

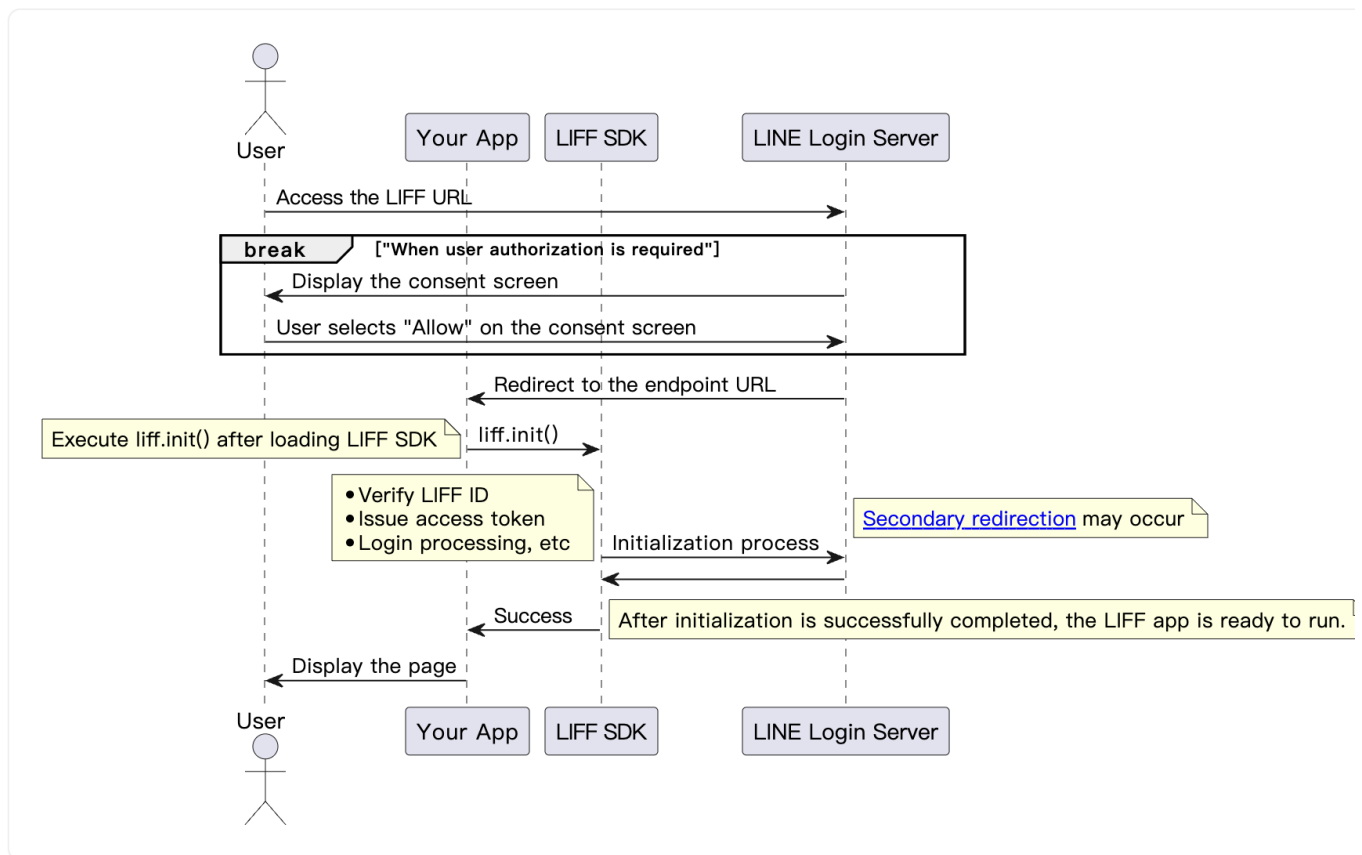
Line的 mini dapp 应用适配

Line 相关

测试登录频道地址

<https://liff.line.me/2006888231-KY98BBj8>

从用户在 LINE 应用程序上访问LIFF URL ②到 LIFF 应用程序初始化的过程如下：



