @@
129 111     const navigation = [
130 112       lockedModalScreen('Scanner', ScannerFragment),
131 113       genericScreen('DeveloperTools', DeveloperToolsFragment),
132 114       genericScreen('DeveloperToolsStorage', DevStorageFragment),
133 115   -
134 116     // lockedModalScreen('Buy', Neocryptofragment),
135 117     // fullScreen('Staking', StakingFragment),
136 118     // modalScreen('StakingTransfer', StakingTransferFragment),
137 119   -
138 120     // modalScreen('App', Appfragment),
139 121   ];
140 122   -
141 123   export const Navigation = React.memo(() => {
142 124     const safeArea = useSafeAreaInsets();
143 125   -
144 126     const recoilUpdater = useRecoilCallback<[any, any], any>(({set}) => (node, value) => set(node, value));
145 127   -
146 128     const engine = React.useMemo(() => {
147 129       let state = getAppState();
148 130       if (0 << state.selected && state.selected < state.addresses.length) {
149 131       -
150 132         @@ -212,8 +176,8 @@
151 133         export const Navigation = React.memo(() => {
152 134           const initial = React.useMemo(() => {
153 135             const onboarding = resolveOnboarding(engine);
154 136             setDefaultCurrency(); // If no currency is found then set a default currency.
155 137             getCurrenciesPrices(); // Get currency prices for all supported currencies.
156 138             saveTONPriceAtStart(); // LOCK TON-USD price for static conversions.
157 139   -
158 140             getCurrencyPrices().then(); // Get currency prices for all supported currencies.
159 141             saveTONPriceAtStart().then(); // LOCK TON-USD price for static conversions.
160 142             setSecretReminderCounter(); // Secret reminder setup
161 143             if (onboarding === 'backup') {
162 144               return 'WalletCreated';
163 145             }
164 146           -
165 147           ...
166 148         );
167 149       -
168 150       ...
169 151     );
170 152   -
171 153   ...

```

```

  ...
  @@ -2,28 +2,23 @@
  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
  height={42}
  width={42}
  -
  src={app.image?.preview256}
  blurhash={app.image.blurhash}
  -
  src={iconUrl}
  blurhash={iconUrl}
  -
  style={{marginRight: 10}}
  borderRadius={8}
  -
  />
  ...
  
```

```

  ...
  @@ -15,7 +15,7 @@
  export const ItemButton = React.memo((props: {
  -
  <Pressable style={({ opacity: iProps.pressed }) => (opacity: iProps.pressed ? 0.3 : 1, flexDirection: 'row', alignItems: 'center', width: (props.width) ? props.width : '100%')}> onPress={props.onPress}
  -
  <View style={{height: 48, paddingLeft: props.leftIcon ? 8 : 16, paddingRight: 16, alignItems: 'center', justifyContent: 'center', flexDirection: 'row', flexGrow: 1, flexBasis: 0 }}>
  -
  <View style={{flexGrow: 1, flexShrink: 1, flexDirection: 'row', alignItems: 'center' }}>
  -
  
```

```
18 -         {props.leftIcon && ()}
18 +         {props.leftIcon && ()}
19 19         <Text
20 20             style={{
21 21                 fontSize: 17,
```

```
...+ 24 app/fragments/MoreSettingsFragment.tsx ...
176 176     @@ -176,18 +176,18 @@ export const MoreSettingsFragment = fragment(() => {
177 177         </View>
178 178         <View style={{
179 179             marginHorizontal: 10, marginTop: 5,
180 180             backgroundColor: Theme.item,
181 181             borderRadius: Theme.borderRadius,
182 182             justifyContent: 'center',
183 183             alignItems: 'center',
184 184             flexShrink: 1,
185 185         }}>
186 186         <View style={{ marginHorizontal: 16, width: '100%' }}>
187 187             <ItemButton leftIcon={require('../assets/extensions.png')} title={t('home.extensions')} onPress={() =>
188 188                 navigation.navigate('Extensions')}>
189 189             </View>
190 190             <View style={{
191 191                 marginHorizontal: 16, width: '100%' }}>
192 192                 <ItemButton leftIcon={require('../assets/extensions.png')} title={t('home.extensions')} onPress={() =>
193 193                     navigation.navigate('Extensions')}>
194 194             </View>
195 195         </View>
196 196         <View style={{
197 197             marginHorizontal: 10, marginTop: 5,
```

```
...+ 11 app/fragments/PrimeHomeFragment.tsx ...
19 19     @@ -19,12 +19,13 @@ import BN from 'bn.js';
20 20     import {InformationFragment} from './information/InformationFragment';
21 21     import {getDarkModeValue} from './utils/misc';
22 +     import {SecretRemindersFragment} from './secret-reminders/SecretRemindersFragment';
23 +     import ExternalAppConnection from './tonconnect/connection-view/ExternalAppConnection';
23 +     import {TonConnectRemoteBridge} from './tonconnect';

22 24     export const PrimeHomeFragment = fragment(() => {
23 25         const safeArea = useSafeAreaInsets();
24 26         const [tab, setTab] = React.useState(0);
25 27         const [infoTab, infoTabRefresher] = React.useState(0);
26 28         const [scrollToTop, setScrollToTop] = React.useState(0);
27 29         const navigation = useTypedNavigation();
28 30         const loader = useGlobalLoader();
29 31         const engine = useEngine();

...+ 9 @@ -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();
116 117         navigation.navigate('Authenticate', {
117 118             session: resolved.session,
118 119             endpoint: resolved.endpoint
119 120         });
120 121         TonConnectRemoteBridge.handleConnectDeeplink(resolved.query).then();
121 122     });
122 123 }

...+ 6 @@ -126,6 +124,7 @@ export const PrimeHomeFragment = fragment(() => {
126 124     return (
127 125         <View style={{flexGrow: 1}}>
128 126             <SecretRemindersFragment/><SecretRemindersFragment>
127 +             <ExternalAppConnection/><ExternalAppConnection>
128 128             <View style={{flexGrow: 1}}>
129 129                 <StatusBar style={{getDarkModeValue() ? 'light' : 'dark'}}/>
130 130                 <View style={{position: 'absolute', top: 0, left: 0, right: 0, bottom: 0, opacity: tab === 0 ? 1 : 0}} pointerEvents={tab === 0 ? 'box-none' : 'none'}>
131 131             </View>
132 133         </View>
```

```
...+ 64 app/fragments/connections/ConnectionsFragment.tsx ...
1 1     @@ -1,4 +1,3 @@
1 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';
...+ 64 @@ -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 { getConnectedAppReference, getDarkModeValue } from '../../../../utils/misc';
14 +     import { IConnectedApp, IConnectedAppConnection } from '../../../../store';
15 +     import { TonConnect } from '../../../../tonconnect';
16 +
17 +     type connectedAppType = { connectedApp: IConnectedApp | null; connection: IConnectedAppConnection | null; }[];
```

```
22 +     const [connectedApps, setConnectedApps] = React.useState<ConnectedAppsType>(getConnectedAppsReference());
23
24 -     Alert.alert(t('auth.revoke.title'), t('auth.revoke.message'), [
25 -         {
26 -             text: t('common.cancel')
27 -         },
28 -         {
29 -             text: t('auth.revoke.action'), style: 'destructive', onPress: () => {
30 -                 addPendingRevoke(src);
31 -                 removeConnectionReference(src);
32 -                 setApps((a) => a.filter((v) => v.key !== src));
33 -                 packoff('revoke', async () => {
34 -                     await axios.post(`https://connect.tonhubapi.com/connect/revoke`, { key: src }, { timeout: 5000 });
35 -                     removePendingRevoke(src);
36 -                 });
37 -             }
38 -         }
39 -     ]);
40
41 -     let disconnectApp = React.useCallback((url: string | undefined) => {
42 -         if (!url) {
43 -             Alert.alert(t('auth.revoke.title'), t('auth.revoke.message'), [
44 -                 {
45 -                     text: t('common.cancel')
46 -                 },
47 -                 {
48 -                     text: t('auth.revoke.action'), style: 'destructive', onPress: async () => {
49 -                         await TonConnect.disconnect(url);
50 -                         setConnectedApps(getConnectedAppsReference());
51 -                     }
52 -                 }
53 -             ]
54 -         );
55 -     };
56
57 -     return (
58 -         <View style={{
59 -             flexGrow: 1, flexBasis: 0,
60 -             @@@ -70,7 +62,8 @@ export const ConnectionsFragment = fragment(() => {
61 -             <Text>{ t('auth.apps.title')}</Text>
62 -         </View>
63 -     })
64 -     <Text style={{ fontSize: 24, textAlign: 'center', color: Theme.textSecondary }}>{t('auth.noApps')}
65 -     <Text style={{ fontSize: 24, textAlign: 'center', color: Theme.textSecondary }}>{t('auth.noApps')} + 'n'</Text>
66 -     <Text style={{ fontSize: 14, textAlign: 'center', color: Theme.textSecondary }}>{t('applications.infoMessage')}

Load diff



This file was deleted.


```

```
  @@ -13,6 +13,8 @@ import { t } from '../../../../../i18n/t';
13 13   import { SystemFragment } from '../../../../../systemFragment';
14 14   import { RoundButton } from '../../../../../components/RoundButton';
15 15   import { Theme } from '../../../../../Theme';
16 + import { IonConnectRemoteBridge } from '../../../../../ionconnect';
17 + import queryString from 'query-string';
18
19  export const ScannerFragment = systemFragment(() => {
20     const safeArea = useSafeAreaInsets();
21
22  @@ -38,13 +40,23 @@ export const ScannerFragment = systemFragment(() => {
23
24    useEffect(() => {
25      if (barcodes && barcodes.length > 0 && barcodes[0]) {
26        console.log(barcodes, route);
27        if (route && (route as any).callback) {
28          setActive(false);
29
30        setTimeout(() => {
31          navigation.goBack();
32
33          if (barcodes[0].content.type === BarcodeValueType.URL) {
34            (route as any).callback(barcodes[0].content.data.url);
35
36          if ((barcodes[0].content.type === BarcodeValueType.TEXT) || ((barcodes[0].content.type === BarcodeValueType.TEXT) && (barcodes[0].content.data.includes('tc://')))) {
37            let endpoint: string | undefined;
38            if (barcodes[0].content.type === BarcodeValueType.URL) {
39              endpoint = barcodes[0].content.data.url;
40            } else {
41              endpoint = barcodes[0].content.data;
42            }
43            if (endpoint) {
44              const query: any = queryString.parseUrl(endpoint).query;
45              IonConnectRemoteBridge.handleConnectDeepLink(query).then();
46            }
47
48        } else if (barcodes[0].content.type === BarcodeValueType.TEXT) {
49          (route as any).callback(barcodes[0].content.data);
50
51      } else if (barcodes[0].content.type === BarcodeValueType.UNKNOWN) {
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
```

```
  @@ -25,13 +25,15 @@ import { LoadingIndicator } from '../../../../../components>LoadingIndicator';
25 25   import { log } from '../../../../../utils/log';
26 26   import { Engine, useEngine } from '../../../../../engine/Engine';
27 27   import { WalletState } from '../../../../../engine/products/WalletProduct';
28 + import { useCallback, useState } from 'react';
29 + import { useCallback, useEffect, useState } from 'react';
30   import Clipboard from 'react-native-clipboard/clipboard';
31   import * as Haptics from 'expo-haptics';
32   import * as Animatable from 'react-native-animatable';
33   import { AppPriceHistoryGraphFragment } from '../../../../../graph/AppPriceHistoryGraphFragment';
34 + import { AppMessageFragment } from '../../../../../app-message/AppMessageFragment';
35 + import { VerifyTransactionModal } from '../../../../../ionconnect/transactions/VerifyTransactionModal';
36 + import { setWalletStateInit } from '../../../../../utils/misc';
37
38  const WalletTransactions = React.memo(props: {
39    txs: { id: string, time: number }[]
40  }) => {
41
42  @@ -102,6 +104,9 @@ function WalletComponent(props: { wallet: WalletState }) {
43
44    const engine = useEngine();
45    const syncState = engine.state.use();
46    const account = props.wallet;
47
48    useEffect(() => {
49      setWalletStateInit().then(); // Set wallet state init.
50    }, []);
51
52    //
53    // Transactions
54    //
55
56  @@ -244,13 +249,7 @@ function WalletComponent(props: { wallet: WalletState }) {
57
58    const engine = useEngine();
59    const syncState = engine.state.use();
60    const account = props.wallet;
61
62    useEffect(() => {
63      if (res && res.type === 'connect') {
64        // if QR is valid navigate to sign fragment
65        navigation.navigate('Authenticate', {
66          session: res.session,
67          endpoint: res.endpoint
68        });
69      }
70
71    }, []);
72
73  @@ -315,6 +314,7 @@ function WalletComponent(props: { wallet: WalletState }) {
74
75    return (
76      <View style={{flexGrow: 1, paddingBottom: safeArea.bottom}}>
77        <VerifyTransactionModal>/>{VerifyTransactionModal}
78
79        <AnimatedScrollView
80          contentContainerStyle={[
81            flexGrow: 1,
82
83          @@ -545,7 +545,7 @@ function WalletComponent(props: { wallet: WalletState }) {
84
85          <View>
86            <View style={{flexGrow: 1, FlexBasis: 0, marginRight: 8, backgroundColor: Theme.item, borderRadius: Theme.borderRadius}}>
87              <TouchableOpacity highlightUnderlineColor={Theme.selector} style={{borderRadius: Theme.borderRadius}} onPress={() => {
88                navigation.navigate('Information', {fullHeight: true});
89
90                navigation.navigate('Connections')
91              }}>
92                <View style={{
93                  justifyContent: 'center',
94
95                  @@ -557,8 +557,8 @@ function WalletComponent(props: { wallet: WalletState }) {
96
97                  flexWrap: 'wrap',
98                  paddingTop: 6
99
100                >>
101
102                <Image source={require '../../../../../assets/homepage-tonbase-icons.png')} style={{width: 20, height: 20, marginRight: 7, tintColor: Theme.tintColor}}></Image>
103
104                <Text style={{fontSize: 14, color: Theme.accentText, textAlignVertical: 'center'}}>{('common.infoTabTitle')}
```

```
562 562             <View>
563 563             </TouchableHighlight>
564 564         </View>
565 565     </View>
566 566     <View style={t('products.productsSection')}>
567 567     <Text style={t('products.productsSectionTitle')}>{t('products.productsSectionTitle')}
```

