



Sign in with Apple

**SICHER, SCHNELL UND EINFACH ZUM NEUEN
ACCOUNT**

ABOUT ME

- » Stefan Herold • @blackjacxxx 
- » Mobile Dev seit 2009
- » Seit 2017 bei ioki im Herzen Frankfurts

IOKI

- » Demand Responsive Transport
- » Leuchtturmpunkt mit HVV Hamburg
- » Whitelabel Passenger App • iOS & Android
- » Driver App • React Native
- » Backend • Ruby

SIGN IN WITH APPLE

- » Registrierung und Login
- » schnell, sicher und privat
- » kein Tracking durch Apple
- » App erhält:
 - » stabile, eindeutige userID
 - » Vor- und Nachname
 - » verifizierte Email Adresse

SIGN IN WITH APPLE

PLATFORM

iOS 13+

iPadOS 13+

tvOS 13+

watchOS 6+

macOS Catalina 10.15+

Web

Windows

Android

APPLE ID

iCloud Nutzer

beliebige Apple ID

SICHER

- » Kein Passwort
- » Zwei-Faktor-Authentisierung
- » Anti-Fraud (Glaubwürdigkeitsprüfung)
- » Jahrelang entwickelter Algorithmus
- » On-Device Machine Learning / Account History
- » 1-Bit-Info über Echtheit des Users
- » iOS only

PRIVAT

Eindeutige, zufällige Email
privaterelay.appleid.com

App sieht nur diese Email

- » Kommunikation mit genau einem Developer
- » Zwei-Wege-Kanal
- » Kein speichern von Emails
- » Weiterleitung deaktivierbar

The screenshot shows a mobile application interface for generating a temporary email address. At the top, it displays "00:00" and signal strength icons. Below the header, there's a back button labeled "Back" and a title "Checkmark". The main content area features a large green circular icon with a white checkmark. To its right, the word "Checkmark" is displayed above the name "Stefan Herold". A descriptive text block below states: "You created an account in this app on 04 November 2019 and shared your name and used 'Hide my email'." A "HIDE MY EMAIL" button is present. Another section shows "This app received" an email from "7pucxd78iq@privaterelay.appleid.com". A "Forward to" field contains "stefan.herold@gmail.com" with a green toggle switch next to it. A note explains that "Hide my email" allows apps to contact you without receiving your real email address, with a "Learn more" link. At the bottom, a red "Stop using Apple ID" button is visible, along with a note about future account creation.

00:00 1

Back Checkmark

Checkmark
Stefan Herold

You created an account in this app on 04 November 2019 and shared your name and used 'Hide my email'.

HIDE MY EMAIL

This app received
7pucxd78iq@privaterelay.appleid.com

Forward to
stefan.herold@gmail.com

'Hide my email' allows apps to contact you without receiving your real email address. [Learn more](#).

Stop using Apple ID

Next time that you use this app, you may be asked to create a new account.