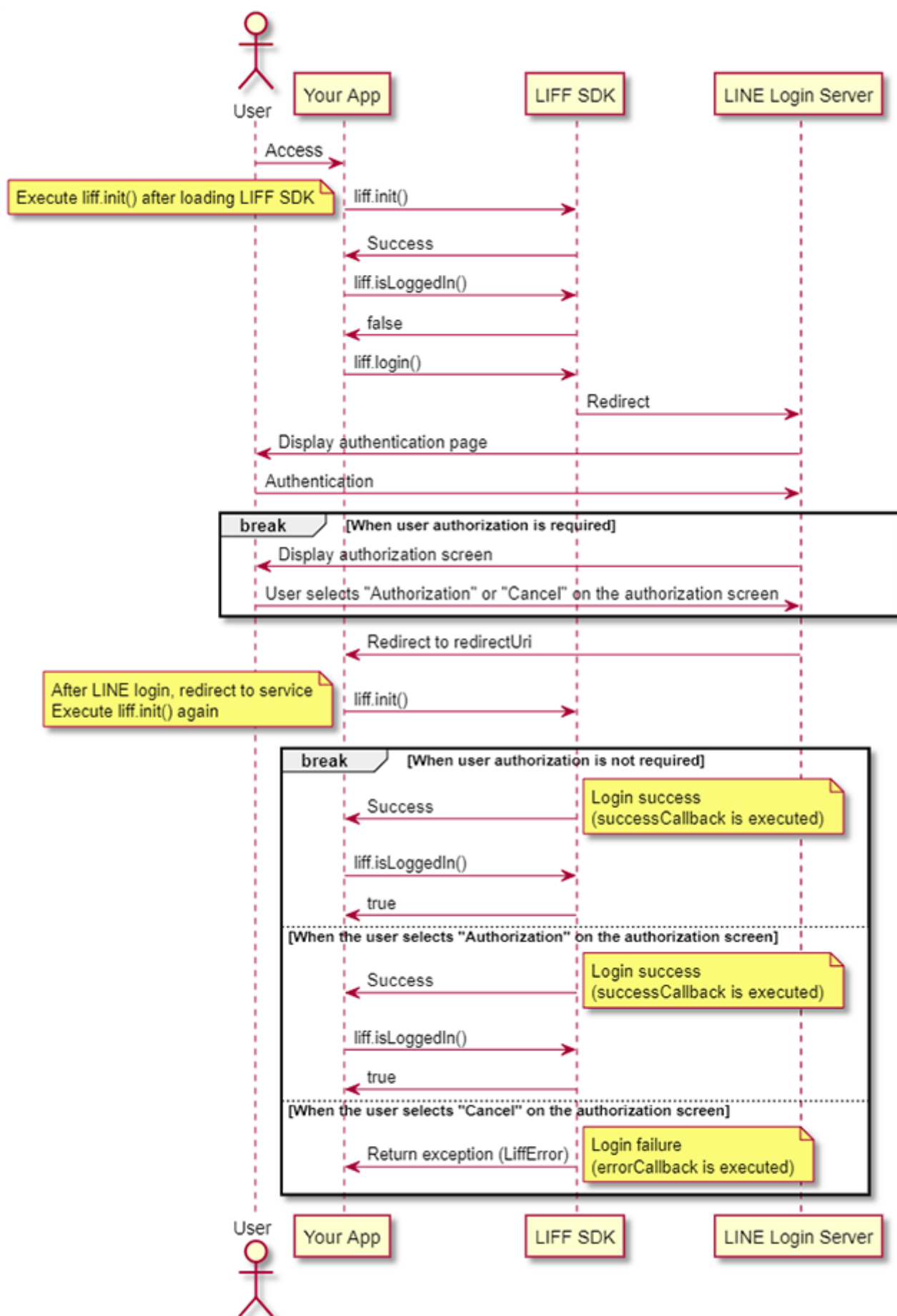
1. 启动参数适配，类似 TG 的 startapp，需要特定的适配，以区分来源和目标页面，邀请用户相关需要用到
2. 获取用户信息接口适配，登录后通过前端 JS 获取用户主资料信息的方法，不知道是否需要
3. 登录适配，需要有后端回调地址，看看能否获取用户头像和姓名，以及如何适配当前用户体系（需后端评估）

