

시빅(Civic)

해당 번역본은 이용자가 디지털자산 관련 정보를 편리하게 확인할 수 있도록
단순 참고용으로 번역된 것이며, 투자 권유를 목적으로 하지 않습니다.
그 내용이 정확하지 않을 수 있으므로 정확한 정보 습득을 위해서는
원문을 참고하시거나 원문 작성 주체 측에 문의하시기 바랍니다.
두나무는 해당 번역본 내용의 진실성 및 정확성을 보장하지 않고,
제공된 정보에 의한 투자 결과에 대하여 어떠한 책임도 지지 않으며,
투자 손실은 투자자에게 귀속됩니다.

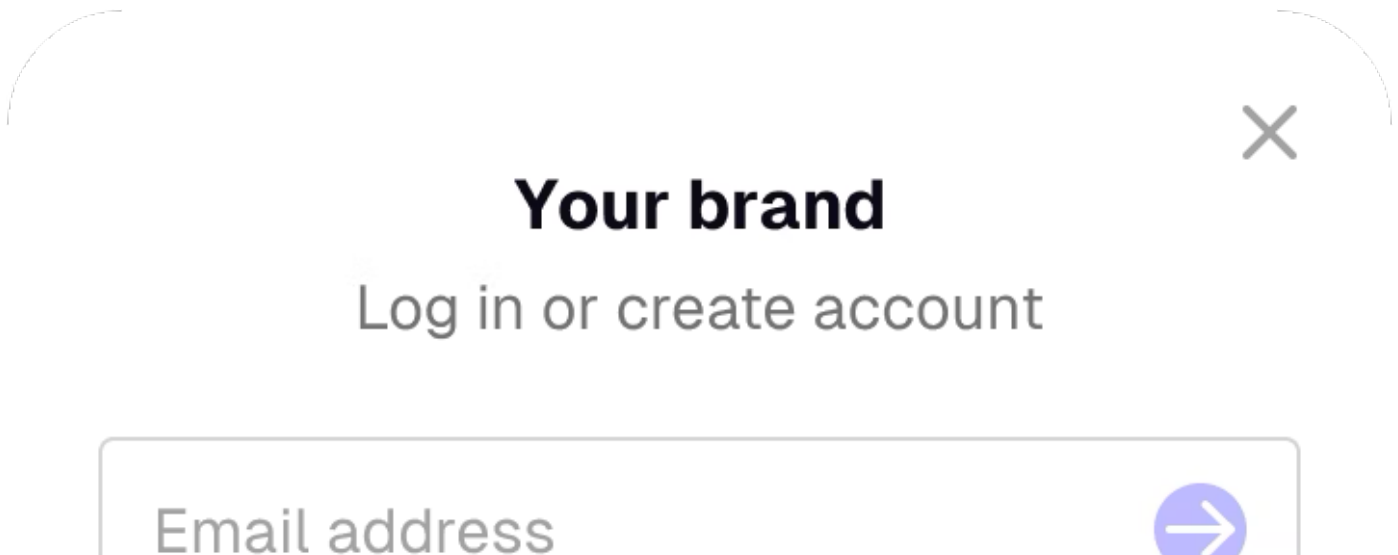
개요

Civic Auth

Civic Auth는 애플리케이션에 인증을 통합하기 위한 간단하고 유연하며 빠른 방법입니다. 익숙한 로그인 옵션을 활성화하면서 선택적으로 내장된 지갑을 제공하고, 사용자를 위한 블록체인 혜택을 활용할 수 있습니다.

Civic Auth란 무엇인가?

Civic Auth는 현재 **베타** 단계입니다. 경험을 개선하는 과정에서 때때로 변경이나 업데이트가 있을 수 있습니다.



OR



COMING SOON



Passkey



By continuing, you agree to our [Terms](#) and [Privacy Policy](#).

Powered by  CIVIC

통합 로그인

사용자는 이메일, Google 계정 또는 지갑을 사용하여 로그인할 수 있습니다. 복잡한 설정은 필요하지 않습니다.

적응형 온보딩

Auth는 모든 사용자에게 익숙한 로그인 경험을 제공합니다. 기존 지갑이나 통합된 내장 지갑을 지원합니다.

내장 지갑

사용자를 위해 자동으로 Web3 지갑을 생성합니다.

멀티체인 지원

Solana뿐 아니라 Base, Binance Smart Chain(BSC), Polygon, Arbitrum, Ethereum 및 기타 EVM 호환 체인을 지원합니다.

빠른 시작

Civic Auth에 가입하여 1분 이내에 auth.civic.com에서 Client ID를 받으세요.