```
78      62          {jettons.map((jt) => (
79      63            @@ -83,17 +67,6 @@ export const ProductsComponent = React.memo(() => {
80              64              engine{engine}
81            /> : <></>
82          69        )));
83        64        /* name=({products.oldWallets.title})*/
84        65        subtitle(t("products.oldWallets.subtitle"))
85        66        icon={OldWalletIcon}
86        67        value={oldWalletsBalance}
87        68        onPress={() => navigation.navigate('Migration')}
88        69        style={marginVertical: 4}
89        70        />})
90        71      });
91      72    })
92    )
93  )
94  )
95  )
96  )
97  )
98  )
99  )
```

@@ -162,6 +162,8 @@ logout: {

162 162 },
163 163 applications: {
164 164 noAppTxt: 'توفّي تطبيقات مصلحة',
165 + infoMessage: 'تم توصيل التطبيق، ارجو تجربة TON Connect',
166 + cannotConnectMessage: 'أثنى وظائف لا يمكن إنشاء الصالح لأن TON Connect يستخدم موقع الويب الجيد إصدار',
167 167 },
168 168 infoTabTitles: {
169 169 title: 'TON',

@@ -278,6 +280,7 @@ logout: {

278 280 expired: 'تمت ملائمة طلب المساعدة من يطلب',
279 281 failed: 'Invalid URL',
280 282 completed: 'تمكّن طلب المساعدة من يطلب',
283 + rejected: 'رفض طلب التفريغ',
284 284 authorized: 'تمكّن طلب الوصول',
285 285 authRequired: 'المساعدة مطلوبة',
286 286 disableAppAuth: 'عد الدخول، تحقق هووية',

@@ -162,6 +162,8 @@ const schema: PrepareSchema<LocalizationSchema, '' | '_plural'> = {

162 162 },
163 163 applications: {
164 164 noAppTxt: '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',
284 284 authorized: 'Authorization request approved',
285 285 authRequired: 'Authentication is required',
286 286 disableAppAuth: 'Startup security check',

@@ -162,6 +162,8 @@ const schema: PrepareSchema<LocalizationSchema, '' | '_plural'> = {

162 162 },
163 163 applications: {
164 164 noAppTxt: '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',
284 284 authorized: 'Solicitud de autorización aprobada',
285 285 authRequired: 'Se requiere autenticación',
286 286 disableAppAuth: 'En el inicio identidad',

@@ -162,6 +162,8 @@ const schema: PrepareSchema<LocalizationSchema, '_0' | '_1' | '_2'> = {

162 162 },
163 163 applications: {
164 164 noAppTxt: 'Нет подключенных приложений.',
165 + infoMessage: 'При подключении dApps выберите TON Connect.',
166 + cannotConnectMessage: 'Удаленный веб-айпорт 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: 'Невероятная ссылка',
280 282 completed: 'Этот запрос на авторизацию уже подтвержден',
283 + rejected: 'Запрос на авторизацию отклонен',
284 284 authorized: 'Запрос на авторизацию одобрен',
285 285 authRequired: 'Требуется аутентификация',
286 286 disableAppAuth: 'Народ при входе',

@@ -162,6 +162,8 @@ const schema: PrepareSchema<LocalizationSchema, '' | '_plural'> = {

162 162 },
163 163 applications: {
164 164 noAppTxt: 'ບໍລິສັດທະນາລົມເອົາໃຫຍ່',
165 + infoMessage: 'ເປົ້າລືວ່າມີລົມເອົາໃຫຍ່, ປຸກລົມເອົາ TON Connect.',
166 + cannotConnectMessage: '(ບໍລິສັດທະນາລົມເອົາໃຫຍ່ TON Connect ຂັ້ນມີການຕົ້ນລົມເອົາໃຫຍ່ ຫຼື ລົມເອົາໃຫຍ່)',
167 167 },
168 168 infoTabTitles: {
169 169 title: 'TON',

```

  278 288     expired: 'ព័ត៌មានការទិន្នន័យត្រូវបានផ្តល់ជូនតាមរយៈ',
  279 281     failed: 'Invalid URL',
  280 282     completed: 'ព័ត៌មានត្រូវបានទទួលខ្លួនដោយការទិន្នន័យ',
  281 283 +     rejected: 'ត្រូវបានសង្ខេបដោយការទិន្នន័យ',
  282 285     authorized: 'ត្រូវបានសង្ខេបដោយការទិន្នន័យ',
  283 286     authRequired: 'ត្រូវបានសង្ខេបដោយការទិន្នន័យ',
  284 287     disableAppAuth: 'បែងចែកគ្រាមគ្រាប់គ្រាប់ការទិន្នន័យ'

  ↴ 3 █ app/i18n/schemas.ts □
    @@ -94,6 +94,8 @@ export type LocalizationSchema = {
  94 94     },
  95 95     applications: {
  96 96         noAppssTxt: string,
  97 97 +         infoMessage: string,
  98 98 +         cannotConnectMessage: string
  99 99     },
 100 100     settings: {
 101 101         title: string,
 102 102     }
 103 103
 104 104     @@ -217,6 +219,7 @@ export type LocalizationSchema = {
 217 219     expired: string,
 218 220     failed: string,
 219 221     completed: string,
 220 222 +     rejected: string,
 221 223     noApps: string,
 222 225     name: string,
 223 226     yourWallet: string,
 224 227 }

  ↴ 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 =
  'eyJhbGciOiJIUzI1NiJ9.eyJ0eXAiOiJKV1QiLCJ4YWNvbmF0dXJuIjoiQ29keG1yNDc1XSwiZXhwIjox00Mx00c0015LClpc3Mi01JAd69uYX8pX2JvdCIsImp0aS16Ik0zNjZSUlRVVhQQTVKSuNQVzRmU8FY
  6 +     UyIsInhjb3ZlrfiicAyydmYiwiw3ViijoidG9wYXplbmh0cDpmb1I.C_ZswZD_6bewTHfAkqjsiPjV9gsR8UtxAOAi-egEFxpRa4c3Dpsbtou_US08wpNqhlXzTAqan9maLbDw';

  7 + export const tonapiIOEndpoint = 'https://tonapi.io';
  8 + const getBulkInfo = async (addresses: string[]) => {
  9 +     const endpoint = getServerConfig('tonapiIOEndpoint');

 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 +
 86 +
 87 + type Balances = {
 88 +     balance: number;

```

```
89 +     version: string;
90 +   };
91 +
92 +   async function resolveDns(domain: string) {
93 +     try {
94 +       const endpoint = getServerConfig('tonapiIOEndpoint');
95 +       const response: any = await axios.get(`${endpoint}/v1/dns/resolve`, {
96 +         headers: {
97 +           Authorization: `Bearer ${getServerConfig('tonApiKey')}`,
98 +         },
99 +         params: {
100 +           name: domain,
101 +         },
102 +       });
103 +       return response.data;
104 +     } catch (e) {
105 +       return false;
106 +     }
107 +   }
108 +
109 +   async function getBalances(pubkey: string) {
110 +     const wallets = await findByPubkey(pubkey);
111 +
112 +     const balances: Balances[] = [];
113 +     for (let wallet of wallets) {
114 +       const versions = ['wallet_v3R1', 'wallet_v3R2', 'wallet_v4R1', 'wallet_v4R2'];
115 +       const detectedVersion = wallet.interfaces.find((version: any) =>
116 +         versions.includes(version),
117 +       );
118 +       if (detectedVersion) {
119 +         if (wallet.balance > 0) {
120 +           const version = detectedVersion.replace('wallet_', '');
121 +           balances.push({
122 +             balance: wallet.balance,
123 +             version,
124 +           });
125 +         }
126 +       }
127 +     }
128 +
129 +     return balances;
130 +   }
131 +
132 +   export const Tonapi = {
133 +     getBulkInfo,
134 +     findByPubkey,
135 +     getSeqno,
136 +     getWalletInfo,
137 +     getBalances,
138 +     resolveDns,
139 +   };

```

```
✓ 1 ━━━━━━ app/libs/Tonapi/index.ts [ ]
...
... @@ -0,0 +1 @@
1 + export * from './Tonapi';
(
```

```
✓ 4 ━━━━━━ app/libs/deeplinking/DeepLinkingContext.ts [ ]
...
... @@ -0,0 +1,4 @@
1 + import React from 'react';
2 + import { DeepLinkingContextValue } from './deeplinking.types';
3 +
4 + export const DeepLinkingContext = React.createContext(null);
(
```

```
✓ 124 ━━━━━━ app/libs/deeplinking/DeepLinkingProvider.tsx [ ]
...
... @@ -0,0 +1,124 @@
1 + import React from 'react';
2 + import {
3 +   DeepLinkingContextValue,
4 +   DeepLinkingResolveOptions,
5 +   DeepLinkingResolver,
6 +   DeepLinkOrigin,
7 + } from './deeplinking-types';
8 + import { DeepLinkingContext } from './DeepLinkingContext';
9 + import { useDeepLinkingListener } from './hooks/useDeepLinkingListener';
10 + import { applyMiddleware, matchPath, Middleware } from './utils';
11 +
12 + interface DeepLinkingProviderProps {
13 +   useIsReadyToListen?: () => boolean;
14 +   useResolvers?: () => void;
15 + }
16 +
17 + export async function delay(ms: number) {
18 +   return new Promise((resolve) => setTimeout(resolve, ms));
19 + }
20 +
21 + export const DeepLinkingProvider: React.FC<DeepLinkingProviderProps> = (props) => {
22 +   const resolvers = React.useRef<Map<string, DeepLinkingResolver>>(<new Map()>.current);
23 +   const middleware = React.useRef<Middleware | undefined>();
24 +   const prefixes = React.useRef<string[]>([]);
25 +
26 +   const add = (pathname: string, resolver: DeepLinkingResolver) => {
27 +     if (!resolvers.current.has(pathname)) {
28 +       resolvers.current.set(pathname, resolver);
29 +     }
30 +   };
31 +
32 +   const getResolver = (url: string, options?: DeepLinkingResolveOptions) => {
33 +     try {
34 +       const prefix = prefixes.current.find((prefix) => {
35 +         return url.startsWith(prefix);
36 +       });
37 +
38 +       if (!prefix) {
39 +         return null;
40 +       }
41 +
42 +       let pathname = url.substring(prefix.length);
43 +       if (!pathname.startsWith('/')) {
44 +         pathname = `/${pathname}`;
45 +       }
46 +
47 +       for (let path of resolvers.current.keys()) {
48 +         const matchedPath = matchPath(path, pathname);
49 +         if (matchedPath) {
50 +           const resolver = resolvers.current.get(path);
51 +
52 +           return async () => {
53 +             if (options?.delay) {
54 +               await delay(options?.delay);
55 +             }
56 +           };
57 +         }
58 +       }
59 +     } catch (e) {
60 +       console.error(`Error getting resolver for ${url}: ${e.message}`);
61 +     }
62 +   };
63 +
64 +   const provider = {
65 +     add,
66 +     getResolver,
67 +     matchPath,
68 +     matchPrefix,
69 +     useResolvers,
70 +   };
71 +
72 +   return provider;
73 + }
74 +
75 + export default DeepLinkingProvider;
```

```

56 +
57 +     const middlewares: Middleware[] = [];
58 +     if (middleware.current) {
59 +         middlewares.push(middleware.current);
60 +     }
61 +
62 +     applyMiddleware(middlewares, () => {
63 +         return resolver({
64 +             origin: options?.origin || DeepLinkOrigin.DEEPLINK,
65 +             params: matchedPath.params,
66 +             // @ts-ignore
67 +             query: matchedPath.query,
68 +             resolveParams: options?.params ?? {}
69 +         });
70 +     });
71 + }
72 +
73 +
74 + ) catch (err) {
75 +     console.log(`[Deeplinking]: error parse`, url, err);
76 +     // debugLog(`[Deeplinking]: error parse`, url, err);
77 + }
78 +
79 + return null;
80 +};
81 +
82 + const resolve = (url: string, options?: DeeplinkingResolveOptions) => {
83 +     try {
84 +         const resolver = getResolver(url, options);
85 +         if (resolver) {
86 +             resolver();
87 +         }
88 +     } catch (err) {
89 +         console.log(`[Deeplinking]: error resolve`, url, err);
90 +     }
91 + };
92 +
93 + const addMiddleware = (fn: Middleware) => {
94 +     middleware.current = fn;
95 + };
96 +
97 + const setPrefixes = (arr: string[]) => {
98 +     prefixes.current = arr;
99 + };
100 +
101 + const value: DeepLinkingContextValue = {
102 +     addMiddleware,
103 +     setPrefixes,
104 +     getResolver,
105 +     resolve,
106 +     add,
107 + };
108 +
109 + return (
110 +     <DeepLinkingContext.Provider value={value}>
111 +         <DeepLinkingListener
112 +             useIsReadyToListen={props.useIsReadyToListen}
113 +             useResolvers={props.useResolvers}
114 +         />
115 +         {props.children}
116 +     </DeepLinkingContext.Provider>
117 + );
118 +
119 +
120 + const DeepLinkingListener = (props: DeepLinkingProviderProps) => {
121 +     useDeepLinkingListener(props);
122 +
123 +     return null;
124 + };

```

```

v 44 ████ app/libs/deeplinking/deeplinking.types.ts □ ...
...
... @@ -0,0 +1,44 @@
1 + import { Middleware } from './utils';
2 +
3 + export interface PathPattern {
4 +     path: Path;
5 +     caseSensitive?: boolean;
6 +     end?: boolean;
7 + }
8 +
9 + export type Params<Key extends string = string> = {
10 +     readonly [key in Key]: string;
11 + };
12 +
13 + export type DeeplinkingResolverOptions = {
14 +     origin: DeepLinkOrigin;
15 +     query: Params;
16 +     params: Params;
17 +     resolveParams: Record<string, any>;
18 + };
19 +
20 + export type DeeplinkingResolver = (
21 +     options: DeeplinkingResolverOptions,
22 + ) => Promise<void> | void;
23 +
24 + export type DeeplinkingResolveOptions = {
25 +     delay?: number;
26 +     origin?: DeepLinkOrigin;
27 +     params?: Record<string, any>;
28 + };
29 +
30 + export type DeepLinkingContextValue = {
31 +     getResolver: (
32 +         path: string,
33 +         options?: DeeplinkingResolveOptions,
34 +     ) => () => Promise<void> | null;
35 +     resolve: (path: string, options: DeeplinkingResolveOptions) => void;
36 +     add: (path: string, resolver: DeeplinkingResolver) => void;
37 +     setPrefixes: (prefixes: string[]) => void;
38 +     addMiddleware: (middleware: Middleware) => void;
39 + };
40 +
41 + export enum DeepLinkOrigin {
42 +     DEEPLINK = 'DEEPLINK',
43 +     QR_CODE = 'QR_CODE',
44 + }

```

```

v 12 ████ app/libs/deeplinking/hooks/useDeepLinking.ts □ ...
...
... @@ -0,0 +1,12 @@
1 + import React from 'react';
2 + import { DeepLinkingContext } from '../DeepLinkingContext';
3 +
4 + export const useDeepLinking = () => {

```

```
5 +     const deepLinking = React.useContext(DeepLinkingContext);
6 +
7 +     if (!deepLinking) {
8 +         throw new Error('No DeepLinkingProvider');
9 +     }
10 +
11 +     return deepLinking;
12 + }
```

...

```
1 + import { Linking } from 'react-native';
2 + import { useCallback, useEffect } from 'react';
3 + import { useDeepLinking } from '../hooks/useDeeplinking';
4 +
5 + const useIsReadyToListenDefault = () => true;
6 + const useResolversDefault = () => {};
7 +
8 + let isInitialURLProcessed = false;
9 +
10+ export function useDeepLinkingListener(
11+   useIsReadyToListen = useIsReadyToListenDefault,
12+   useResolvers = useResolversDefault
13+ ) {
14+   useResolvers();
15+
16+   const isReadyToListen = useIsReadyToListen();
17+   const deepLinking = useDeepLinking();
18+   deepLinking.setPrefixes([
19+     'ton://',
20+     'tonkeeper://',
21+     'https://app.tonkeeper.com',
22+     'https://tonhub.com',
23+   ]);
24+
25+   const handleUrl = useCallback(({ url }) => {
26+     Linking.canOpenURL(url).then((supported) => {
27+       if (supported) {
28+         deepLinking.resolve(url);
29+       }
30+     });
31+   }, []);
32+
33+   useEffect(() => {
34+     if (!isReadyToListen) {
35+       return;
36+     }
37+
38+     const linkingListener = Linking.addEventListener('url', handleUrl);
39+     if (!isInitialURLProcessed) {
40+       isInitialURLProcessed = true;
41+       Linking.getInitialURL()
42+         .then((url) => {
43+           if (url) {
44+             deepLinking.resolve(url);
45+           }
46+         })
47+         .catch((err) => console.error('An error occurred', err));
48+     }
49+
50+     return () => {
51+       linkingListener.remove();
52+     };
53+   }, [isReadyToListen]);
54+ }
```

...

```
1 + export { DeepLinkingProvider } from './DeepLinkingProvider';
2 + export { useDeepLinking } from '../hooks/useDeeplinking';
3 + export * from './deepLinking.types';
4 
```

...

```
1 + export type MaybePromise<T> = T | Promise<T>;
2 +
3 + export type Middleware = (next: () => void) => MaybePromise<any>;
4 +
5 + export function applyMiddleware(middlewares: Middleware[], last: () => any) {
6+ // @ts-ignore
7+   async function dispatch(i: number) {
8+     let fn = middlewares[i];
9+     if (i === middlewares.length) {
10+       return last();
11+     }
12+     return fn && fn(dispatch.bind(null, i + 1));
13+   }
14+
15+   return dispatch(0);
16+ }
```

...

