

# Android React Native

## Here's a step-by-step guide for implementing WhatsApp Login in a React Native app:

### Step 1: Set up the Deep Link

1. Open the `AndroidManifest.xml` file in your project's `android/app/src/main` directory.
2. Add the following code to the file to create a deep link:

```
<activity
  android:name="com.yourcompanyname.main.MainActivity"
  android:launchMode="singleTop"
  android:exported="true">
  <intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <category android:name="android.intent.category.DEFAULT" />
    <category android:name="android.intent.category.BROWSABLE" />
    <data
      android:host="YOUR_COMPANYNAME"
      android:scheme="otplless" />
    </intent-filter>
  </activity>
```

Note that you need to replace "YOURCOMPANYNAME" with your own name (write your first name without any spaces)

### Step 2: Create the Login Button

1. In your app, create a button with the text "Login with WhatsApp" and Add an `onPress` prop to the button, and set its value to a function that will handle the button press:

```
import React, { useCallback } from 'react';
import { Alert, Button, Linking } from 'react-native';

const LoginWithWhatsAppButton = () => {
```

```
const handlePress = useCallback(async () => {
  const url = 'YOUR_AUTHLINK';
  const supported = await Linking.canOpenURL(url);
  if (supported) {
    await Linking.openURL(url);
  } else {
    Alert.alert(`Don't know how to open this URL: ${url}`);
  }
}, []);

return (
  <Button
    title="Login with WhatsApp"
    onPress={handlePress}
  />
);
};

export default LoginWithWhatsAppButton;
```

Note that you need to define the `url` variable, which is the authlink that you'll use to authenticate the user. The url will be [https://your\\_companyname.authlink.me?redirectUri=otpless://your\\_companyname](https://your_companyname.authlink.me?redirectUri=otpless://your_companyname)

## Step 3: Listen for the Deep Link

1. In your app's main file (usually `App.js`), add the following code to listen for the deep link:

```
useEffect(() => {
  const linkingEvent = Linking.addEventListener('url', handleDeepLink);
  Linking.getInitialURL().then(url => {
    if (url) {
      handleDeepLink({url});
    }
  });
});
return () => {
  linkingEvent.remove();
};
}, [handleDeepLink]);

const handleDeepLink = async url => {
  console.log(url);
  // Extract the `waId` query parameter from the URL
```

```
const waId = new URLSearchParams(url.query).get('waId');  
  
};
```