The screenshot shows the Civic Auth dashboard. At the top, there's a header with the Civic logo, the user's email 'jane.smith@gmail.com's App', and the environment 'Production'. The left sidebar contains navigation links: Setup, Users, Account, Billing, Integration docs, Request marketing, Feedback, Community, and Log out. The main content area is titled 'Integration steps' and lists three steps: 1. Install the Civic SDK (code: `npm install @civic/auth`), 2. Copy your client ID (ID: `0396b122-b70b-43cc-9ca9-734032f7af28`), and 3. Add to your app (options: React, Next.js, Other). Below step 3, there's a code snippet for integrating Civic Auth into a React application.

```
import { CivicAuthProvider } from "@civic/auth/react";

function App({ children }) {
  return (
    <CivicProvider clientId="7f7fb8ec-944c-4c9d-84fa-cfc5c154ec7f"
      {children}
    </CivicProvider>
  );
}
```

Web3 지갑

사용자에게 Web3 지갑을 제공할 계획이라면 Civic Auth Web3 SDK를 사용할 수 있습니다.

Setup ▾

Integration steps

Login options

SSO connections

Domains

Application options

Users

Account

Billing

Integration docs

Request marketing

Feedback

Community

Log out

Require an email address



Users must verify an email address when connecting with a method that does not provide one. Civic Auth links user accounts with the same email address to optimize the user experience.

Block email subaddresses



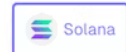
Users cannot create an account using an email address with a subaddress (IE: joe+1@example.com).

Enable embedded wallets



Issue an embedded wallet to your users using their social login and email address.

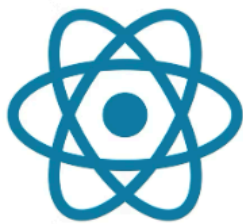
Select chains to support:



Save

사용법

애플리케이션에 Civic Auth를 통합하는 방법에 대한 지침을 얻으려면 프레임워크를 선택하세요.



React



Next.js



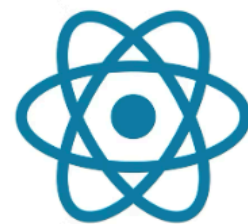
Vanilla JavaScript



Node.js



Python



React Native

다른 환경에서 OIDC 또는 OAuth 2.0을 준수하는 클라이언트 라이브러리를 사용하는 통합에 대해서는 [여기](#)를 참조하세요.

인증 가격

단 몇 줄의 코드로 원활한 인증을 제공합니다.

Civic Auth로 구축할 때는 예상치 못한 비용이 발생하지 않습니다.

우리의 가격 정책은 단순하고 투명하여 빌더들이 초기에 제품에 집중하고 확장하면서 복잡한 정책을 걱정하지 않도록 설계되었습니다.

애플리케이션에 가치를 더하는 활성 사용자에 대해서만 비용을 지불하면 됩니다 — 그리고 이는 **Auth 전용**으로 15,000 MAU를 초과한 경우 또는 **지갑을 사용하는 경우** 2,000 MAU를 초과한 경우에만 해당됩니다.

요금제 비교

인증 가격

Auth Only	무료	Build
비용	\$0	월 \$25
MAU	15,000 월간 활성 사용자(MAU) 포함	추가 MAU당 \$0.025 (처음 15,000 명은 무료!)
SSO 옵션	다중 SSO 옵션	포함
맞춤형 로그인 UI	드래그 앤 드롭 커스터마이징으로 원활한 브랜딩	포함
통합 속도	초고속: 몇 분 안에 설정 및 실행	포함
확장성	안전하고 신뢰할 수 있으며 현대적인 애플리케이션에 적합	포함

Web3 지갑 인증 가격

내장 지갑 인증

기능	무료	Build
비용	\$0	월 \$25
MAU	2,000 월간 활성 사용자(MAU) 포함	추가 MAU당 \$0.025 (처음 2,000 명은 무료!)
SSO 옵션	다중 SSO 옵션	포함
맞춤형 로그인 UI	드래그 앤 드롭 커스터마이징으로 원활한 브랜딩	포함
통합 속도	초고속: 몇 분 안에 설정 및 실행	포함
확장성	안전하고 신뢰할 수 있으며 현대적인 애플리케이션에 적합	포함

앱을 프로덕션으로 배포하기

실제 환경에서 앱을 사용하기 위해 따라야 하는 단계입니다.

Civic Auth Dashboard를 사용하여 앱을 설정한 후, 앱을 프로덕션으로 승격하기 위해 수행해야 할 단계는 다음과 같습니다.

SSO 자격 증명 구성

샌드박스 모드에서는 앱이 Google과 같은 SSO 공급자를 위해 Civic의 자격 증명을 사용합니다. 그러나 실제 앱에서는 선택한 공급자에 대해 자체 자격 증명을 제공해야 합니다.