WER'S BRAUCHT

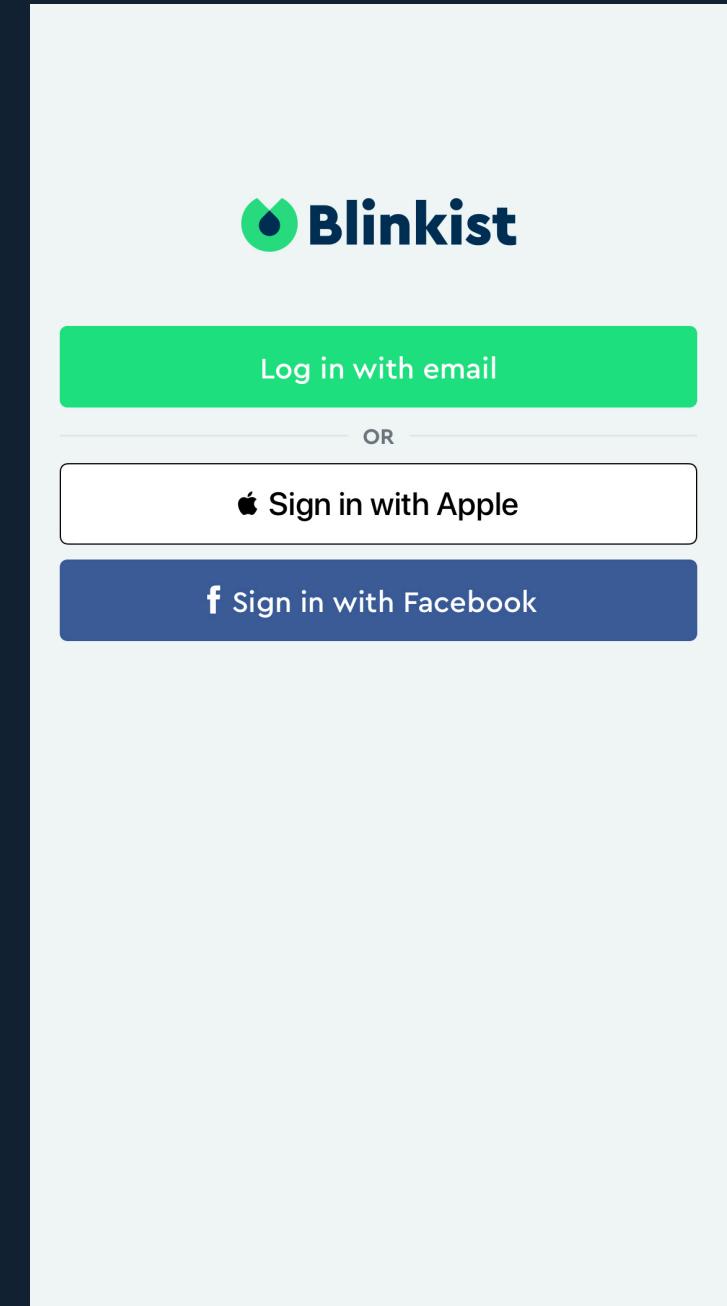
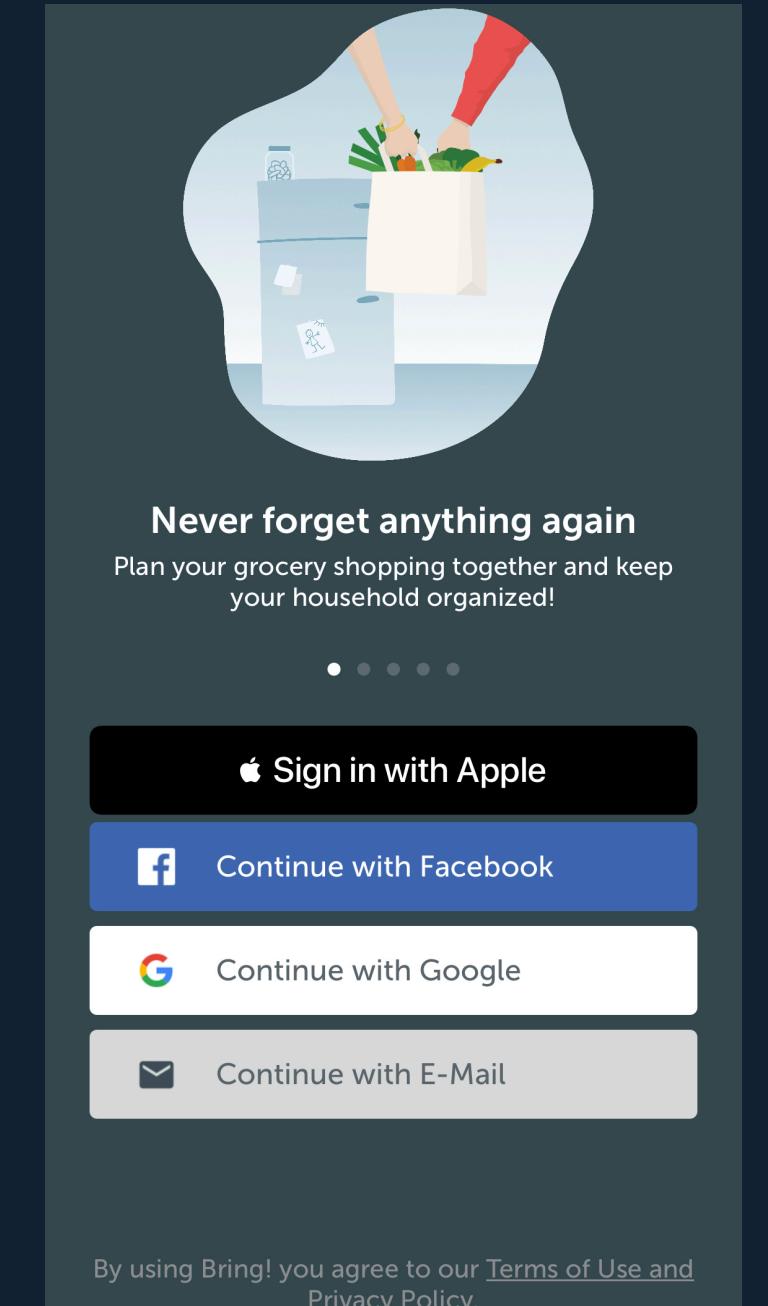