```
1 + export function compilePath(
2+   path: string,
3+   caseSensitive = false,
4+   end = true
5+ ): [RegExp, string[]] {
6+   let paramNames: string[] = [];
7+   let regexpSource =
8+     `^`;
9+   path
10+     .replace(/\/*\?$/i, '')
11+     .replace(/\/*\//i, '/')
12+     .replace(/(\/*\?|(\/*))/g, "\$1");
13+     .replace(/:(\w+)/g, (_: string, paramName: string) => {
14+       paramNames.push(paramName);
15+       return `((\/*\?|)` + paramName + `)`;
16+     });
17+
18+   if (path.endsWith('*')) {
19+     paramNames.push('path');
20+     regexpSource += `((\/*|` + path + `))`;
21+   }
22+   else if (path === '*' || path === '/**') {
23+     regexpSource += `((\/*|` + path + `))`;
24+   }
25+   else {
26+     regexpSource += end
27+     ? `/*$`
```

```
27 +         : "(?:(:|=|---)|@[0-9A-F]{2})|\\b|\\/|$";
28 +
29 +
30 +     let matcher = new RegExp(regexpSource, caseSensitive ? undefined : "i");
31 +
32 +     return [matcher, paramNames];
33 + }
```

```
✓ 3 app/libs/deeplinking/utils/index.ts ...
...
...
1 + export * from './applyMiddleware';
2 + export * from './compilePath';
3 + export * from './matchPath';
```

```
✓ 61 app/libs/deeplinking/utils/matchPath.ts ...
...
...
1 + import queryParser from 'query-string';
2 + import { compilePath } from './compilePath';
3 + import { Params, PathPattern } from '../deeplinking.types';
4 +
5 + export function matchPath(
6 +   ParamKey extends string,
7 +   Path extends string
8 + ) {
9 +   pattern: PathPattern<Path> | Path,
10 +   url: string
11 + ) {
12 +   const parsedQuery = queryParser.parseUrl(url);
13 +   const query = parsedQuery.query;
14 +   const pathname = parsedQuery.url;
15 +
16 +   if (typeof pattern === "string") {
17 +     pattern = { path: pattern, caseSensitive: false, end: true };
18 +   }
19 +
20 +   let [matcher, paramNames] = compilePath(
21 +     pattern.path,
22 +     pattern.caseSensitive,
23 +     pattern.end
24 +   );
25 +
26 +   let match = pathname.match(matcher);
27 +   if (!match) return null;
28 +
29 +   let matchedPathname = match[0];
30 +   let pathnameBase = matchedPathname.replace(/(.)/+$/, "$1");
31 +   let captureGroups = match.slice(1);
32 +   let params: Params = paramNames.reduce(
33 +     (memo: any, paramName, index) => {
34 +       let query = '';
35 +       if (paramName === "path") {
36 +         const rawQuery = url.split('?')[1];
37 +         if (rawQuery) {
38 +           query = rawQuery ? `?${rawQuery}` : '';
39 +         }
40 +
41 +         const splatValue = captureGroups[index] ?? '';
42 +         pathnameBase = matchedPathname
43 +           .slice(0, matchedPathname.length - splatValue.length)
44 +           .replace(/(.)/+$/, "$1");
45 +       }
46 +
47 +       const paramValue = captureGroups[index];
48 +       memo[paramName] = `${paramValue}${query}`;
49 +     },
50 +     {}
51 +   );
52 + }
53 +
54 + return {
55 +   query,
56 +   params,
57 +   pathname: matchedPathname,
58 +   pathnameBase,
59 +   pattern,
60 + };
61 + }
```

```
✓ 42 app/libs/index.ts ...
...
...
1 + import { memo } from 'react';
2 + export const chunk = (input: any[], size: number) => {
3 +   return input.reduce((arr, item, idx) => {
4 +     return idx % size === 0
5 +       ? [...arr, [item]]
6 +       : [...arr.slice(0, -1), [...arr.slice(-1)[0], item]];
7 +   }, []);
8 + };
9 +
10 + export function shuffle(a: any[]) {
11 +   for (let i = a.length - 1; i > 0; i--) {
12 +     const j = Math.floor(Math.random()) * (i + 1);
13 +     [a[i], a[j]] = [a[j], a[i]];
14 +   }
15 +   return a;
16 + }
17 +
18 + export function getRandomInt(min: number, max: number) {
19 +   return Math.floor(Math.random() * (max - min)) + min;
20 + }
21 +
22 + export function sleep(timeout: number): Promise<void> {
23 +   return new Promise(resolve => {
24 +     setTimeout(() => resolve(), timeout);
25 +   });
26 + }
27 +
28 + export const getStackTrace = function () {
29 +   try {
30 +     const obj: any = {};
31 +     Error.captureStackTrace(obj, getStackTrace);
32 +     return obj.stack;
33 +   } catch (e) {
34 +     return e.stack;
35 +   }
36 + };
37 +
38 + export function lowerCaseFirstLetter(string: string) {
39 +   return string.charAt(0).toLowerCase() + string.slice(1);
40 + }
```

```
41 +     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 +     Cht = 'cht',
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: '0xdec17f958d2ee523a2206206994597c13d831ec7',
39 +         blockchain: 'ethereum',
40 +         decimals: 18,
41 +     },
42 +     [CryptoCurrencies.Wbtc]: {
43 +         address: '0x2260fac5e5542a73aa44fbcfedf7c193bc2c599',
44 +         blockchain: 'ethereum',
45 +         decimals: 8,
46 +     },
47 +     [CryptoCurrencies.Usdc]: {
48 +         address: '0x408b6991c6218b36c1d1904a2e9Eb0cE3606e848',
49 +         blockchain: 'ethereum',
50 +         decimals: 8,
51 +     },
52 +     [CryptoCurrencies.Dai]: {
53 +         address: '0xb0175474eb9094c44da98b954eedeac495271d0f',
54 +         blockchain: 'ethereum',
55 +         decimals: 8,
56 +     },
57 + },
58 +
59 + export const TokenConfigTestnet: { [index: string]: any } = {
60 +     [CryptoCurrencies.Usdt]: {
61 +         address: '0xdec856ae55b6e1a249f04cd3b947141bc146273c',
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: '0xdec856ae55b6e1a249f04cd3b947141bc146273c', // 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 +     tomkeeperEndpoint: string;
4 +     tomEndpoint: string;
5 +     tomEndpointAPIMKey: string;
6 +     tomApiEndpoint: string;
7 +     neocryptTomEndpoint: string;
8 +     supportLink: string;
9 +     isExchangeEnabled: string;
10 +    exchangePostUrl: string;
11 +    mercurySecret: string;
12 +    appsflyerDevKey: string;
13 +    appsflyerAppId: string;
14 +    tomNftsMarketplaceEndpoint: string;
15 +    tomapi0Endpoint: string;
16 +    tomapiMainnetHost: string;
17 +    tomapiTestnetHost: string;
18 +    tomApiKey: string;
19 +    cachedMediaEndpoint: string;
20 +    cachedMediaApiKey: string;
21 +    cachedMediaSsl: string;
22 +    NFTOnExplorerUrl: string;
23 +    flags: Record<string, boolean>;
24 + }
25 +
26 + let config: ServerConfig | null = null;
27 +
28 + export function setServerConfig(data: any, isTestnet: boolean) {
29 +     config = {

```

```

30 +     _version: 0,
31 +     tonkeeperEndpoint: data.tonkeeperEndpoint || 'https://api.tonkeeper.com',
32 +     tonEndpoint: data.tonEndpoint || 'https://toncenter.com/api/v2/jsonRPC',
33 +     tonEndpointAPIKey: data.tonEndpointAPIKey,
34 +     tonApiEndpoint: data.tonApiEndpoint || 'https://toncenter.com/api/v2',
35 +     neocryptoWebView: data.neocryptoWebView,
36 +     supportLink: data.supportLink || 'mailto:support@tonkeeper.com',
37 +     isExchangeEnabled: data.isExchangeEnabled,
38 +     exchangePostUrl: data.exchangePostUrl,
39 +     mercurySecret: data.mercurySecret,
40 +     appsflyerDevKey: data.appsflyerDevKey,
41 +     appsflyerAppId: data.appsflyerAppId,
42 +     tonNFTsMarketplaceEndpoint: data.tonNFTsMarketplaceEndpoint,
43 +     tonapiIOEndpoint: data.tonapiIOEndpoint || 'https://keeper.tonapi.io',
44 +     tonApiKey: data.tonApiKey,
45 +     tonapiMainnetHost: data.tonapiMainnetHost || 'https://tonapi.io',
46 +     tonapiTestnetHost: data.tonapiTestnetHost || 'https://testnet.tonapi.io',
47 +     cachedMediaEndpoint: data.cachedMediaEndpoint,
48 +     cachedMediaKey: data.cachedMediaKey,
49 +     cachedMediaSalt: data.cachedMediaSalt,
50 +     NFTOnExplorerUrl: data.NFTOnExplorerUrl || 'https://tonscan.org/nft/%s',
51 +     flags: data.flags || (),
52 +   };
53 + }
54 +
55 + export function updateServerConfig(jsonConfig: any) {
56 +   if (!jsonConfig) return;
57 +   try {
58 +     Object.entries(JSON.parse(jsonConfig)).map(([key, value]) => {
59 +       if (config) {
60 +         // @ts-ignore
61 +         config[key] = value;
62 +       }
63 +     });
64 +   } catch (e) {
65 +   }
66 + }
67 +
68 + export function isServerConfigLoaded() {
69 +   return !!config;
70 + }
71 +
72 + export function getServerConfig<T extends keyof ServerConfig>(key: T): ServerConfig[T] {
73 +   if (!config) {
74 +     throw new Error('Config is not loaded');
75 +   }
76 +
77 +   return config[key];
78 + }
79 +
80 + export function getServerConfigSafe<T extends keyof ServerConfig>(
81 +   key: T,
82 + ): ServerConfig[T] | 'none' {
83 +   return config ? config[key] : 'none';
84 + }

```

✓ 14 app/shared/dynamicConfig.ts

```

... ...
1 + import { CryptoCurrency, TokenConfig } from '../shared/constants';
2 +
3 + export function getTokenConfig(token: CryptoCurrency) {
4 +   const configs = TokenConfig;
5 +   return configs[token];
6 +
7 +
8 + export function getWalletName(): string {
9 +   return `${getChainName()}_default`;
10 +
11 +
12 + export function getChainName(): 'mainnet' | 'testnet' {
13 +   return 'mainnet';
14 +

```

✓ 5 app/store/index.ts

```

... ...
1 + import { createStore } from "zustand";
2 + export const store = createStore(() => {});
3 +
4 + // new store
5 + export * from './zustand';


```

✓ 179 app/store/models.ts

```

... ...
1 + export type TransactionType =
2 +   | 'receive'
3 +   | 'sent'
4 +   | 'buy'
5 +   | 'subscription'
6 +   | 'unsubscription';
7 +
8 + export interface Account {
9 +   address: string;
10 +  name?: string;
11 +  icon?: string;
12 +  isScam?: boolean;
13 + }
14 +
15 + export interface Jetton {
16 +   id: string;
17 +   name: string;
18 +   symbol?: string;
19 +   icon?: string;
20 +   divisibility?: number;
21 + }
22 +
23 + export interface TonTransfer {
24 +   amount: string; // nanocoins
25 +   recipient: Account; // address of the recipient
26 +   sender: Account; // address of the sender
27 +   payload?: string;
28 +   comment?: string;
29 + }
30 +
31 + export interface TonDeploy {
32 +   amount: string; // nanocoins
33 +   recipient: Account; // address of the recipient
34 +   sender: Account; // address of the sender
35 +   state_init: string; // recipient's StateInit
36 +   payload: string;
37 +   comment: string; // optional
38 + }
39 + 
```

```
40 + export interface NftItemTransfer {
41 +   sender: Account; // address of the sender
42 +   recipient: Account; // address of the recipient
43 +   payload: string;
44 +   comment: string; // utf8 string from payload
45 + }
46 +
47 + export interface JettonTransfer {
48 +   jetton: Jetton;
49 +   amount: string; // base units for this jetton
50 +   sender: Account; // address of the sender
51 +   recipient: Account; // address of the recipient
52 +   payloads: string;
53 +   comment: string; // utf8 string from payload
54 + }
55 +
56 + export interface ContractDeploy {
57 +   address: string;
58 +   deployer: Account; // address of the deployer
59 +   interfaces: string[];
60 + }
61 +
62 + export enum ActionType {
63 +   TokenTransfer = 'tokenTransfer',
64 +   JettonTransfer = 'jettonTransfer',
65 +   ContractDeploy = 'contractDeploy',
66 +   NftItemTransfer = 'nftItemTransfer',
67 +   Subscribe = 'subscribe',
68 +   UnSubscribe = 'unsubscribe',
69 +   Unknown = 'unknown',
70 +   AuctionBid = 'auctionBid',
71 + }
72 +
73 + export interface SubscriptionModel {
74 +   id?: string;
75 +   productName: string;
76 +   channelId: string;
77 +   amountNano: string;
78 +   intervalSec: number;
79 +   address: string;
80 +   status: string;
81 +   merchantName: string;
82 +   merchantPhoto: string;
83 +   returnUrl: string;
84 +   subscriptionId: number;
85 +   subscriptionAddress: string;
86 +   isActive?: boolean;
87 +   chargedAt: number;
88 +   fee: string;
89 +   userReturnUrl: string;
90 + }
91 +
92 + export interface InternalNotificationModel {
93 +   id: string;
94 +   title: string;
95 +   caption: string;
96 +   mode: 'warning';
97 +   action: {
98 +     type: 'open_link';
99 +     label: string;
100 +     url?: string;
101 +   };
102 +   isPersistenceHide?: boolean;
103 + }
104 +
105 + export interface ExchangeCategoryModel {
106 +   title: string;
107 +   subtitle: string;
108 +   items: string[];
109 + }
110 +
111 + export interface MarketplaceModel {
112 +   id: string;
113 +   title: string;
114 +   description: string;
115 +   icon_url: string;
116 +   marketplace_url: string;
117 + }
118 +
119 + export interface ExchangeMethodModel {
120 +   id: string;
121 +   title: string;
122 +   subtitle: string;
123 +   description: string;
124 +   disabled?: boolean;
125 +   badge: string;
126 +   badgeStyle: 'blue' | 'red';
127 +   features: [
128 +     {
129 +       text: string;
130 +       icon: string;
131 +     }
132 +   ];
133 +   icon_url: string;
134 +   action_button: {
135 +     title: string;
136 +     url: string;
137 +   };
138 +   info_buttons: [
139 +     {
140 +       title: string;
141 +       url: string;
142 +     }
143 +   ];
144 +   successUrlPattern?: {
145 +     pattern: string;
146 +     purchasedIndex: number;
147 +   };
148 + }
149 +
150 + export interface CollectionModel {
151 +   ownerAddressToDisplay?: string;
152 +   name: string;
153 +   description: string;
154 +   address: string;
155 +   addressRaw: string;
156 +   getGemsModerated?: boolean;
157 + }
158 +
159 + export interface TonDiamondMetadata {
160 +   animation_url: string;
161 +   image: string;
162 +   image_diamond: string;
163 +   lottie: string;
164 +   theme: {
165 +     main: string;
166 +   };
167 + }
168 + export interface JettonMetadata {
```

```
166 +     jettomAddress: string;
167 +     decimals: number;
168 +     symbol?: string;
169 +     image_data?: string;
170 +     image?: string;
171 +     description?: string;
172 +     name?: string;
173 +   }
174 +
175 + export interface FavoriteModel {
176 +   name: string;
177 +   address: string;
178 +   domain?: string;
179 + }
```

✓ 4 app/store/rootReducer.ts

```
... @@ -0,0 +1,4 @@
1 + // import { combineReducers } from '@reduxjs/toolkit';
2 + // export const rootReducer = combineReducers({});
3 + //
4 + // export type RootState = ReturnType<typeof rootReducer>;
```

✓ 1 app/store/rootSaga.ts

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

✓ 2 app/store/zustand/appslist/index.ts

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

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

```
... @@ -0,0 +1,15 @@
1 + export interface IAppMetadata {
2 +   name: string;
3 +   icon: string;
4 +   url: string;
5 + }
6 +
7 + export interface IAppsListStore {
8 +   fetching: boolean;
9 +   appslist: IAppMetadata[];
10 +   moreEnabled: boolean;
11 +   moreUrl: string;
12 +   actions: {
13 +     fetchPopularApps: () => void;
14 +   };
15 + }
```

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