<https://developers.line.biz/en/docs/line-login/overview/#auth-method>

<https://developers.line.biz/en/services/line-login/>

登录版本 2.1 此版本可以处理基于 OAuth 2.0 授权码流程的登录请求（打开新窗口）。它还支持 OpenID Connect（打开新窗口）协议，并允许您使用 ID 令牌获取用户数据。

<https://developers.line.biz/en/docs/liff/developing-liff-apps/#to-use-line-login-in-external-browser>



通过用户 id 或者token 获取用户信息：

<https://developers.line.biz/en/docs/liff/using-user-profile/#page-title>

4. 应用内链接钱包部分，兼容 Kaia 钱包

```
1  networks: {  
2      kaia: {  
3          url: "https://public-en.node.kaia.io",  
4          accounts: [process.env.PRIVATE_KEY],  
5      },  
6      kairos_testnet: {  
7          url: "https://public-en-kairos.node.kaia.io",  
8          accounts: [process.env.PRIVATE_KEY],  
9      }  
10 }
```

- Kaia SDK
- <https://static.kaiawallet.io/js/dapp-portal-sdk.js>
- 测试水(24h): <https://www.kaia.io/faucet>

5. TG 的 Modal UI，看是否需要替换

6. 点击跳转链接，比如任务部分，需要将一些 tg 的操作平替

<https://developers.line.biz/en/docs/liff/opening-liff-app/#create-a-primary-redirect-url>

7. TGSDK 相关 @telegram-apps/sdk-react，均需移除或者寻找平替方法

不支持的信息

LIFF浏览器不支持 [外部浏览器](#) 支持的某些 Web 技术。LIFF 浏览器不支持的 Web 技术包括：

Web 技术	描述
主题颜色 Meta 标签	指定用户界面颜色的功能
下载属性	使用超链接下载资源而不是导航到资源的功能
添加到主屏幕 (A2HS)	允许用户将 Web 应用程序添加到用户设备主屏幕的功能。 在 LINE MINI App 中，可以使用操作按钮或方法中的 添加到主屏幕 将 LINE MINI App 的快捷方式添加到用户设备主屏幕。有关更多信息，请参阅LINE MINI App 文档中的将 LINE MINI App 的快捷方式添加到用户设备主屏幕。 <code>liff.createShortcutOnHomeScreen()</code>
服务工作者	在 Web 应用程序中启用离线支持、后台同步、推送通知等功能

上面列出的 Web 技术将来可能会被 LIFF 浏览器支持。

LIFF 浏览器是否支持除上述技术之外的其他 Web 技术均符合[WKWebView的规范](#) 和[Android WebView](#) 。有关更多信息，请参阅[我可以使用的...](#) 。

Playground

<https://liff-playground.netlify.app/>

Line 应用的示例代码

<https://developers.line.biz/en/docs/liff/trying-liff-app/#what-is-liff-starter-app>

```
1  import { useEffect, useState } from "react";
2  import liff from "@line/liff";
3  import "./App.css";
4
5  function App() {
6    const [message, setMessage] = useState("");
7    const [error, setError] = useState("");
8
9    useEffect(() => {
10      liff
11        .init({
12          liffId: import.meta.env.VITE_LIFF_ID
```

```
13     })
14     .then(() => {
15         setMessage("LIFF init succeeded.");
16         // TODO: 将所有构建文件上传到网络服务器, 例如 Netlify
17         if (!liff.isLoggedIn()) {
18             liff.login();
19         }
20     })
21     .catch((e: Error) => {
22         setMessage("LIFF init failed.");
23         setError(`${e}`);
24     });
25 });
26
27 return (
28     <div className="App">
29         <h1>create-liff-app</h1>
30         {message && <p>{message}</p>}
31         {error && (
32             <p>
33                 <code>{error}</code>
34             </p>
35         )}
36         <a
37             href="https://developers.line.biz/ja/docs/liff/"
38             target="_blank"
39             rel="noreferrer"
40         >
41             LIFF Documentation
42         </a>
43     </div>
44 );
45 }
46
47 export default App;
48
```