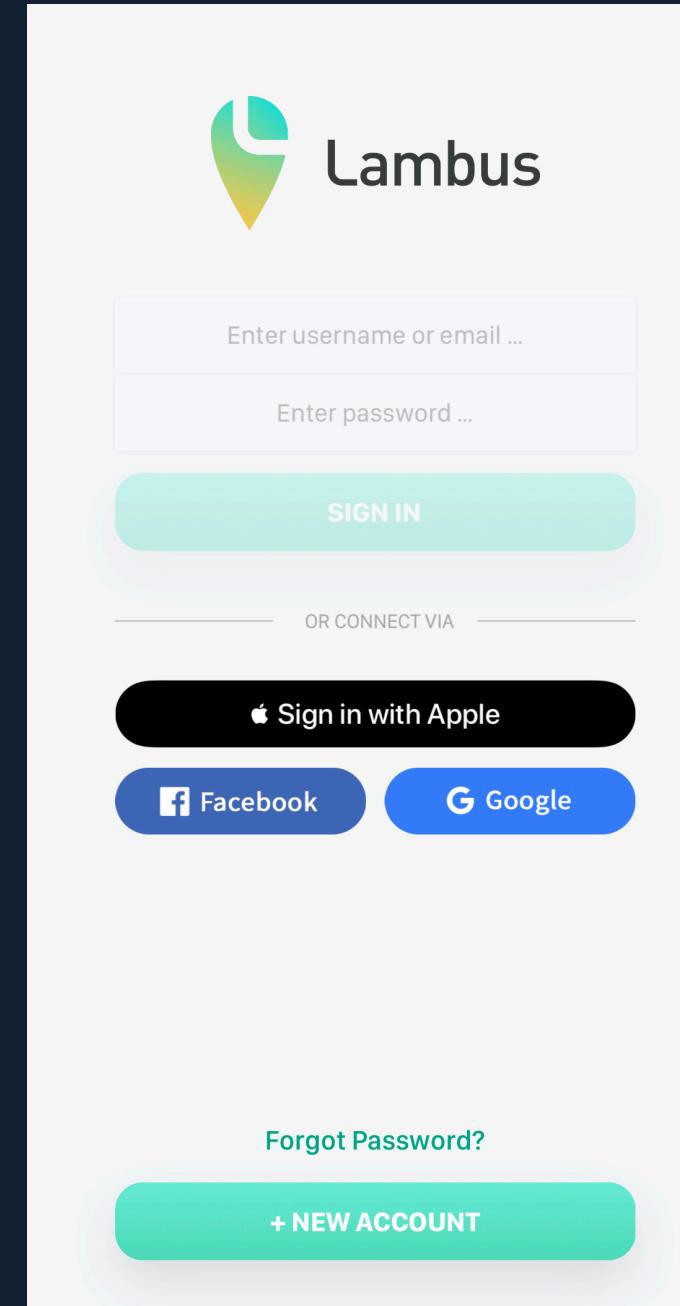