```
... @@ -0,0 +1,52 @@
1 + import AsyncStorage from '@react-native-async-storage/async-storage';
2 + import axios from 'axios';
3 + import FastImage from 'react-native-fast-image';
4 + import create from 'zustand';
5 + import { persist } from 'zustand/middleware';
6 + import { IAppsListStore } from './types';
7 +
8 + const initialState: Omit<IAppsListStore, 'actions'> = {
9 +   fetching: true,
10 +   appslist: [],
11 +   moreEnabled: false,
12 +   moreUrl: '',
13 + };
14 +
15 + export const useAppsListStore = create(
16 +   persist<IAppsListStore>(
17 +     (set) => ({
18 +       ...initialState,
19 +       actions: {
20 +         fetchPopularApps: async () => {
21 +           set({ fetching: true });
22 +           try {
23 +             const response = await axios.get(
24 +               'https://api.tonkeeper.com/apps/popular',
25 +             );
26 +
27 +             const { apps, moreEnabled, moreUrl } = response.data.data;
28 +
29 +             FastImage.preload(
30 +               apps.map((app: any) => ({
31 +                 uri: app.icon,
32 +               })),
33 +             );
34 +
35 +             set({ appslist: apps, moreEnabled, moreUrl });
36 +           } catch {
37 +             } finally {
38 +               set({ fetching: false });
39 +             }
40 +           },
41 +         },
42 +       )),
43 +       {
44 +         name: 'appslist',
45 +         getStorage: () => AsyncStorage,
46 +         partialize: ({ appslist, moreEnabled }) => {
47 +           ({ appslist, moreEnabled } as IAppsListStore),
48 +         },
49 +       },
50 +     );
51 +
52 +     useAppsListStore.getState().actions.fetchPopularApps();
53 +   })
54 + );
```

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

```
... @@ -0,0 +1,73 @@
1 + import { getChainName } from '../../../../../shared/dynamicConfig';
2 + import { getConnectedAppByUrl } from './selectors';
3 + import { IConnectedAppConnection, IConnectedApp, TonConnectBridgeType } from './types';
4 + import { useConnectedAppStore } from './useConnectedAppStore';
5 + import { getCurrentAddress } from '../../../../../storage/appState';
6 + import { AppConfig } from '../../../../../AppConfig';
7 +
8 + export const saveAppConnection = (
9 +   walletAddress: string,
10 +   appData: Omit<IConnectedApp, 'connections'>,
11 +   connection: IConnectedAppConnection,
12 + ) => {
13 +   const chainName = getChainName();
```

```

13 +   .useConnectedAppStore
14 +   .getState()
15 +   .actions.saveAppConnection(getChainName(), walletAddress, appData, connection);
16 + };
17 +
18 + export const removeConnectedApp = (url: string) => {
19 +   const currentWalletAddress = getCurrentAddress().address.toFriendly({testOnly: AppConfig.isTestnet});
20 +
21 +   useConnectedAppStore
22 +   .getState()
23 +   .actions.removeApp(getChainName(), currentWalletAddress, url);
24 + };
25 +
26 + export const removeInjectedConnection = (url: string) => {
27 +   const currentWalletAddress = getCurrentAddress().address.toFriendly({testOnly: AppConfig.isTestnet});
28 +
29 +   useConnectedAppStore
30 +   .getState()
31 +   .actions.removeInjectedConnection(getChainName(), currentWalletAddress, url);
32 + };
33 +
34 + export const findConnectedAppByUrl = (url: string): IConnectedApp | null => {
35 +   const currentWalletAddress = getCurrentAddress().address.toFriendly({testOnly: AppConfig.isTestnet});
36 +
37 +   return getConnectedAppByUrl(
38 +     currentWalletAddress,
39 +     url,
40 +     useConnectedAppsStore.getState(),
41 +   );
42 + };
43 +
44 + export const findConnectedAppByClientSessionId = (
45 +   clientSessionId: string,
46 + ): { connectedApp: IConnectedApp | null; connection: IConnectedAppConnection | null } => {
47 +   const currentWalletAddress = getCurrentAddress().address.toFriendly({testOnly: AppConfig.isTestnet});
48 +
49 +   const connectedAppsList = Object.values(
50 +     useConnectedAppsStore.getState().connectedApps[getChainName()]
51 +     currentWalletAddress
52 +   ) || [];
53 + };
54 +
55 + let connection: IConnectedAppConnection | null = null;
56 +
57 + const connectedApp = connectedAppsList.find((app) =>
58 +   app.connections.find((item) => {
59 +     if (
60 +       item.type === TonConnectBridgeType.Remote &&
61 +       item.clientSessionId === clientSessionId
62 +     ) {
63 +       connection = item;
64 +
65 +       return true;
66 +     }
67 +
68 +     return false;
69 +   }));
70 +
71 + return { connectedApp: connectedApp ?? null, connection };
72 + };
73 + };

```

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

```

...
...
```

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

```

...
...
```

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

```

...
...
```

```

24 +     mainSettings;
25 +
26 +     icon: string;
27 +     autoConnectDisabled?: boolean;
28 +     connections: IConnectedAppConnection[];
29 +
30 +   export interface IConnectedAppsStore {
31 +     connectedApps: {
32 +       [chainName: string]: {
33 +         [walletAddress: string]: {
34 +           [domain: string]: IConnectedApp;
35 +         };
36 +       };
37 +     };
38 +     actions: {
39 +       saveAppConnection: (
40 +         chainName: 'mainnet' | 'testnet',
41 +         walletAddress: string,
42 +         appData: Omit<IConnectedApp, 'connections'>,
43 +         connection: IConnectedAppConnection,
44 +       ) => void;
45 +       removeApp: (
46 +         chainName: 'mainnet' | 'testnet',
47 +         walletAddress: string,
48 +         url: string,
49 +       ) => void;
50 +       removeInjectedConnection: (
51 +         chainName: 'mainnet' | 'testnet',
52 +         walletAddress: string,
53 +         url: string,
54 +       ) => void;
55 +     };
56 +   }

```

```

v 18 ████ app/store/zustand/connectedApps/useConnectedAppsList.ts ⓘ ...
... @@ -0,0 +1,18 @@
...
1 + import { getChainName } from '../../../../../shared/dynamicConfig';
2 + import { useCallback } from 'react';
3 + import { IConnectedApp } from './types';
4 + import { useConnectedAppsStore } from './useConnectedAppsStore';
5 + import * as React from 'react';
6 + import {getCurrentAddress} from "../../storage/appState";
7 + import {AppConfig} from "../../..../AppConfig";
8 +
9 + export const useConnectedAppsList = (): IConnectedApp[] => {
10 +   const address = React.useMemo(() => getCurrentAddress().address, []);
11 +
12 +   return useConnectedAppsStore(
13 +     useCallback(
14 +       (s) => Object.values(s.connectedApps[getChainName()]).filter((app) => !app.isTestnet),
15 +       [address,toFriendly((testOnly: AppConfig.isTestnet))],
16 +     ),
17 +   );
18 + }

```

```

v 95 ████ app/store/zustand/connectedApps/useConnectedAppsStore.ts ⓘ ...
... @@ -0,0 +1,95 @@
...
1 + import {generateAppHashFromUrl} from '../../../../../utils/misc';
2 + import AsyncStorage from '@react-native-async-storage/async-storage';
3 + import create from 'zustand';
4 + import { persist, subscribeWithSelector } from 'zustand/middleware';
5 + import { IConnectedAppsStore, TonConnectBridgeType } from './types';
6 +
7 + const initialState: Omit<IConnectedAppsStore, 'actions'> = {
8 +   connectedApps: {
9 +     mainnet: {},
10 +     testnet: {},
11 +   },
12 + };
13 +
14 + export const useConnectedAppsStore = create(
15 +   subscribeWithSelector(
16 +     persist(IConnectedAppsStore)<
17 +       (set) => (
18 +         ...initialState,
19 +         actions: {
20 +           saveAppConnection: (chainName, walletAddress, appData, connection) => {
21 +             set((connectedApps) => {
22 +               if (!connectedApps[chainName][walletAddress]) {
23 +                 connectedApps[chainName][walletAddress] = {};
24 +               }
25 +
26 +               const hash = generateAppHashFromUrl(appData.url);
27 +
28 +               const alreadyConnectedApp = connectedApps[chainName][walletAddress][hash];
29 +
30 +               if (alreadyConnectedApp) {
31 +                 connectedApps[chainName][walletAddress][hash] = {
32 +                   ...alreadyConnectedApp,
33 +                   ...appData,
34 +                   icon: appData.icon || alreadyConnectedApp.icon,
35 +                   autoConnectDisabled:
36 +                     alreadyConnectedApp.autoConnectDisabled &
37 +                     connection.type !== TonConnectBridgeType.Injected,
38 +                   connections: connection
39 +                     ? [...alreadyConnectedApp.connections, connection]
40 +                     : alreadyConnectedApp.connections,
41 +                 };
42 +               } else {
43 +                 connectedApps[chainName][walletAddress][hash] = {
44 +                   ...appData,
45 +                   connections: connection ? [connection] : [],
46 +                 };
47 +               }
48 +
49 +               return { connectedApps };
50 +             });
51 +           },
52 +           removeInjectedConnection: (chainName, walletAddress, url) => {
53 +             const hash = generateAppHashFromUrl(url);
54 +
55 +             set((connectedApps) => {
56 +               if (connectedApps[chainName][walletAddress].hasOwnProperty(hash)) {
57 +                 connectedApps[chainName][walletAddress][hash].autoConnectDisabled = true;
58 +
59 +                 connectedApps[chainName][walletAddress][hash].connections = connectedApps[
60 +                   chainName
61 +                 ][walletAddress][hash].connections.filter(
62 +                   (connection) => connection.type !== TonConnectBridgeType.Injected,
63 +                 );
64 +
65 +                 if (
66 +                   connectedApps[chainName][walletAddress][hash].connections.length === 0
67 +                 ) {

```

```
68 +         delete connectedApps[chainName][walletAddress][hash];
69 +     }
70 +   }
71 +
72 +   return { connectedApps };
73 + });
74 + },
75 + removeApp: (chainName, walletAddress, url) => {
76 +   const hash = generateAppHashFromUrl(url);
77 +
78 +   set((( connectedApps )) => [
79 +     if (connectedApps[chainName][walletAddress].hasOwnProperty(hash)) {
80 +       delete connectedApps[chainName][walletAddress][hash];
81 +     }
82 +
83 +     return { connectedApps };
84 +   ]);
85 + },
86 + )),
87 + )),
88 + {
89 +   name: 'TCApps',
90 +   getStorage: () => AsyncStorage,
91 +   partialize: (( connectedApps )) => ({ connectedApps } as IConnectedAppsStore),
92 + },
93 + ),
94 + ),
95 + );
```

```
✓ 3 ━━━━ app/store/zustand/devFeaturesToggle/index.ts ⌂ ...
... ... @@ -0,0 +1,3 @@
1 + export * from './useDevFeaturesToggle';
2 + export * from './types';
3 + export * from './useDevFeatureEnabled';
```

```
✓ 10 ━━━━ app/store/zustand/devFeaturesToggle/types.ts ⌂ ...
... ... @@ -0,0 +1,10 @@
1 + export enum DevFeature {
2 +   Example = 'Example',
3 + }
4 +
5 + export interface IDevFeaturesToggleStore {
6 +   devFeatures: { [key in DevFeature]: boolean };
7 +   actions: {
8 +     toggleFeature: (feature: DevFeature) => void;
9 +   };
10 + }
```

```
✓ 7 ━━━━ app/store/zustand/devFeaturesToggle/useDevFeatureEnabled.ts ⌂ ...
... ... @@ -0,0 +1,7 @@
1 + import { DevFeature, useDevFeaturesToggle } from '../../../../../store';
2 +
3 + export const useDevFeatureEnabled = (feature: DevFeature) => {
4 +   const { devFeatures } = useDevFeaturesToggle();
5 +
6 +   return devFeatures[feature];
7 + },
```

```
✓ 33 ━━━━ app/store/zustand/devFeaturesToggle/useDevFeaturesToggle.ts ⌂ ...
... ... @@ -0,0 +1,33 @@
1 + import AsyncStorage from '@react-native-async-storage/async-storage';
2 + import create from 'zustand';
3 + import { persist } from 'zustand/middleware';
4 + import { DevFeature, IDevFeaturesToggleStore } from './types';
5 +
6 + const initialState: Omit = {
7 +   devFeatures: {
8 +     [DevFeature.Example]: false,
9 +   },
10 + };
11 +
12 + export const useDevFeaturesToggle = create(
13 +   persist(
14 +     (set) => {
15 +       ...initialState,
16 +       actions: {
17 +         toggleFeature: async (name: DevFeature) => {
18 +           set((state) => {
19 +             const devFeatures = state.devFeatures;
20 +             devFeatures[name] = !devFeatures[name];
21 +
22 +             return { devFeatures };
23 +           });
24 +         },
25 +       },
26 +     },
27 +   ),
28 +   name: 'devFeaturesToggle',
29 +   getStorage: () => AsyncStorage,
30 +   partialize: (( devFeatures ) => ({ devFeatures } as IDevFeaturesToggleStore),
31 +   ),
32 + ),
33 + );
```

```
✓ 3 ━━━━ app/store/zustand/index.ts ⌂ ...
... ... @@ -0,0 +1,3 @@
1 + export * from './connectedApps';
2 + export * from './appList';
3 + export * from './devFeaturesToggle';
```

```
✓ 17 ━━━━ app/ionconnect/ConnectEventError.ts ⌂ ...
... ... @@ -0,0 +1,17 @@
1 + import {
2 +   ConnectEventError as IConnectEventError,
3 +   CONNECT_EVENT_ERROR_CODES,
4 + } from '@ionconnect/protocol';
5 +
6 + export class ConnectEventError implements IConnectEventError {
7 +   event: IConnectEventError['event'];
8 +   payload: IConnectEventError['payload'];
9 +
10 +   constructor(code = CONNECT_EVENT_ERROR_CODES.UNKNOWN_ERROR, message: string) {
11 +     this.event = 'connect_error';
12 +     this.payload = {
13 +       code,
14 +       message,
15 +     };
16 +   }
17 + }
```

```
17 | + }
18 |
19 | <!-- 142 --> app/onconnect/ConnectReplyBuilder.ts
20 | ...
21 | ...
22 | ...
23 | ...
24 | ...
25 | ...
26 | ...
27 | ...
28 | ...
29 | ...
30 | ...
31 | ...
32 | ...
33 | ...
34 | ...
35 | ...
36 | ...
37 | ...
38 | ...
39 | ...
40 | ...
41 | ...
42 | ...
43 | ...
44 | ...
45 | ...
46 | ...
47 | ...
48 | ...
49 | ...
50 | ...
51 | ...
52 | ...
53 | ...
54 | ...
55 | ...
56 | ...
57 | ...
58 | ...
59 | ...
60 | ...
61 | ...
62 | ...
63 | ...
64 | ...
65 | ...
66 | ...
67 | ...
68 | ...
69 | ...
70 | ...
71 | ...
72 | ...
73 | ...
74 | ...
75 | ...
76 | ...
77 | ...
78 | ...
79 | ...
80 | ...
81 | ...
82 | ...
83 | ...
84 | ...
85 | ...
86 | ...
87 | ...
88 | ...
89 | ...
90 | ...
91 | ...
92 | ...
93 | ...
94 | ...
95 | ...
96 | ...
97 | ...
98 | ...
99 | ...
100 | ...
101 | ...
102 | ...
103 | ...
104 | ...
105 | ...
106 | ...
107 | ...
108 | ...
109 | ...
110 | ...
111 | ...
112 | ...
113 | ...
114 | ...
115 | ...
116 | ...
117 | ...
118 | ...
119 | ...
120 | ...
121 | ...
```