각 공급자에 대해 OAuth 앱을 설정할 때 제공되는 `clientId`와 `clientSecret`을 설정해야 합니다.

또한 Civic Auth 도메인을 OAuth 화이트리스트 URL에 추가해야 합니다.

예: Google

다음 도메인을 추가해야 합니다:

`https://auth.civic.com` 및

`https://auth.civic.com/login/api/callback/google`

다른 공급자의 경우, 콜백 URL은 다른 접미사를 가집니다.

도메인 설정

앱이 귀하의 웹사이트에서만 사용 가능하도록 보장하기 위해, 웹사이트가 호스팅될 도메인을 추가해야 합니다. 최소한 하나의 도메인을 정의해야 하며, 보안이 적용된 https 사이트를 권장합니다.

참고: 테스트 중에는 localhost 도메인을 추가할 수 있지만, 앱을 프로덕션으로 배포할 때는 localhost 값을 제거하는 것이 권장됩니다. 그렇지 않으면 localhost 앱을 실행하는 누구든 귀하의 clientId를 사용할 수 있습니다.

결제 정보 추가

결제 정보는 Billing 페이지를 통해 추가해야 합니다. Auth Pricing에 대한 자세한 내용은 여기를 클릭하세요.

앱 변환 또는 복제

모든 단계를 완료한 후, 'Sandbox' 헤더를 클릭하고 '+ Launch to production' 옵션을 선택할 수 있습니다.

이제 기존 앱을 프로덕션으로 배포하는 두 가지 옵션이 있습니다: Convert와 Duplicate.

Convert

샌드박스 모드에서 사용한 clientId와 모든 설정을 동일하게 유지하려면 이 옵션을 선택하세요.

Duplicate

이 옵션은 모든 앱 설정을 새 앱으로 복사하며, 프로덕션 사이트에서 사용할 수 있는 새로운 clientId를 생성합니다.

암호화 지갑 활성화 (Web3 앱 전용)

사용자가 내장 지갑에 접근할 수 있도록 하려면, Crypto Wallets 대시보드 페이지에서 'Enable embedded wallets' 설정을 활성화해야 합니다.

로그인 옵션

개요

사용자가 앱에 로그인할 수 있도록 설정할 수 있는 로그인 옵션입니다.

Civic Auth Dashboard를 사용하여 앱의 다양한 로그인 옵션을 선택할 수 있습니다. 지원하고자 하는 공급자를 선택하고, 사용자에게 표시할 순서를 지정할 수 있습니다.

공급자 아이콘을 클릭하여 활성화 또는 비활성화할 수 있으며, 옵션을 위나 아래로 드래그하여 순서를 변경할 수 있습니다. 업데이트를 수행하는 즉시, 사용자가 보게 될 미리 보기를 확인할 수 있습니다.

The screenshot shows the Civic Auth Dashboard interface. At the top, there's a header with the Civic logo, the user email 'jane.smith@gmail.com's App', and the environment 'Production'. The left sidebar contains navigation links: Setup (with a dropdown), Integration steps (highlighted), Login options, SSO connections, Domains, Application options, Users, Account, Billing, Integration docs, Request marketing, Feedback, Community, and Log out. The main content area displays three steps for integration:

- 1 Install the Civic SDK**
Code: `npm install @civic/auth`
- 2 Copy your client ID**
Client ID: `0396b122-b70b-43cc-9ca9-734032f7af28`
- 3 Add to your app**
Frameworks: **React** (selected), Next.js, Other
Text: View the [docs](#) for more information.
Code example:

```
import { CivicAuthProvider } from "@civic/auth/react";

function App({ children }) {
  return (
    <CivicProvider clientId="7f7fb8ec-944c-4c9d-84fa-cfc5c154ec7f"
      {children}
    </CivicProvider>
  );
}
```

소셜 제공자 (OAuth)

현재 지원되는 소셜 제공자는 다음과 같습니다:

- Google
- Discord
- Github
- Facebook
- X
- Apple
- Microsoft (곧 지원 예정)

원하는 제공자가 보이지 않나요? 저희에게 알려주시면 검토해 보겠습니다!

이메일

사용자는 이메일로 전송된 OTP(일회용 PIN)를 입력하여 자신의 이메일 주소 소유권을 증명함으로써 로그인합니다.

계정 연결

사용자가 이메일 주소로 로그인할 때, OTP 코드가 검증되면 해당 이메일이 이미 Google 또는 Apple 계정과 함께 사용된 적이 있는지 확인합니다.