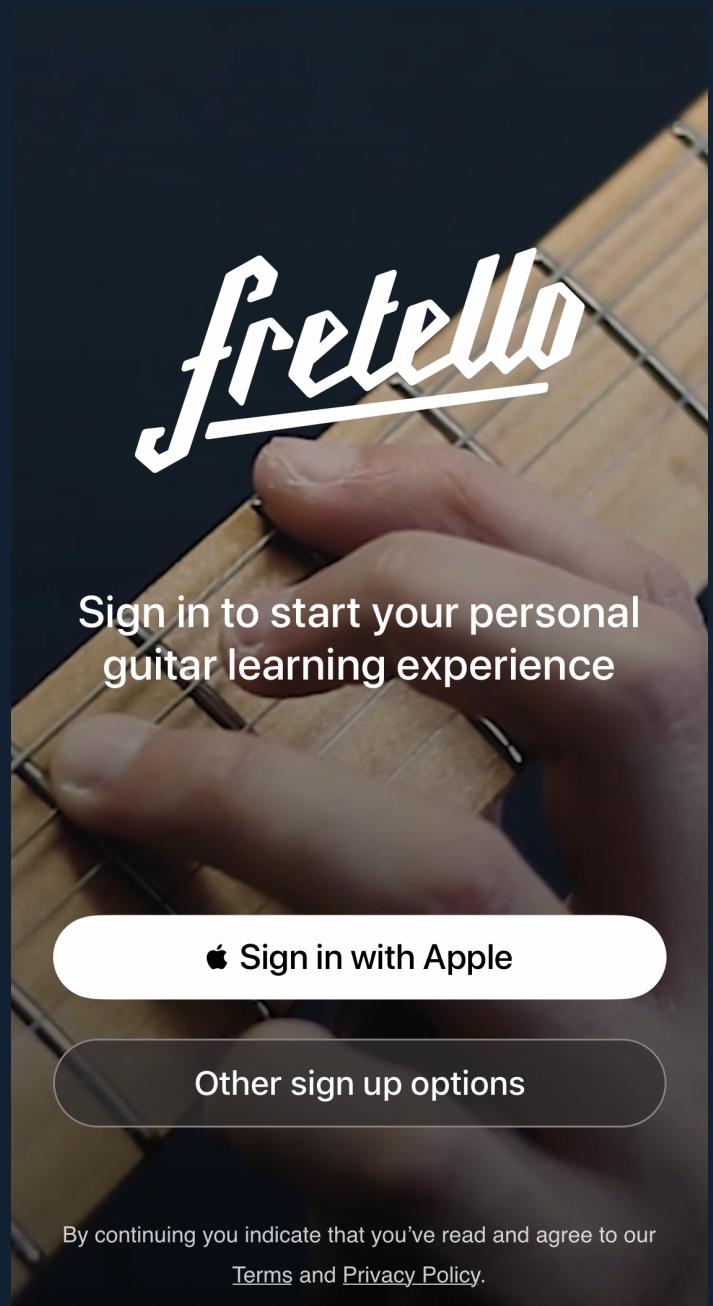
Apps die exklusiv third-party login service nutzen