```
122 +         name: (requestItem as ConnectItem).name,
123 +         error: { code: CONNECT_EVENT_ERROR_CODES.METHOD_NOT_SUPPORTED },
124 +     } as unknown as ConnectItemReply;
125 +   }
126 + });
127 +
128 +   return replyItems;
129 + }
130 +
131 + static createAutoConnectReplyItems(addr: string): ConnectItemReply[] {
132 +   const address = new TonWeb.utils.Address(addr).toString(false, true, true);
133 +   return [
134 +     {
135 +       name: 'ton_addr',
136 +       address,
137 +       network: ConnectReplyBuilder.getNetwork(),
138 +       walletStateInit: walletStateInit,
139 +     },
140 +   ];
141 + }
142 + }
```

```
✓ 23 app/onconnect/SendTransactionError.ts ...
...
1 + import {
2 +   SendTransactionRpcResponseError,
3 +   SEND_TRANSACTION_ERROR_CODES,
4 + } from '@tonconnect/protocol';
5 +
6 + export class SendTransactionError implements SendTransactionRpcResponseError {
7 +   id: SendTransactionRpcResponseError['id'];
8 +   error: SendTransactionRpcResponseError['error'];
9 +
10+   constructor(
11+     requestId: string,
12+     code: SEND_TRANSACTION_ERROR_CODES,
13+     message: string,
14+     data?: any,
15+   ) {
16+     this.id = requestId;
17+     this.error = {
18+       code,
19+       message,
20+       data,
21+     };
22+   }
23+ }
```

```
✓ 345 app/onconnect/TonConnect.ts ...
...
1 + import {
2 +   findConnectedAppByClientSessionId,
3 +   findConnectedAppByUrl,
4 +   IConnectedApp,
5 +   saveAppConnection,
6 +   removeConnectedApp,
7 +   TonConnectBridgeType,
8 +   IConnectedAppConnectionRemote
9 + } from './store';
10+ import {getTimeSec} from '../utils/getTimeSec';
11+ import {
12+   AppRequest,
13+   ConnectEvent,
14+   ConnectRequest,
15+   CONNECT_EVENT_ERROR_CODES,
16+   RpcMethod,
17+   SEND_TRANSACTION_ERROR_CODES,
18+   SessionCrypto,
19+   WalletResponse,
20+ } from '@tonconnect/protocol';
21+ import axios from 'axios';
22+ import FastImage from 'react-native-fast-image';
23+ import {MIN_PROTOCOL_VERSION, tonConnectDeviceInfo} from './config';
24+ import {ConnectEventError} from './ConnectEventError';
25+ import {ConnectReplyBuilder} from './ConnectReplyBuilder';
26+ import {DAppManifest, SignRawParams, TransactionModalParams} from './models';
27+ import {SendTransactionError} from './SendTransactionError';
28+ import {TonConnectRemoteBridge} from './TonConnectRemoteBridge';
29+ import {getCurrentAddress} from './storage/appState';
30+ import {AppConfig} from './AppConfig';
31+ import {getAppsConnectModalTriggerObject, getTransactionModalTriggerObject} from './modalTrigger';
32+
33+ class TonConnectService {
34+
35+   checkProtocolVersionCapability(protocolVersion: number) {
36+     if (typeof protocolVersion !== 'number' || protocolVersion < MIN_PROTOCOL_VERSION) {
37+       throw new ConnectEventError(
38+         CONNECT_EVENT_ERROR_CODES.BAD_REQUEST_ERROR,
39+         `Protocol version ${String(protocolVersion)} is not supported by the wallet app`,
40+       );
41+     }
42+   }
43+
44+   verifyConnectRequest(request: ConnectRequest) {
45+     if (!request || !request.manifestUrl || request.items?.length) {
46+       throw new ConnectEventError(
47+         CONNECT_EVENT_ERROR_CODES.BAD_REQUEST_ERROR,
48+         'Wrong request data',
49+       );
50+     }
51+   }
52+
53+   async getManifest(request: ConnectRequest) {
54+     try {
55+       const {data: manifest} = await axios.get<DAppManifest>(request.manifestUrl);
56+
57+       const isValid =
58+         manifest &&
59+         typeof manifest.url === 'string' &&
60+         typeof manifest.name === 'string' &&
61+         typeof manifest.iconUrl === 'string';
62+
63+       if (!isValid) {
64+         throw new ConnectEventError(
65+           CONNECT_EVENT_ERROR_CODES.MANIFEST_CONTENT_ERROR,
66+           'Manifest is not valid',
67+         );
68+       }
69+
70+       if (manifest.iconUrl) {
71+         FastImage.preload([{uri: manifest.iconUrl}]);
72+       }
73+     } catch (e) {
74+       throw new ConnectEventError(
75+         CONNECT_EVENT_ERROR_CODES.MANIFEST_ERROR,
76+         `Error while getting manifest: ${e.message}`,
77+       );
78+     }
79+   }
80+ }
```

```

    +     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 connectedAppModal = getAppsConnectModalTriggerObject();
  +           connectedAppModal.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
  +       );
  +       if (type === TonConnectBridgeType.Injected) {
  +         replyItems.push({
  +           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()

```

```

201 +         request: AppRequest<'sendTransaction'>,
202 +     ): Promise<WalletResponse<'sendTransaction'>> {
203 +       try {
204 +         const params = JSON.parse(request.params[0]) as any;
205 +
206 +         const isValidRequest =
207 +           params &&
208 +           typeof params.valid_until === 'number' &&
209 +           Array.isArray(params.messages) &&
210 +           params.messages.every((msg: any) => !msg.address && !msg.amount);
211 +
212 +         if (!isValidRequest) {
213 +           throw new SendTransactionError(
214 +             request.id,
215 +             SEND_TRANSACTION_ERROR_CODES.BAD_REQUEST_ERROR,
216 +             'Bad request',
217 +           );
218 +         }
219 +
220 +         const {valid_until, messages} = params;
221 +
222 +         if (valid_until < getTmeSec()) {
223 +           throw new SendTransactionError(
224 +             request.id,
225 +             SEND_TRANSACTION_ERROR_CODES.BAD_REQUEST_ERROR,
226 +             'Request timed out',
227 +           );
228 +         }
229 +
230 +         const currentWalletAddress = getCurrentAddress().address.toFriendly({testOnly: AppConfig.isTestnet});
231 +
232 +         const txParams: SignRawParams = {
233 +           valid_until,
234 +           messages,
235 +           source: currentWalletAddress,
236 +         };
237 +
238 +         const boc = await new Promise<string>(async (resolve, reject) => {
239 +           const transactionModalObject = getTransactionModalTriggerObject();
240 +           const dataForTransactionModal: TransactionModalParams = {
241 +             params: txParams,
242 +             onSuccess: resolve,
243 +             onDismiss: () =>
244 +               reject(
245 +                 new SendTransactionError(
246 +                   request.id,
247 +                   SEND_TRANSACTION_ERROR_CODES.USER_REJECTS_ERROR,
248 +                   'Wallet declined the request',
249 +                 ),
250 +               ),
251 +             options: {
252 +               expires_sec: valid_until,
253 +               response_options: {
254 +                 broadcast: false,
255 +               }
256 +             }
257 +           };
258 +           transactionModalObject.subject.next({
259 +             params: dataForTransactionModal.params,
260 +             options: dataForTransactionModal.options,
261 +             rResolve: dataForTransactionModal.onSuccess,
262 +             rReject: dataForTransactionModal.onDismiss
263 +           });
264 +         });
265 +
266 +         return {
267 +           result: boc,
268 +           id: request.id,
269 +         };
270 +       } catch (error) {
271 +         if (error instanceof SendTransactionError) {
272 +           return error;
273 +         }
274 +
275 +         return new SendTransactionError(
276 +           request.id,
277 +           SEND_TRANSACTION_ERROR_CODES.UNKNOWN_ERROR,
278 +           error?.message,
279 +         );
280 +       }
281 +     }
282 +
283 +     private async handleRequest<T extends RpcMethod>(
284 +       request: AppRequest<T>,
285 +       connectedApp: IConnectedApp | null,
286 +     ): Promise<WalletResponse<T>> {
287 +       if (!connectedApp) {
288 +         return {
289 +           error: {
290 +             code: SEND_TRANSACTION_ERROR_CODES.UNKNOWN_APP_ERROR,
291 +             message: 'Unknown app',
292 +           },
293 +           id: request.id,
294 +         };
295 +       }
296 +       if (request.method === 'sendTransaction') {
297 +         return this.sendTransaction(request);
298 +       }
299 +
300 +       return {
301 +         error: {
302 +           code: SEND_TRANSACTION_ERROR_CODES.BAD_REQUEST_ERROR,
303 +           message: `Method "${request.method}" does not supported by the wallet app`,
304 +         },
305 +         id: request.id,
306 +       };
307 +     }
308 +
309 +     async handleRequestFromInjectedBridge<T extends RpcMethod>(
310 +       request: AppRequest<T>,
311 +       webViewUrl: string,
312 +     ): Promise<WalletResponse<T>> {
313 +       const connectedApp = findConnectedAppByUrl(webViewUrl);
314 +       console.log('HERE ... 1');
315 +       return this.handleRequest(request, connectedApp);
316 +     }
317 +
318 +     async handleRequestFromRemoteBridge<T extends RpcMethod>(
319 +       request: AppRequest<T>,
320 +       clientSessionId: string,
321 +     ): Promise<WalletResponse<T>> {
322 +       const connectedApp = findConnectedAppByClientSessionId(clientSessionId);
323 +       return this.handleRequest(request, connectedApp);
324 +     }
325 +
326 +     async disconnect(url: string) {

```

```

327 +     const connectedApp = findConnectedAppByUrl(url);
328 +
329 +     if (!connectedApp) {
330 +         return;
331 +     }
332 +
333 +     const remoteConnections = connectedApp.connections.filter(
334 +       (connection) => connection.type === TonConnectBridgeType.Remote,
335 +     ) as IConnectedAppConnectionRemote[];
336 +
337 +     remoteConnections.forEach((connection) =>
338 +       TonConnectRemoteBridge.sendDisconnectEvent(connection),
339 +     );
340 +
341 +     removeConnectedApp(url);
342 +
343 +   }
344 +
345 + export const TonConnect = new TonConnectService();

```

```

✓ 239 ██████████ app/tonconnect/TonConnectRemoteBridge.ts □ ...
...
```

```

... ... @@ -0,0 +1,239 @@
1 + import {
2 +   IConnectedAppConnection,
3 +   IConnectedAppConnectionRemote,
4 +   TonConnectBridgeType,
5 + } from './store';
6 + import {
7 +   AppRequest,
8 +   Base64,
9 +   ConnectEvent,
10 +   ConnectRequest,
11 +   DisconnectEvent,
12 +   hexToByteArray,
13 +   RpcMethod,
14 +   SEND_TRANSACTION_ERROR_CODES,
15 +   SessionCrypto,
16 +   WalletResponse,
17 + } from '@tonconnect/protocol';
18 + import debounce from "lodash/debounce";
19 + import { TonConnect } from './TonConnect';
20 + import { IConnectQuery } from './models';
21 + import EventSource, { EventSourceListener, MessageEvent } from 'react-native-sse';
22 + import { storage as AsyncStorage } from '../storage/storage';
23 + import { DeepLinkOrigin } from '../libs/deplinking';
24 + import Minimizer from 'react-native-minimizer';
25 + import { getCurrentAddress } from '../storage/appState';
26 + import { Tonapi } from '../libs/tонапи';
27 + import { t } from '../libs/t';
28 +
29 + class TonConnectRemoteBridgeService {
30 +   private readonly storeKey = 'ton-connect-http-bridge-lastEventId';
31 +
32 +   private readonly bridgeUrl = 'https://bridge.tonapi.io/bridge';
33 +
34 +   private readonly defaultTtl = 300;
35 +
36 +   private eventSource: EventSource | null = null;
37 +
38 +   private connections: IConnectedAppConnectionRemote[] = [];
39 +
40 +   private activeRequests: { [From: string]: AppRequest<RpcMethod> } = {};
41 +
42 +   private origin: DeepLinkOrigin | null = null;
43 +
44 +   public setOrigin(origin: DeepLinkOrigin) {
45 +     this.origin = origin;
46 +   }
47 +
48 +   async open(connections: IConnectedAppConnection[]) {
49 +     this.close();
50 +
51 +     this.connections = connections.filter(
52 +       (item) => item.type === TonConnectBridgeType.Remote,
53 +     ) as IConnectedAppConnectionRemote[];
54 +
55 +     if (this.connections.length === 0) {
56 +       return;
57 +     }
58 +
59 +     const walletSessionIds = this.connections
60 +       .map((item) => new SessionCrypto(item.sessionKeyValuePair).sessionId)
61 +       .join(',');
62 +
63 +     let url = `${this.bridgeUrl}/events?client_id=${walletSessionIds}`;
64 +
65 +     const lastEventId = await this.getLastEventId();
66 +
67 +     if (lastEventId) {
68 +       url += `&last_event_id=${lastEventId}`;
69 +     }
70 +
71 +     console.log('sse connect', url);
72 +
73 +     this.eventSource = new EventSource(url);
74 +
75 +     this.eventSource.addEventListener(
76 +       'message',
77 +       debounce(this.handleMessage.bind(this), 200) as EventSourceListener,
78 +     );
79 +
80 +     this.eventSource.addEventListener('open', () => {
81 +       console.log('sse connect: opened');
82 +     });
83 +
84 +     this.eventSource.addEventListener('error', (event) => {
85 +       console.log('sse connect: error', event);
86 +     });
87 +
88 +
89 +     close() {
90 +       if (this.eventSource) {
91 +         this.eventSource.removeAllEventListeners();
92 +         this.eventSource.close();
93 +         this.eventSource = null;
94 +         console.log('sse close');
95 +       }
96 +     }
97 +
98 +     private async setLastEventId(lastEventId: string) {
99 +       try {
100 +         await AsyncStorage.set(this.storeKey, lastEventId);
101 +       } catch {}
102 +     }

```

```

103 +
104 +     private async getLastEventId() {
105 +         try {
106 +             return await AsyncStorage.getString(this.storeKey);
107 +         } catch {
108 +             return null;
109 +         }
110 +     }
111 +
112 +     private async sendT extends RpcMethod<{
113 +         response: WalletResponseT | ConnectEvent | DisconnectEvent,
114 +         sessionCrypto: SessionCrypto,
115 +         clientSessionId: string,
116 +         ttl?: number,
117 +     }: Promise<void> {
118 +         try {
119 +             const url = `${this.bridgeUrl}/message?client_id=${sessionCrypto.sessionId}&to=${clientSessionId}&ttl=${(ttl || this.defaultTtl)}`;
120 +
121 +             const encodedResponse = sessionCrypto.encrypt(
122 +                 JSON.stringify(response),
123 +                 hexToByteArray(clientSessionId),
124 +             );
125 +
126 +             await fetch(url, {
127 +                 body: Base64.encode(encodedResponse),
128 +                 method: 'POST',
129 +             });
130 +         } catch (e) {
131 +             console.log('send fail', e);
132 +         }
133 +     }
134 +
135 +
136 +     private async handleMessage(event: MessageEvent) {
137 +         try {
138 +             if (event.lastEventId) {
139 +                 this.setLastEventId(event.lastEventId);
140 +             }
141 +
142 +             const { from, message } = JSON.parse(event.data!);
143 +
144 +             console.log('handleMessage', from);
145 +
146 +             const connection = this.connections.find((item) => item.clientSessionId === from);
147 +
148 +             if (!connection) {
149 +                 console.log(`connection with clientId "${from}" not found!`);
150 +                 return;
151 +             }
152 +
153 +             const sessionCrypto = new SessionCrypto(connection.sessionKeyPair);
154 +             const request: AppRequest<RpcMethod> = JSON.parse(
155 +                 sessionCrypto.decrypt(
156 +                     Base64.decode(message).toUint8Array(),
157 +                     hexToByteArray(from),
158 +                 ),
159 +             );
160 +
161 +
162 +             if (this.activeRequests[from]) {
163 +                 await this.send(
164 +                     {
165 +                         error: {
166 +                             code: SEND_TRANSACTION_ERROR_CODES.USER_REJECTS_ERROR,
167 +                             message: 'User has already opened the previous request',
168 +                         },
169 +                         id: request.id,
170 +                     },
171 +                     sessionCrypto,
172 +                     from,
173 +                 );
174 +
175 +                 return;
176 +             }
177 +
178 +             this.activeRequests[from] = request;
179 +
180 +             console.log('HandleMessage request', request);
181 +
182 +             const response = await TonConnect.handleRequestFromRemoteBridge(request, from);
183 +
184 +             delete this.activeRequests[from];
185 +
186 +             console.log('HandleMessage response', response);
187 +
188 +             await this.send(response, sessionCrypto, from);
189 +
190 +             this.redirectIfNeeded();
191 +         } catch (e) {
192 +             console.log('HandleMessage error');
193 +             console.error(e);
194 +         }
195 +     }
196 +
197 +     private redirectIfNeeded() {
198 +         if (this.origin === DeeplinkOrigin.DEEPLINK) {
199 +             Minimizer.goBack();
200 +         }
201 +
202 +         this.origin = null;
203 +     }
204 +
205 +     async handleConnectDeepLink(query: IConnectQrQuery) {
206 +         // if (typeof query.v === 'undefined') {
207 +         //     alert('applications.cannotConnectMessage');
208 +         //     return;
209 +         // }
210 +
211 +         try {
212 +             const protocolVersion = Number(query.v);
213 +             const request = JSON.parse(decodeURIComponent(query.r)) as ConnectRequest;
214 +             const clientSessionId = query.id;
215 +             console.log('HandleConnectDeepLink request', request);
216 +             const sessionCrypto = new SessionCrypto();
217 +             const response = await TonConnect.connect(
218 +                 protocolVersion,
219 +                 request,
220 +                 sessionCrypto,
221 +                 clientSessionId,
222 +             );
223 +             console.log('HandleConnectDeepLink response', response.event);
224 +             await this.send(response, sessionCrypto, clientSessionId);
225 +             // this.redirectIfNeeded();
226 +         } catch (err) {
227 +             console.log('HandleConnectDeepLink error', err);
228 +         }
229 +     }

```

