Engineer

Solanaのモバイルウォレット内でdAppを開く

2026年8月10日

`@solana/wallet-adapter-react` で構築したReactのdAppは、パソコンのウォレットには問題なく接続します。しかしスマートフォンでは、同じ手順がうまくいきません。ウォレット内のブラウザーでdAppを開く必要があるのに、それを行うはずのディープリンクが、何も言わず失敗します。Backpackでは「アプリをダウンロード」のページになり、Solflareではアプリは開いてもサイトが開きません。どの変形も失敗するように見えました。問題を分解すると、一つの症状に見える、四つの別々の問題でした。

技術的な取り組みを自社のシステムに生かす

「Solanaのモバイルウォレット内でdAppを開く」で扱う技術的な取り組みは、次の変更を考える参考になります。お客様のソフトウェア、データ、インフラで起きる同様の挙動を調査できます。 読み込みが速く、スマートフォンでも使え、お客様がどのブラウザーを使っていても同じように動く画面を提供します。

サービスを見る お問い合わせ →

推論とその適用範囲をたどる

この技術ノートの推論と、その適用範囲をご覧ください。お客様の環境でも同じ診断が成り立つかを検証できます。 フルスタックのGIS地図開発を指揮 — DBからブラウザーまでのフルスタック地図開発

フルスタックのGIS地図開発を指揮 お問い合わせ →

四つの問題

- **Backpackのリンク形式が誤っていました。** 文書にある形式は `https://backpack.app/ul/v1/browse/<url>?ref=<ref>` だけです。対象URLをパスに置き、 `ref` を必須とするユニバーサルリンクです。推測した `backpack://ul/v1/browse?url=...` のような独自スキームは、アプリが登録するどのルートにも一致せず、ウォレットのインストールページに着きます。

- **Solflareにもユニバーサルリンクが必要です。**
`https://solflare.com/ul/v1/browse/<url>?ref=<ref>` を使い、単なる `solflare://` スキームは使いません。スキームだけでは、アプリを起動しても移動先を指定できない場合があります。まさに「アプリは開くが、サイトのタブは手動で開く必要がある」という状態です。
- **両方のパラメーターをエンコードする必要があります。** `url` はdAppの完全な絶対アドレス、`ref` は要求元のオリジンで、それぞれ `encodeURIComponent` を通します。対象URLにエンコードしていない `?` や `&` があると解析が崩れ、ウォレットはブラウザータブではなくホーム画面を開きます。
- **起動のきっかけも、リンク自体と同じくらい重要です。** ユニバーサルリンクがアプリを切り替えるのは、OSが信頼する移動操作のときだけです。アドレスバーに貼り付けても、意図的に何もしません。そのため、完全に正しいリンクもテスト中に「失敗」します。

文書にある形式

- Phantom： `https://phantom.app/ul/browse/<url>?ref=<ref>` 。この形式に `/v1` はありません。
- Solflare： `https://solflare.com/ul/v1/browse/<url>?ref=<ref>`
- Backpack： `https://backpack.app/ul/v1/browse/<url>?ref=<ref>`

三つとも、同じパターンで扱えます。

```
const WALLET_BROWSE = {
  phantom: (url, ref) =>
    `https://phantom.app/ul/browse/${url}?ref=${ref}`,
  solflare: (url, ref) =>
    `https://solflare.com/ul/v1/browse/${url}?ref=${ref}`,
  backpack: (url, ref) =>
    `https://backpack.app/ul/v1/browse/${url}?ref=${ref}`,
};

function walletBrowseLink(
  walletName,
  targetUrl = window.location.href,
) {
  const build = WALLET_BROWSE[walletName.toLowerCase()];
  if (!build) return null;
  return build(
    encodeURIComponent(targetUrl),
    encodeURIComponent(window.location.origin),
  );
}
```

iOSとAndroidが受け入れる方法でリンクを開く

- **事前計算した、本物のリンク要素を描画します。** 単純な `<a href={walletBrowseLink('phantom')}>` が、両方の環境で最も確実な起動方法です。
- **プログラムから行う必要がある場合**は、タップのハンドラー内で、同期的に `window.location.href` へ代入します。その前に `await` 、 `fetch` 、 `setTimeout` を挟みません。非同期処理の後では操作の文脈が失われ、iOSはウォレットのウェブサイトへ移動します。 `window.open` は使いません。
- **アドレスバーへの貼り付けでは、決してテストしません。** ユニバーサルリンクは、そこでは意図的に動作しません。リンクをタップするか、カメラでQRコードを読み取ってテストします。
- **メッセージアプリ内のウェブビューに注意します。** TelegramやInstagramのアプリ内ブラウザーでは、ユニバーサルリンクがしばしば処理されず、ウォレットの通常のウェブサイトが読み込まれます。ユーザーエージェントによる判定は、よくても推測にとどまります。そのため、「SafariまたはChromeで開いてから接続してください」という、見える回避方法も利用者に示します。

Androidでの、より根本的な修正

独自のディープリンクを組み立てる方法は、iOS向けの話です。Androidでは、Solana MobileのMobile Wallet Adapterにより、モバイルブラウザーで動くdAppから、インストール済みのウォレットアプリへ直接接続でき、アプリ内ブラウザーを経由する必要はありません。最近の `@solana/wallet-adapter-react` は、モバイルアダプターを自動登録するため、wallet-adapter関連パッケージの更新だけでAndroidの問題が解消する場合もあります。目標の構成は、AndroidではMobile Wallet Adapter、相当する仕組みをAppleが認めていないiOSでは閲覧用ユニバーサルリンクです。

実機での検証

- ウォレットをインストールした実機で、メッセージアプリではなくOSのブラウザーからリンクを開きます。
- 描画されたリンクをタップするか、QRコードを読み取ります。アドレスバーには貼り付けません。
- ウォレットが開き、そのアプリ内ブラウザーのタブにdAppが読み込まれることを確認します。失敗するのは、この後半です。

- ウォレットをインストールしていない状態でも繰り返します。ユニバーサルリンクは、ウォレットのウェブサイトに移動するはずです。インストール済みなのにそのページが出る場合は、リンクか起動方法がまだ誤っています。
- その後で、メッセージアプリ経由の動作をテストし、そこで失敗する場合は「ブラウザーで開く」案内を追加します。

出典

- Phantom：iOSとAndroidのディープリンク
- Solflare：Browseディープリンク
- Backpack：Browseディープリンク
- Solana Mobile：Mobile Wallet Adapter

すべてのブログ記事

変更の前に測る

業務の流れ、既存のインターフェース、変更を必要とする利用者をお知らせください。検証可能な納品内容と、お客様のチームが保守を続けられる引き継ぎ内容を定められます。

料金を見る お問い合わせ →

観測したことを相談する

ご自身のプロジェクトに即して「Solanaのモバイルウォレット内でdAppを開く」についてご相談ください。何が観測されたか、どのような結果が役立つかをお聞かせください。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/notes/solana-mobile-wallet-deeplinks/