- » Facebook Login
- » Google Sign-In
- » Sign in with Twitter
- » Sign In with Linked-In
- » Login with Amazon
- » WeChat Login

WER NICHT

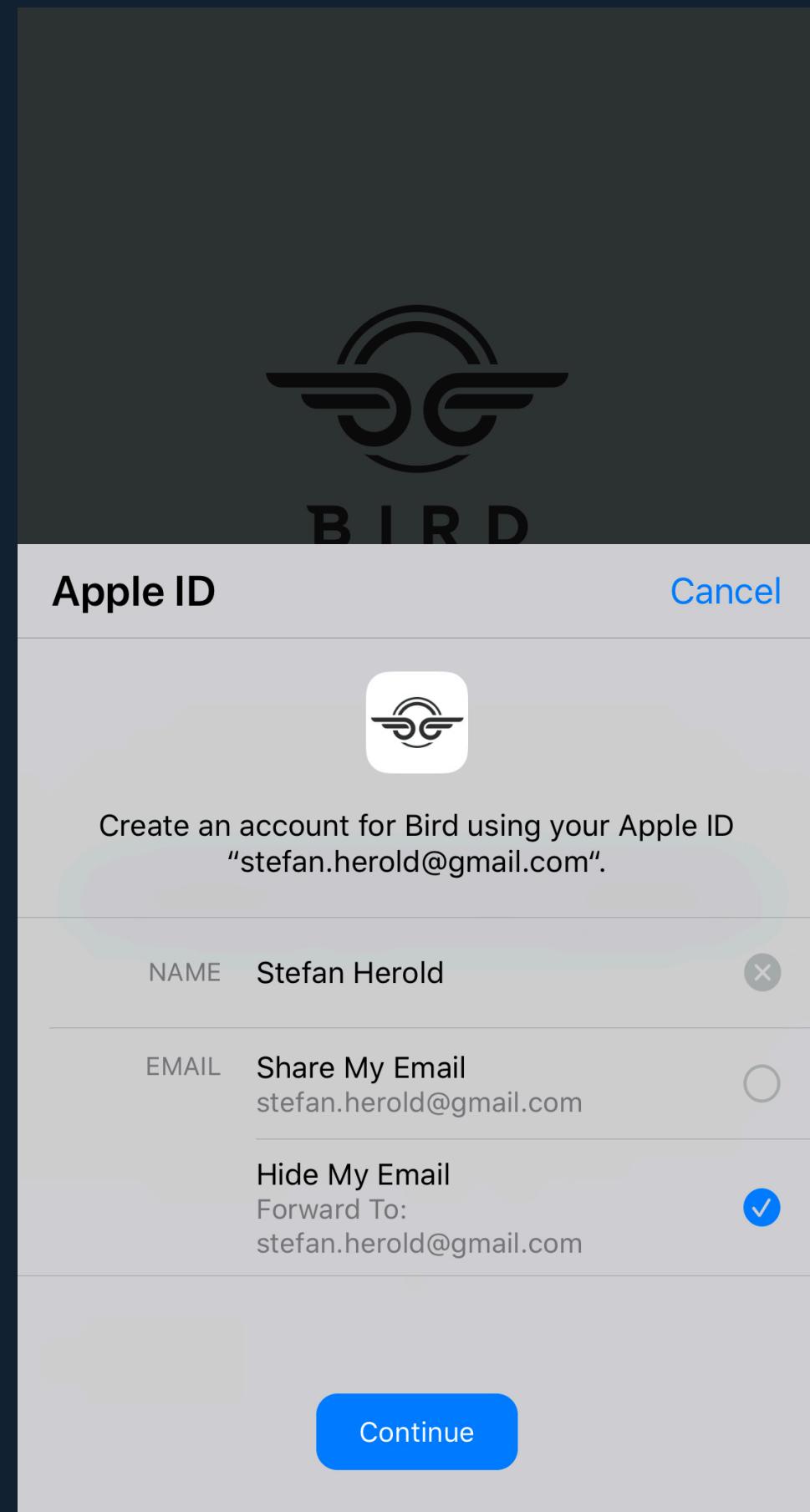
- » App nutzt firmeneigenes Login-System
- » App nutzt ausweisbasiertes Login-System
- » App ist Client für 3rd party / social service
- » Bildungs-, Enterprise- oder Business-App mit existierendem Firmen-Account