```
229 +
230 +     sendDisconnectEvent(connection: IConnectedAppConnectionRemote) {
231 +         const sessionCrypto = new SessionCrypto(connection.sessionKeyPair);
232 +
233 +         const event: DisconnectEvent = { event: 'disconnect', payload: {} };
234 +
235 +         this.send(event, sessionCrypto, connection.clientSessionId);
236 +     }
237 +
238 +
239 +     export const TonConnectRemoteBridge = new TonConnectRemoteBridgeService();
```

```
✓ 23 app/tонconnect/config.ts
...
...
@@ -0,0 +1,23 @@
1 + import { DeviceInfo } from '@tonconnect/protocol';
2 + import RNDeviceInfo From 'react-native-device-info';
3 + import { Platform } From 'react-native';
4 +
5 + export const CURRENT_PROTOCOL_VERSION = 2;
6 +
7 + export const MIN_PROTOCOL_VERSION = 2;
8 +
9 + const getPlatform = (): DeviceInfo['platform'] => {
10 +     if (Platform.OS === 'ios') {
11 +         return RNDeviceInfo.isTablet() ? 'ipad' : 'iphone';
12 +     }
13 +
14 +     return Platform.OS as DeviceInfo['platform'];
15 + };
16 +
17 + export const tonConnectDeviceInfo: DeviceInfo = {
18 +     platform: getPlatform(),
19 +     appName: RNDeviceInfo.getApplicationName(),
20 +     appVersion: RNDeviceInfo.getReadableVersion(),
21 +     maxProtocolVersion: CURRENT_PROTOCOL_VERSION,
22 +     features: ['SendTransaction'],
23 + };

```

```
✓ 275 app/tонconnect/connection-view/ExternalAppConnection.tsx
...
...
@@ -0,0 +1,275 @@
1 + import React, {useState} From "react";
2 + import {Text, View} From "react-native";
3 + import Modal From "react-native-modal";
4 + import {Theme} From ".../.../Theme";
5 + import LottieView From "lottie-react-native";
6 + import {t, t$styled} From ".../.../libN$ts";
7 + import {RoundButton} From ".../.../components/RoundButton";
8 + import {WImage} From ".../.../components/WImage";
9 + import {getAppsConnectModalTriggerObject} From ".../modalTrigger";
10 + import {getDarkModeValue} From ".../.../utils/misc";
11 + import ProtectedIcon From ".../.../assets/ic_protected.svg";
12 + import {DAppManifest, tonConnectResponseModel} From ".../models";
13 + import {ConnectReplyBuilder} From ".../ConnectReplyBuilder";
14 + import {contractFromPublicKey} From ".../.../engine/contractFromPublicKey";
15 + import {getCurrentAddress} From ".../.../storage/appState";
16 + import {loadWalletKeys, WalletKeys} From ".../.../storage/walletKeys";
17 + import {AppConfig} From ".../.../AppConfig";
18 + import {warn} From ".../.../utils/log";
19 +
20 + type stateType = {
21 +     applicationUrl: string, appTitle: string, protocolVersion: number, manifest: DAppManifest, replyBuilder: ConnectReplyBuilder, requestPromise: {
22 +         resolve: (response: TonConnectResponseModel) => void;
23 +         reject: () => void;
24 +     }, hideImmediately: boolean
25 + } | null;
26 +
27 + function ExternalAppConnection() {
28 +     const connectedAppsModalObject = getAppsConnectModalTriggerObject();
29 +     const [state, setState] = useState<stateType>(null);
30 +     const [currentAction, setCurrentAction] = useState<{action: 'requesting', message: ''}>; // requesting - approved - rejected - expired
31 +     React.useEffect(() => {
32 +         const connectedAppsModalSubscriber = connectedAppsModalObject.observable.subscribe((data: any) => {
33 +             // if (data.protocolVersion !== 1) {
34 +             //     data.requestPromise.reject();
35 +             // }
36 +             const collectedData: stateType = {
37 +                 hideImmediately: false,
38 +                 applicationUrl: data.manifest.iconUrl,
39 +                 appTitle: data.manifest.name,
40 +                 protocolVersion: data.protocolVersion,
41 +                 manifest: data.manifest,
42 +                 replyBuilder: data.replyBuilder,
43 +                 requestPromise: data.requestPromise
44 +             }
45 +             setState(collectedData);
46 +         });
47 +         return function cleanup() {
48 +             connectedAppsModalSubscriber.unsubscribe();
49 +         }
50 +     }, [connectedAppsModalObject]);
51 +
52 +     const acc = React.useMemo(() => getCurrentAddress(), []);
53 +     let active = React.useRef(true);
54 +     React.useEffect(() => {
55 +         return () => {
56 +             active.current = false;
57 +         };
58 +     }, []);
59 +
60 +     const createResponse = React.useCallback(async () => {
61 +         try {
62 +             if (state) {
63 +                 const contract = await contractFromPublicKey(acc.publicKey);
64 +                 let address = contract.address.toFriendly({testOnly: AppConfig.isTestnet});
65 +                 let walletKeys: WalletKeys;
66 +                 try {
67 +                     walletKeys = await loadWalletKeys(acc.secretKeyEnc);
68 +                 } catch (e) {
69 +                     warn(e);
70 +                     return;
71 +                 }
72 +
73 +                 const protocolVersion = state?.protocolVersion;
74 +                 if (protocolVersion !== 1) {
75 +                     const { replyBuilder, requestPromise } = state;
76 +                     const replyItems = replyBuilder.createReplyItems(address, walletKeys.keyPair.secretKey);
77 +                     requestPromise.resolve({address, replyItems});
78 +                 }
79 +                 setCurrentAction({action: 'approved', message: t('auth.authorized'))};
80 +             }
81 +         } catch (error) {
82 +             let message = error.message;
83 +             setCurrentAction({action: 'rejected', message: message});
84 +         }
85 +     }, []);
86 + }
```

```

  64 +       )
  65 +     ), [state]);
  66 +
  67 +   const cancelRequest = React.useCallback(async () => {
  68 +     if (state && state.requestPromise) {
  69 +       state.requestPromise.reject();
  70 +       setCurrentAction({action: 'rejected', message: t('auth.rejected')});
  71 +     }
  72 +   ), [state]);
  73 +
  74 +   const closeModal = React.useCallback(async () => {
  75 +     setState(null);
  76 +     setCurrentAction({action: 'requesting', message: ''});
  77 +   ), []);
  78 +
  79 +   return (
  80 +     <Modal isVisible={!state}>
  81 +       <View style={{width: '100%', backgroundColor: Theme.item, paddingHorizontal: 5, paddingVertical: 10}}>
  82 +         <text style={{
  83 +           fontSize: 19,
  84 +           fontWeight: '600',
  85 +           color: Theme.textColor,
  86 +           textAlign: 'center',
  87 +           textAlignVertical: 'center'
  88 +         }}>{t('auth.title')}

```

```

210 +           )}
211 +
212 +           <LottieView
213 +             source={require '../../../../../assets/animations/green-check.json'}
214 +             style={{width: 140}}
215 +             autoPlay={true}
216 +             loop={true}
217 +           />
218 +           <Text style={{
219 +             marginTop: 5,
220 +             fontSize: 16,
221 +             fontWeight: '600',
222 +             color: Theme.primeGreen,
223 +             textAlign: 'center'
224 +           }}>{currentAction.message}</Text>
225 +           <Text style={{
226 +             marginTop: 5,
227 +             fontSize: 12,
228 +             fontWeight: '400',
229 +             color: Theme.textColor,
230 +             textAlign: 'left'
231 +           }}>{('auth.authorizedDescription')}</Text>
232 +           <RoundButton
233 +             title={'('+'common.close'+')'}
234 +             display={(getDarkModeValue() ? 'outline' : 'default')}
235 +             style={{marginTop: 25, width: '60%'}}
236 +             action={closeModal}
237 +           />
238 +         </View>
239 +       )
240 +
241 +     {currentAction.action === 'rejected' && (
242 +       <View style={{
243 +         flexGrow: 1,
244 +         width: '100%',
245 +         alignItems: 'center',
246 +         justifyContent: 'center',
247 +         padding: 20
248 +       }}>
249 +         <LottieView
250 +           source={require '../../../../../assets/animations/red-cross.json'}
251 +           style={{width: 140}}
252 +           autoPlay={true}
253 +           loop={true}
254 +         />
255 +         <Text style={{
256 +           marginTop: 5,
257 +           fontSize: 16,
258 +           fontWeight: '600',
259 +           color: Theme.primeRed
260 +         }}>{currentAction.message}</Text>
261 +         <RoundButton
262 +           title={'('+'common.close'+')'}
263 +             display={(getDarkModeValue() ? 'outline' : 'default')}
264 +             style={{marginTop: 25, width: '60%'}}
265 +             action={closeModal}
266 +           />
267 +         </View>
268 +       )
269 +
270 +     </View>
271 +   );
272 + }
273 +
274 +
275 + export default ExternalAppConnection;

```

...

1 1 app/tonconnect/hooks/index.ts

... ... @@ -0,0 +1 @@

1 + export * from './useRemoteBridge';

...

26 26 app/tonconnect/hooks/useRemoteBridge.ts

... ... @@ -0,0 +1,26 @@

```

1 + import { getAllConnections, useConnectedAppsStore } from '../../../../../store';
2 + import { useEffect } from 'react';
3 + import { TonConnectRemoteBridge } from '../TonConnectRemoteBridge';
4 + import { getCurrentAddress } from '../../../../../storage/appState';
5 + import { AppConfig } from '../../../../../AppConfig';
6 +
7 + export const useRemoteBridge = () => {
8 +   const { address } = getCurrentAddress();
9 +   useEffect(() => {
10 +     const initialConnections = getAllConnections(
11 +       useConnectedAppsStore.getState(),
12 +       address.toFriendly({testOnly: AppConfig.isTestnet}),
13 +     );
14 +     TonConnectRemoteBridge.open(initialConnections);
15 +     const unsubscribe = useConnectedAppsStore.subscribe(
16 +       (s: any) => getAllConnections(s, address.toFriendly({testOnly: AppConfig.isTestnet})),
17 +       (connections: any) => {
18 +         TonConnectRemoteBridge.open(connections);
19 +       },
20 +     );
21 +     return () => {
22 +       unsubscribe();
23 +       TonConnectRemoteBridge.close();
24 +     };
25 +   }, [address.toFriendly({testOnly: AppConfig.isTestnet})]);
26 + };

```

...

6 6 app/tonconnect/index.ts

... ... @@ -0,0 +1,6 @@

```

1 + export * from './config';
2 + export * from './TonConnect';
3 + export * from './ConnectReplyBuilder';
4 + export * from './TonConnectRemoteBridge';
5 + export * from './hooks';
6 + export * from './models';

```

...

12 12 app/tonconnect/modalTrigger.ts

... ... @@ -0,0 +1,12 @@

```

1 + import {Subject} from 'rxjs';
2 +
3 + const modalTriggerSubject = new Subject();
4 + const transactionModalTriggerSubject = new Subject();
5 +
6 + export function getAppsConnectModalTriggerObject() {
7 +   return {subject: modalTriggerSubject, observable: modalTriggerSubject.asObservable()};
8 + }
9 +

```

```
10 + export function getTransactionModalTriggerObject() {
11 +   return {subject: transactionModalTriggerSubject, observable: transactionModalTriggerSubject.asObservable()};
12 + }
```

```
✓ 77 ━━━━━━ app/tonconnect/models.ts [ ]
...
...
...
1 + import (AuthRequestBody, TonLoginClient) from "@tonapps/tonlogin-client";
2 + import (ConnectReplyBuilder) from "./ConnectReplyBuilder";
3 + import (ConnectItemReply) from "@tonconnect/protocol";
4 +
5 + export interface IConnectQRQuery {
6 +   v: string;
7 +   r: string;
8 +   id: string;
9 + }
10 +
11 + export interface DAppManifest {
12 +   url: string;
13 +   name: string;
14 +   iconUrl?: string;
15 +   termsOfUseUrl?: string;
16 +   privacyPolicyUrl?: string;
17 + }
18 +
19 + export interface TonConnectResponseModel {
20 +   address: string;
21 +   replyItems: ConnectItemReply[];
22 + }
23 +
24 + export type TonConnectModel = {
25 +   protocolVersion: 2;
26 +   manifest: DAppManifest;
27 +   replyBuilder: ConnectReplyBuilder;
28 +   requestPromise: (
29 +     resolve: (response: TonConnectResponseModel) => void;
30 +     reject: () => void;
31 +   );
32 +   hideImmediately: boolean;
33 + };
34 +
35 + export interface SignRawMessage {
36 +   address: string;
37 +   amount: string; // (decimal string): number of nanocoins to send.
38 +   payload: string; // (string base64, optional): raw one-cell Boc encoded in Base64.
39 +   stateInit: string; // (string base64, optional): raw once-cell Boc encoded in Base64.
40 + }
41 +
42 + export type TransactionModalParams = {
43 +   params: SignRawParams,
44 +   options: TxBodyOptions,
45 +   onsuccess?: (boc: string) => void,
46 +   ondismiss?: () => void,
47 + } | null;
48 +
49 + export type SignRawParams = {
50 +   source: string;
51 +   valid_until: number;
52 +   messages: SignRawMessage[];
53 + };
54 +
55 + export type TxBodyOptions = Omit<TxRequestBody, 'params' | 'type'>;
56 +
57 + export type TxRequestBody<TParams = TParams> = {
58 +   type: TxTypes;
59 +   expires_sec: number;
60 +   response_options: TxResponseOptions;
61 +   params: TParams;
62 + };
63 +
64 + export type TxParams =
65 +   | SignRawParams
66 +   | SignRawParams;
67 +
68 + export type TxTypes =
69 +   | 'sign-raw-payload'
70 +   | 'deploy';
71 +
72 + export type TxResponseOptions = {
73 +   broadcast: boolean;
74 +   return_url?: string;
75 +   callback_url?: string;
76 +   ondone?: () => void;
77 + };
```

```
✓ 331 ━━━━━━ app/tonconnect/transactions/VerifyTransactionModal.tsx [ ]
...
...
...
1 + import React, {useState} from "react";
2 + import {Platform, Text, View} from "react-native";
3 + import Modal from "react-native-modal";
4 + import {Theme} from "../../Theme";
5 + import LottieView from "lottie-react-native";
6 + import {T} from "../../i18n";
7 + import {RoundButton} from "../../components/RoundButton";
8 + import {getTransactionModalTriggerObject} from "../../modals/trigger";
9 + import {base64ToCell, encodeBytes, getBarkModeValue} from "../../utils/misc";
10 + import {contractFromPublicKey} from "../../engine/contractFromPublicKey";
11 + import {getCurrentAddress} from "../../storage/appState";
12 + import {loadWalletKeys, WalletKeys} from "../../storage/walletKeys";
13 + import {Configuration, SendApi} from "tonapi-sdk-js";
14 + import {fromNano, toNano} from "ton";
15 + import TonWeb from "tonweb";
16 + import BN from "bn.js";
17 + import {useEngine} from "../../engine/Engine";
18 + import {useItem} from "../../engine/persistence/PersistedItem";
19 + import {bearerToken, tonap10Endpoint} from "../../libs/Tonapi";
20 + import {warn} from "../../utils/log";
21 +
22 + type stateType = { action: string, message: string }; // 'requesting, rejected, approved'
23 + type verifyTransactionView = any;
24 +
25 + function VerifyTransactionModal() {
26 +   const engine = useEngine();
27 +   const account = useItem(engine.model.wallet(engine.address));
28 +   const acc = React.useMemo(() => getCurrentAddress(), []);
29 +   const balance = React.useMemo(() => {
30 +     return account.balance;
31 +   }, [account.balance]);
32 +   const transactionModalObject = getTransactionModalTriggerObject();
33 +   const [transactionObject, setTransactionObject] = useState<verifyTransactionView>(null);
34 +   const [currentAction, setCurrentAction] = useState<stateType>({action: 'requesting', message: ''}); // 'requesting, rejected, approved'
35 +   React.useEffect(() => {
36 +     const transactionModalSubscriber = transactionModalObject.observable.subscribe(async (transaction: any) => {
37 +       if (transaction) {
```

```

38 +         const messages = transaction.params.messages;
39 +         const target = new TonWeb.Address(messages[messages.length - 1].address).toString(true, true, false);
40 +         const amount = fromMano(messages[messages.length - 1].amount);
41 +         const resolver = transaction.tResolve;
42 +         const rejection = transaction.tReject;
43 +         setTransactionObject({target, amount, messages, resolve: resolver, reject: rejection});
44 +     }
45 +   );
46 +   return function cleanup() {
47 +     transactionModalSubscriber.unsubscribe();
48 +   }
49 + ), [transactionModalObject]);
50 +
51 + async function getWalletMethods(p_messages: any) {
52 +   const ton = new TonWeb();
53 +   const address = getCurrentAddress();
54 +   const wallet = await new ton.wallet.all['v4R2'](ton.provider, {
55 +     publicKey: address.publicKey,
56 +     wc: 0,
57 +   });
58 +   const seqno = await wallet.methods.seqno().call();
59 +   const sendMode = 3;
60 +   const signingMessage = (wallet as any).createSigningMessage(seqno);
61 +   let walletKeys: WalletKeys;
62 +   try {
63 +     walletKeys = await loadWalletKeys(acc.secretKeyEnc);
64 +   } catch (e) {
65 +     warn(e);
66 +     return;
67 +   }
68 +   const secretKey: Uint8Array = walletKeys.keyPair.secretKey;
69 +   const messages = p_messages.splice(0, 4);
70 +   for (let message of messages) {
71 +     const add = new TonWeb.utils.Address(message.address);
72 +     const order = TonWeb.Contract.createCommonMsgInfo(
73 +       TonWeb.Contract.createInternalMessageHeader(
74 +         new TonWeb.Address(add),
75 +         new TonWeb.utils.BN(message.amount),
76 +       ),
77 +       // @ts-ignore
78 +       base64ToCell(message.stateInit),
79 +       base64ToCell(message.payload),
80 +     );
81 +     signingMessage.bits.writeUInt8(sendMode);
82 +     signingMessage.refs.pushOrder();
83 +   }
84 +
85 +   return TonWeb.Contract.createMethod(
86 +     wallet.provider,
87 +     (wallet as any).createExternalMessage(
88 +       signingMessage,
89 +       secretKey,
90 +       seqno,
91 +       !secretKey,
92 +     ),
93 +   );
94 + }
95 +
96 + const doSend = React.useCallback(async () => {
97 +   let address = transactionObject?.target;
98 +   let value: BN;
99 +   let errorMsg: string | null = null;
100 +
101 +   if (address) {
102 +     try {
103 +       if (transactionObject?.amount) {
104 +         const validAmount = transactionObject?.amount.replace(',', '.');
105 +         value = toHano(validAmount);
106 +       } else {
107 +         errorMsg = t('transfer.error.invalidAmount');
108 +         setCurrentAction({action: 'rejected', message: errorMsg});
109 +         return;
110 +       }
111 +     } catch (e) {
112 +       errorMsg = t('transfer.error.invalidAmount');
113 +       setCurrentAction({action: 'rejected', message: errorMsg});
114 +       return;
115 +     }
116 +
117 +     // Load contract
118 +     const contract = await contractFromPublicKey(acc.publicKey);
119 +
120 +     // Check if same address
121 +     if (address && address === (contract.address.toFriendly({testOnly: false}))) {
122 +       errorMsg = t('transfer.error.sendingToYourself');
123 +     }
124 +
125 +     // Check amount
126 +     if (!value.eq(balance) && balance.lt(value)) {
127 +       errorMsg = t('transfer.error.insufficientAmountTitle'), t('transfer.error.notEnoughCoins');
128 +     }
129 +     if (value.eq(new BN(0))) {
130 +       errorMsg = t('transfer.error.zeroCoins');
131 +     }
132 +
133 +     if (errorMsg) {
134 +       setCurrentAction({action: 'rejected', message: errorMsg});
135 +       transactionObject.reject();
136 +       return;
137 +     } else {
138 +       const messages = transactionObject.messages;
139 +       const methods = await getWalletMethods(messages);
140 +       if (methods) {
141 +         const queryMsg = await methods.getQuery();
142 +         const boc = encodeBytes(await queryMsg.toBoc(false));
143 +         const response = await sendAPI().sendBoc({sendBocRequest: {boc}});
144 +         transactionObject.response(response);
145 +         setCurrentAction({action: 'approved', message: 'Transaction approved!'});
146 +         return response;
147 +       } else {
148 +         setCurrentAction({action: 'rejected', message: 'Something went wrong.\nPlease try again.'});
149 +         transactionObject.reject();
150 +         return;
151 +       }
152 +     }
153 +   } else {
154 +     setCurrentAction({action: 'rejected', message: t('transfer.error.invalidAddress')});
155 +     transactionObject.reject();
156 +     return;
157 +   }
158 + }, [transactionObject, acc]);
159 + function sendAPI() {
160 +   return new SendApi(
161 +     new Configuration({
162

```

```

164 +         basePath: tonapIEndpoint,
165 +         headers: {
166 +             Authorization: `Bearer ${bearerToken}`,
167 +         },
168 +     ),
169 +   )
170 +
171 +
172 +   const cancelRequest = React.useCallback(async () => {
173 +     if (transactionObject) {
174 +       transactionObject.reject();
175 +       setCurrentAction({action: 'rejected', message: 'User cancelled the the request.'});
176 +     }
177 +   }, [transactionObject]);
178 +
179 +   const closeModal = React.useCallback(async () => {
180 +     setTransactionObject(null);
181 +     setCurrentAction({action: 'requesting', message: ''});
182 +   }, []);
183 +
184 +
185 +   return (
186 +     <Modal isVisible={!transactionObject}>
187 +       <View style={{width: '100%', backgroundColor: Theme.item, paddingHorizontal: 5, paddingVertical: 10}}>
188 +         <text style={{fontSize: 19, fontWeight: '600', color: Theme.textColor, textAlign: 'center', textAlignVertical: 'center'}}>${products.transactionRequest.title}</text>
189 +       </View>
190 +       <View style={{backgroundColor: Theme.item}}>
191 +         <View style={{
192 +           flexDirection: 'row',
193 +           width: '100%',
194 +           height: 145,
195 +           backgroundColor: Theme.background,
196 +           alignItems: 'center',
197 +           justifyContent: 'space-around'
198 +         }}>
199 +           <Image style={{width: '100%', alignItems: 'center', justifyContent: 'center'}}>
200 +             <LottieView
201 +               source={require('../..../assets/animations/transaction-waiting.json')}
202 +               style={{width: 140}}
203 +               autoPlay={true}
204 +             />
205 +           </Image>
206 +           <Text>${currentAction.action === 'requesting' && (
207 +             <View style={{
208 +               flexDirection: 'column',
209 +               marginTop: 5,
210 +               justifyContent: 'space-evenly',
211 +               alignItems: 'stretch'
212 +             }}>
213 +               <View style={{
214 +                 background: Theme.lightGreen,
215 +                 padding: 10,
216 +                 margin: 5
217 +               }}>
218 +                 <Text style={{fontSize: 16, textAlign: 'center', color: Theme.textColor, fontWeight: '400'}}>${t('common.amount')}: ${transactionObject?.amount} TON</Text>
219 +               </View>
220 +             )}>
221 +             ${transactionObject?.target}
222 +           </Text>
223 +           <Text style={{
224 +             fontSize: 16,
225 +             textAlign: 'center',
226 +             color: Theme.textColor,
227 +             fontVariant: ['tabular-nums'],
228 +             fontFamily: Platform.OS === 'ios' ? 'Menlo-Bold' : 'monospace'
229 +           }}>
230 +             ${t('common.confirm')}
231 +           </Text>
232 +           <Text style={{
233 +             fontSize: 16,
234 +             textAlign: 'center',
235 +             color: Theme.textColor,
236 +             fontVariant: ['tabular-nums'],
237 +             fontFamily: Platform.OS === 'ios' ? 'Menlo-Bold' : 'monospace'
238 +           }}>
239 +             ${transactionObject?.target}
240 +           </Text>
241 +           <View style={{paddingHorizontal: 16, paddingVertical: 16, margin: 10}}>
242 +             <RoundButton
243 +               title={t('common.confirm')}
244 +               display={(getDarkModeValue()) ? 'default' : 'outline'}
245 +               action={doSend}
246 +             />
247 +             <RoundButton
248 +               title={t('common.cancel')}
249 +               display={(getDarkModeValue()) ? 'outline' : 'default'}
250 +               style={{margin: 10}}
251 +               action={cancelRequest}
252 +             />
253 +           </View>
254 +         </View>
255 +       </View>
256 +     </Modal>
257 +   );
258 +   const doSend = () => {
259 +     if (transactionObject) {
260 +       transactionObject.resolve();
261 +     }
262 +   };
263 +
264 +
265 +
266 +
267 +   if (currentAction.action === 'approved' && (
268 +     <View style={{
269 +       flexGrow: 1,
270 +       width: '100%',
271 +       alignItems: 'center',
272 +       alignContent: 'center',
273 +       padding: 20
274 +     }}>
275 +       <Image style={{width: 140, autoPlay: true, loop: true}}>
276 +         <LottieView
277 +           source={require('../..../assets/animations/green-check.json')}
278 +           style={{width: 140}}
279 +           autoPlay={true}
280 +           loop={true}
281 +         />
282 +         <Text style={{fontSize: 16, fontWeight: '600', color: Theme.primeGreen, textAlign: 'center'}}>${currentAction.message}</Text>
283 +       </Image>
284 +       <RoundButton
285 +         title={t('common.close')}
286 +       />
287 +     </View>
288 +   );
289 + 
```

```

298 +             display={getDarkModeValue() ? 'outline' : 'default'}
299 +             style={{marginTop: 25, width: '60%'}}
300 +             action={closeModal}
301 +         )}
302 +     )}
303 +   )}
304 +   )}
305 +   )}
306 +   )}
307 +   )}
308 +   )}
309 +   )}
310 +   )}
311 +   )}
312 +   )}
313 +   )}
314 +   )}
315 +   )}
316 +   )}
317 +   )}
318 +   )}
319 +   )}
320 +   )}
321 +   )}
322 +   )}
323 +   )}
324 +   )}
325 +   )}
326 +   )}
327 +   )}
328 +   );
329 +   )}
330 +   )}
331 +   export default VerifyTransactionModal;

```

diff --git app/utils/SupportedDomains.ts app/utils/SupportedDomains.ts

```

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

```

diff --git app/utils/getTimeSec.ts app/utils/getTimeSec.ts

```

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

```

diff --git app/utils/misc.ts app/utils/misc.ts

```

@@ -2,9 +2,31 @@ import {storage} from "../../storage/storage";
 2   import {fetchPriceHistory} from "./engine/api/fetchPrice";
 3   import {getCurrentCurrencyObject, getTON2USDValue} from "./currency";
 4   import {formatDate} from "./dates";
 5 + import queryParser from 'query-string';
 6 + import {createHash} from 'crypto';
 7 + import {findConnectedAppByClientSessionId, getAllConnections, IConnectedApp, IConnectedAppConnection,
 8 +   useConnectedAppStore
 9 + } from "../store";
10 + import {AppConfig} from "../AppConfig";
11 + import {getCurrentAddress} from "../storage/appState";
12 + import naclUtils from "tweetnacl-util";
13 + import {Cell} from "ton";
14 + import TonWeb from "tonweb";
15 +
16   export const minHideBalanceValue = 0.05;
17 - export let graphDetails = {data: [], hideIndices: [], dataAvailable: false, min: 0, max: 0, start: '', end: '', startPrice: null, endPrice: null,
18 -   fromCache: false, apicallended: false};
19 +   data: [],
20 +   hideIndices: [],
21 +   dataAvailable: false,
22 +   min: 0,
23 +   max: 0,
24 +   start: '',
25 +   end: '',
26 +   startPrice: null,
27 +   endPrice: null,
28 +   fromCache: false,
29 +   apicallended: false
30 +
31   export function setHideBalanceValue(value: boolean) {
32     storage.set('settings:hideSmallBalances', value ? 'true' : 'false');
33   }
34 +
35   @@ -128,7 +142,13 @@ export function parseHistoryData(history: any) {
36     graphDataArr.push(parseFloat(price.toString().substring(0, 4)));
37     graphDataArr.push(formatDate(new Date().getTime() / 1000));
38   }
39 -   return {data: graphDataArr, date: graphDateArr, hideIndices: min: Math.min.apply(Math, graphDataArr), max: Math.max.apply(Math, graphDataArr)};
40 +   return {
41 +     data: graphDataArr,
42 +     date: graphDateArr,
43 +     hideIndices,
44 +     min: Math.min.apply(Math, graphDataArr),
45 +     max: Math.max.apply(Math, graphDataArr)
46 +   };
47 }
48 +
49   export function getPriceHistory() {
50     @@ -143,11 +171,35 @@ export function getPriceHistory() {
51       );
52       resolve(graphDetails);
53     } else {
54 -       graphDetails = {data: [], hideIndices: [], dataAvailable: false, min: 0, max: 0, start: '', end: '', startPrice: null, endPrice: null,
55 -         fromCache: false, apicallended: true};
56 +       graphDetails = {
57 +         data: [],
58 +         hideIndices: []
59 +       };
60     }
61   }

```