APPS



REGISTRIERUNG

APP	EMAIL	NAME
Bird	✓	✓
Lambus	✓	✓
Bring	✓	✓
Blinkist	✓	✗
Parcel	✗	✗



REGISTRIERUNG

Folgendes bekommen wir:

- » UserID • eindeutig, stabil über alle Geräte mit gleicher AppleID
- » Identity Token • Nutzerverifizierung
- » Auth Code • Refresh Token
- » Verifizierte Email - entfällt beim Onboarding
- » Vor- und Zuname als PersonNameComponents
- » Real User Indicator - Boolean: User / Unknown
- » Credential State - authorized, revoked, notFound

DEMO



BACKEND GOOD TO KNOW

- » Identity Token zur Verifizierung der userID
- » Token nur bei Registrierung - 10 min gültig
- » Login stellt nur userId zur Verfügung
- » OAuth Flow (Access-/Refresh-Token) notwendig¹

¹ <https://blog.curtisherbert.com/so-theyve-signed-in-with-apple-now-what>

ZUSAMMENFASSUNG

- » Überblick über Vor- / Nachteile
- » Was bedeuten Sicher & Privat
- » Theorie: Registrierung & Login
- » Praxis: Erweitern einer bestehenden App
- » Exkurs Backend: Fallstricke

VIELEN DANK FÜR'S ZUHÖREN



⚠ Demo App Code & Slides

<https://github.com/Blackjacx/SignInWithApple>

📺 Introducing Sign In with Apple - Session Video Notes

<https://github.com/Blackjacx/WWDC#introducing-sign-in-with-apple>

💻 Apple Docs

<https://developer.apple.com/sign-in-with-apple>

💡 Review Guidelines

<https://developer.apple.com/app-store/review/guidelines/#sign-in-with-apple>

🌐 REST API

<https://developer.apple.com/documentation/signinwithapplerestapi>

🐦 Twitter

@blackjacxxx

BACKUP

MEHR LINKS

- » Answers to your burning questions
<https://techcrunch.com/2019/06/07/answers-to-your-burning-questions-about-how-sign-in-with-apple-works>
- » Ray Wenderlich Tutorial mit SwiftUI
<https://www.raywenderlich.com/4875322-sign-in-with-apple-using-swiftui>
- » Erzeugen des Private Keys
<https://developer.apple.com/account/resources/authkeys/add>
- » Konfiguration erlaubter Email-Adressen
<https://developer.apple.com/account/resources/services/configure>
- » Sign in with Apple für Web
<https://developer.okta.com/blog/2019/06/04/what-the-heck-is-sign-in-with-apple>
- » 9To5Mac Artikel
<https://9to5mac.com/2019/10/15/how-to-use-sign-in-with-apple-iphone-ipad-mac>
- » Token Handling im Backend
<https://blog.curtisherbert.com/so-theyve-signed-in-with-apple-now-what>

REGISTRIERUNG

```
func didPressSignInWithApple(_ sender: UIButton) {  
  
    let provider = ASAuthorizationAppleIDProvider()  
    let request = provider.createRequest()  
    request.requestedScopes = [.email, .fullName] // optional - only request what's required  
  
    let controller = ASAuthorizationController(authorizationRequests: [request])  
    controller.delegate = self  
    controller.presentationContextProvider = self  
    controller.performRequests()  
}
```

LOGIN

Beim Appstart:

```
let provider = ASAuthorizationAppleIDProvider()  
  
provider.getCredentialState(forUserID: userId) { (state, error) in  
    // evaluate state  
}
```

State-Änderungen:

```
let name = ASAuthorizationAppleIDProvider.credentialRevokedNotification  
center.addObserver(forName: name, object: nil, queue: nil) { [weak self] _ in  
    self?.performSignOut()  
}
```

OAUTH