```

176 +         hideIndices: [],
177 +         dataAvailable: false,
178 +         min: 0,
179 +         max: 0,
180 +         start: '',
181 +         end: '',
182 +         startPrice: null,
183 +         endPrice: null,
184 +         fromCache: false,
185 +         apiCallEnded: true
186 +       );
187 +     resolve(graphDetails);
188   }
189 }, (err) => {
190 -   graphDetails = { data: [], hideIndices: [], dataAvailable: false, min: 0, max: 0, start: '', end: '', startPrice: null, endPrice: null, fromCache: false, apiCallEnded: true }; // Load cached results
191 +
192 +   graphDetails = {
193 +     data: [],
194 +     hideIndices: [],
195 +     dataAvailable: false,
196 +     min: 0,
197 +     max: 0,
198 +     start: '',
199 +     end: '',
200 +     startPrice: null,
201 +     endPrice: null,
202 +     fromCache: false,
203 +     apiCallEnded: true
204 +   }; // Load cached results
205 +   resolve(graphDetails);
206 +
207  @@ -157,4 +209,37 @@ export function getFreshGraphData() {
208    return graphDetails;
209  }
210
211
212 + export const generateAppHashFromUrl = (url: string) => {
213 +   // get url without query
214 +   const {url: parsedUrl} = queryParser.parseUrl(url);
215 +
216 +   // remove last slash if it exists
217 +   const fixedUrl = parsedUrl.replace(/\/$/, '');
218 +
219 +   const hash = createHash('sha256').update(Buffer.from(fixedUrl)).digest('hex');
220 +
221 +   return hash;
222 + };
223 +
224 + export const getConnectedAppsReference = () => {
225 +   const {address} = getCurrentAddress();
226 +   const initialConnections = getAllConnections(
227 +     useConnectedAppsStore.getState(),
228 +     address.toFriendly({testOnly: AppConfig.isTestnet}),
229 +   );
230 +
231 +   if (initialConnections.length > 0) {
232 +     const apps: {connectedApp: IConnectedApp | null; connection: IConnectedAppConnection | null;}[] = [];
233 +     initialConnections.forEach((connection: any) => {
234 +       apps.push(findConnectedAppByClientId(connection.clientSessionId));
235 +     });
236 +   }
237 +   return apps;
238 + }
239 +
240 + export function Utf8ArrayToString(array: Uint8Array): string {
241 +   let out = '';
242 +   let len = array.length;
243 +   let i = 0;
244 +   let c: any = null;
245 +   while (i < len) {
246 +     c = array[i];
247 +     switch (c >> 4) {
248 +       case 0:
249 +       case 1:
250 +       case 2:
251 +       case 3:
252 +       case 4:
253 +       case 5:
254 +       case 6:
255 +       case 7:
256 +         // 0xxxxx
257 +         out += String.fromCharCode(c);
258 +         break;
259 +       case 12:
260 +       case 13:
261 +         // 110x xxxx 10xx xxxx
262 +         let char = array[i+1];
263 +         out += String.fromCharCode(((c & 0x1f) << 6) | (char & 0x3f));
264 +         break;
265 +       case 14:
266 +         // 1110 xxxx 10xx xxxx
267 +         let a = array[i+1];
268 +         let b = array[i+2];
269 +         out += String.fromCharCode(
270 +           ((c & 0x0f) << 12) | ((a & 0x3f) << 6) | ((b & 0x3f) << 0),
271 +         );
272 +         break;
273 +       }
274 +     }
275 +   return out;
276 + }
277 +
278 + export function encodeBytes(bytes: Uint8Array) {
279 +   return neclUtils.encodeBase64(bytes);
280 + }
281 +
282 + export function base64ToCell(base64: string): Cell | null {
283 +   if (base64) {
284 +     const bytes = new Uint8Array(Buffer.from(base64, 'base64'));
285 +     // @ts-ignore
286 +     return TonWeb.boc.cell.oneFromBoc(bytes);
287 +   } else {
288 +     return null;
289 +   }
290 + }
291 +
292 + export let walletStateInit: string = '';
293 +
294 + export async function setWalletStateInit () {
295 +   const ton = new TonWeb();
296 +   const address = getCurrentAddress();
297 +   const wallet = await new ton.wallet.all['v4R2'](ton.provider, {
298 +     publicKey: address.publicKey,
299 +     wc: 0,
300 +   });
301 +   const { stateInit } = await wallet.createStateInit();

```

```

302 +     walletStateInit = ton.utils.bytesToBase64(await stateinit.toBoc(false));
303 +
304 +
160 305 export type AddressContact = { name: string, fields: { key: string, value: string | null | undefined }[] } | null;

```

(*)

```

  ✓ 37 ████ app/utils/resolveUrl.ts [ ]
    @@ -15,6 +15,9 @@ type ResolvedUrl = {
15 15     type: 'connect',
16 16     session: string,
17 17     endpoint: string | null
18 + } | (
19 +   type: 'ton-connect',
20 +   query: any
18 21   ) | null;
19 22
20 23 export function resolveUrl(src: string): ResolvedUrl {
    @@ -73,20 +76,10 @@ export function resolveUrl(src: string): ResolvedUrl {
73 76     }
74 77
75 78     // ton url connect
76 -     if ((url.protocol.toLowerCase() === 'ton' || url.protocol.toLowerCase() === 'ton-test') && url.host.toLowerCase() === 'connect' &&
77 -       url.pathname.startsWith('/')) {
78 -         let session = url.pathname.slice(1);
79 -         let endpoint: string | null = null;
80 -         if (url.query) {
81 -           for (let key in url.query) {
82 -             if (key.toLowerCase() === 'endpoint') {
83 -               endpoint = url.query[key];
84 -             }
85 -           }
86 -         }
87 -         if (((url.protocol.toLowerCase() === 'ton' || url.protocol.toLowerCase() === 'ton-test') && url.host.toLowerCase() === 'ton-connect') ||
88 -             (url.protocol.toLowerCase() === 'tc')) {
89 -           return {
90 -             type: 'connect',
91 -             session,
92 -             endpoint
93 -           }
94 93         }
95 94       }
96 95     }
97 96     // HTTP(s) Sign Url
163 156 + // HTTP(s) Sign Url For ton-connect
164 157   if ((url.protocol.toLowerCase() === 'http:' || url.protocol.toLowerCase() === 'https:')
165 158     && (SupportedDomains.find(d => d === url.host.toLowerCase())))
166 -     && (url.pathname.toLowerCase().startsWith('/connect/')) {
167 -       let session = url.pathname.slice('/connect/'.length);
168 -       let endpoint: string | null = null;
169 -       if (url.query) {
170 -         for (let key in url.query) {
171 -           if (key.toLowerCase() === 'endpoint') {
172 -             endpoint = url.query[key];
173 -           }
174 -         }
175 -       }
159 +     && (url.pathname.toLowerCase().includes('ton-connect')) {
176 160       return {
177 -         type: 'connect',
178 -         session,
179 -         endpoint
161 +       type: 'ton-connect',
162 +       query: url.query
180 163     }
181 164   }
182 165

```

(*)

```

  ✓ 1 ████ assets/animations/connection.json [ ]
    @@ -0,0 +1 @@
1 + {"nm":"steps file","mn":"","layers":[{"ty":4,"nm":"Shape Layer
7","mn":"","sr":1,"st":0,"op":0,"ip":0,"hd":false,"cl":"","in":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":0,"k":0,"o":0
["-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,"k":0},,"o":0
["a":0,"k":100,"ix":11},,"r":["a":0,"k":0,"ix":10)},,"ef":[],,"shapes":[{"ty":4,"gr":1,"bm":0,"cl":1,"in":1,"hd":false,"mn":1,"nm":1,"Ellipse
1,"ix":1,"cx":2,"mp":3,"it":1,"iy":1,"el":1,"be":0,"cl":1,"in":1,"hd":false,"mn":1,"nm":1,"Ellipse Path 1","d":1,"p":["a":0,"k":0
[0,0],,"ix":3}),"s":{"a":0,"k":21,[432,21,432]},,"ix":2}},{("ty":4,"st":0,"bm":0,"cl":1,"in":1,"hd":false,"mn":1,"nm":1,"Stroke
1,"ic":1,"l":1,"m":4,"o":0,"k":100,"ix":4}),"w":["a":0,"k":0,"ix":5}),"d":1,"p":["a":0,"k":0,9320,0.33331],"ix":3};,"r":1,,"o":0
["a":0,"k":100,"ix":3}},{"ty":4,"tr":0,"c":0,"k":0,"ix":1}),"s":{"a":0,"k":0,"k":0,"o":0
["-121.284,1.716],"ix":2}},{"ty":4,"tr":0,"c":0,"k":0,"ix":6}),"r":["a":0,"k":100,"ix":7}]},,"ind":1};,"ty":4,"rm":1,"shape Layer
6,"mn":"","sr":1,"st":0,"op":0,"ip":0,"hd":false,"cl":"","in":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":0,"k":0
[-122.944,0.782,0),"ix":1}],"s":{"a":0,"k":111.817,111.817,100),"ix":6}),"sk":{"a":0,"k":0,"k":0},,"p":["a":0,"k":(201.886,50,0),"ix":2};,"sa":{"a":0,"k":0,"k":0},,"o":0
["a":0,"k":100,"ix":11},,"r":["a":0,"k":0,"ix":10)}},,"ef":[],,"shapes":[{"ty":4,"gr":1,"bm":0,"cl":1,"in":1,"hd":false,"mn":1,"nm":1,"Ellipse
1,"ix":1,"cx":2,"mp":3,"it":1,"iy":1,"el":1,"be":0,"cl":1,"in":1,"hd":false,"mn":1,"nm":1,"Ellipse Path 1","d":1,"p":["a":0,"k":0
[0,0],,"ix":3}),"s":{"a":0,"k":68,[408,68,408]},,"ix":2}},{("ty":4,"st":0,"bm":0,"cl":1,"in":1,"hd":false,"mn":1,"nm":1,"Stroke
1,"ic":1,"l":1,"m":4,"o":0,"k":100,"ix":4}),"w":["a":0,"k":0,"ix":5}),"d":1,"p":["a":0,"k":0,9320,0.33331],"ix":3};,"r":1,,"o":0
["a":0,"k":100,"ix":3}},{"ty":4,"tr":0,"c":0,"k":0,"ix":1}),"s":{"a":0,"k":0,"k":0,"o":0
["-122.944,0.782,0],"ix":2}},{"ty":4,"tr":0,"c":0,"k":0,"ix":6}),"r":["a":0,"k":100,"ix":7}]},,"ind":2};,"ty":4,"rm":1,"shape Layer
5,"mn":"","sr":1,"st":0,"op":0,"ip":0,"hd":false,"cl":"","in":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":0,"k":0
[-121.284,1.716],"ix":1}],"s":{"a":0,"k":100,100,100),"ix":6}),"sk":{"a":0,"k":0,"k":0},,"p":["a":0,"k":(356.886,50,0),"ix":2};,"sa":{"a":0,"k":0,"k":0},,"o":0
["a":0,"k":100,"ix":11},,"r":["a":0,"k":0,"ix":10)}},,"ef":[],,"shapes":[{"ty":4,"gr":1,"bm":0,"cl":1,"in":1,"hd":false,"mn":1,"nm":1,"Ellipse
1,"ix":1,"cx":2,"mp":3,"it":1,"iy":1,"el":1,"be":0,"cl":1,"in":1,"hd":false,"mn":1,"nm":1,"Ellipse Path 1","d":1,"p":["a":0,"k":0
[0,0],,"ix":3}),"s":{"a":0,"k":21,[432,21,432]},,"ix":2}},{("ty":4,"st":0,"bm":0,"cl":1,"in":1,"hd":false,"mn":1,"nm":1,"Stroke
1,"ic":1,"l":1,"m":4,"o":0,"k":100,"ix":4}),"w":["a":0,"k":0,"ix":5}),"d":1,"p":["a":0,"k":0,9320,0.33331],"ix":3};,"r":1,,"o":0
["a":0,"k":100,"ix":3}},{"ty":4,"tr":0,"c":0,"k":0,"ix":1}),"s":{"a":0,"k":0,"k":0,"o":0
["-122.944,0.782,0],"ix":2}},{"ty":4,"tr":0,"c":0,"k":0,"ix":6}),"r":["a":0,"k":100,"ix":7}]},,"ind":3};,"ty":4,"rm":1,"shape Layer
4,"mn":"","sr":1,"st":0,"op":0,"ip":0,"hd":false,"cl":"","in":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":0,"k":0
[-121.284,1.716],"ix":1}],"s":{"a":0,"k":111.817,111.817,100),"ix":6}),"sk":{"a":0,"k":0,"k":0},,"p":["a":0,"k":(49,48,49,48),"ix":2};,"sa":{"a":0,"k":0,"k":0},,"o":0
["a":0,"k":100,"ix":11},,"r":["a":0,"k":0,"ix":10)}},,"ef":[],,"shapes":[{"ty":4,"gr":1,"bm":0,"cl":1,"in":1,"hd":false,"mn":1,"nm":1,"Ellipse
1,"ix":1,"cx":2,"mp":3,"it":1,"iy":1,"el":1,"be":0,"cl":1,"in":1,"hd":false,"mn":1,"nm":1,"Ellipse Path 1","d":1,"p":["a":0,"k":0
[0,0],,"ix":3}),"s":{"a":0,"k":68,[408,68,408]},,"ix":2}},{("ty":4,"st":0,"bm":0,"cl":1,"in":1,"hd":false,"mn":1,"nm":1,"Stroke
1,"ic":1,"l":1,"m":4,"o":0,"k":100,"ix":4}),"w":["a":0,"k":0,"ix":5}),"d":1,"p":["a":0,"k":0,9320,0.33331],"ix":3};,"r":1,,"o":0
["a":0,"k":100,"ix":3}},{"ty":4,"tr":0,"c":0,"k":0,"ix":1}),"s":{"a":0,"k":0,"k":0,"o":0
["-122.944,0.782,0],"ix":2}},{"ty":4,"tr":0,"c":0,"k":0,"ix":6}),"r":["a":0,"k":100,"ix":7}]},,"ind":4};,"ty":4,"rm":1,"shape Layer
3,"mn":"","sr":1,"st":0,"op":0,"ip":0,"hd":false,"cl":"","in":"","ddd":0,"bm":0,"hasMask":false,"ao":0,"ks":{"a":0,"k":0
[-121.284,1.716],"ix":1}],"s":{"a":0,"k":100,100,100),"ix":6}),"sk": {"a":0,"k":0,"k":0},,"p": ["a":0,"k":(41,886,50,0),"ix":2};,"sa": {"a":0,"k":0,"k":0},,"o":0
["a":0,"k":100,"ix":11},,"r": ["a":0,"k":0,"ix":10)}},,"ef":[],,"shapes":[{"ty":4,"gr":1,"bm":0,"cl":1,"in":1,"hd":false,"mn":1,"nm":1,"Ellipse
1,"ix":1,"cx":2,"mp":3,"it":1,"iy":1,"el":1,"be":0,"cl":1,"in":1,"hd":false,"mn":1,"nm":1,"Ellipse Path 1","d":1,"p":["a":0,"k":0
[0,0],,"ix":3}),"s": {"a":0,"k":21,[432,21,432]},,"ix":2}},{("ty":4,"st":0,"bm":0,"cl":1,"in":1,"hd":false,"mn":1,"nm":1,"Stroke
1,"ic":1,"l":1,"m":4,"o":0,"k":100,"ix":4}),"w": ["a":0,"k":0,"ix":5}),"d":1,"p": ["a":0,"k":0,9320,0.33331],"ix":3};,"r":1,,"o":0
["a":0,"k":100,"ix":3}},{"ty":4,"tr":0,"c":0,"k":0,"ix":1}),"s": {"a":0,"k":0,"k":0,"o":0
["-122.944,0.782,0],"ix":2}},{"ty":4,"tr":0,"c":0,"k":0,"ix":6}),"r": ["a":0,"k":100,"ix":7}]},,"ind":5};,"ty":4,"rm":1,"shape Layer

```



```
{"ty":"fl","bm":0,"cl":0,"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","o":("a":0,"k":([0,0]),"ix":1),"s":("a":0,"k":([77.596,77.596]),"ix":3),"sk":("a":0,"k":([0,1]),"ix":4),"p":("a":0,"k":([3.5,-183.5],["x":2]),"r":("a":0,"k":([0,1]),"sa":("a":0,"k":([0,1]),"ix":5)),"o":("a":0,"k":100,"ix":7))),["ind":5]),"ddd":0,"h":560,"w":560,"meta":("a":0,"k":0,"d":0,"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 ...
...
1  @@ -0,0 +1,4 @@
1 + /* eslint-disable no-undef */
2 + // Inject node globals into React Native global scope.
3 + global.Buffer = global.Buffer || require('buffer').Buffer;
4 + global.process = require('process');
...
```

```
... 4 index.js ...
...
1  @@ -1,5 +1,7 @@
1 - global.Buffer = global.Buffer || require('buffer').Buffer;
1 + import './global';
2 + import 'react-native-get-random-values';
2 + import 'react-native-url-polyfill/auto';
3 + import 'text-encoding-polyfill';
3 ...
4 6 // Disable warnings
5 7 import { LogBox } from 'react-native';
...
```

Load diff

Load diff

Load diff

Load diff

Load diff