이러한 제공자 중 하나와 연결된 이메일 주소가 로그인에 사용된 이메일 주소와 일치하는 경우, 계정은 자동으로 연결됩니다. 이는 반환된 사용자 ID가 두 로그인 방식 모두에서 동일하며, 사용자의 프로필 정보(예: 이름, 프로필 사진 및 기타 관련 정보)가 병합됨을 의미합니다.

패스키

패스키는 비밀번호 대신 생체 인증, 디바이스 PIN 또는 보안 키를 사용하여 원활한 사용자 경험과 향상된 보안을 제공합니다.

패스키 설정 및 사용에 대한 자세한 내용을 알아보세요.

패스키

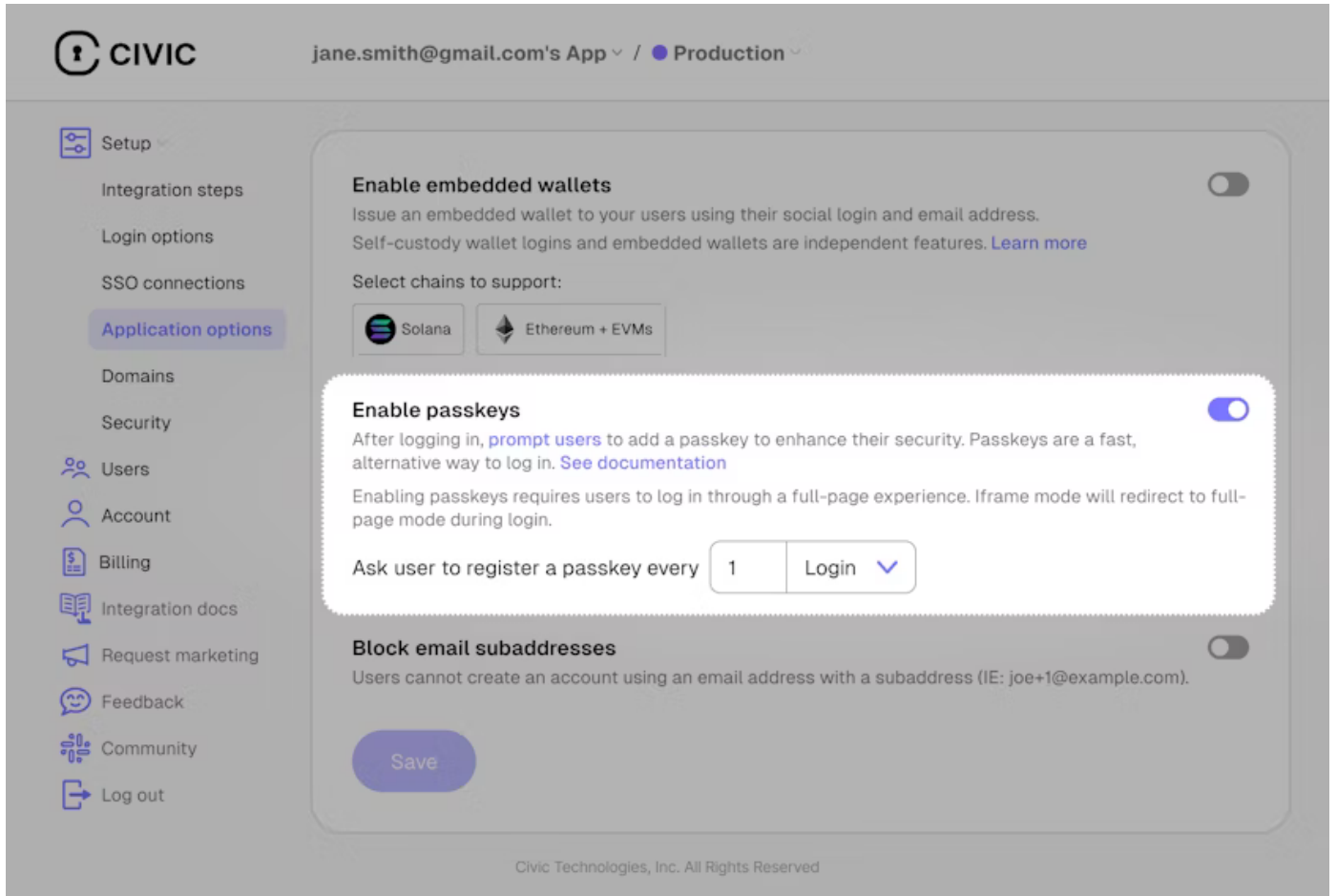
Civic Auth와 함께 패스워드 없는 인증을 위해 패스키를 구현하고 사용하는 완벽한 가이드입니다.

패스키는 기존 비밀번호를 대체하는 안전하고 비밀번호 없는 수단을 제공합니다. 이는 향상된 보안을 제공하면서 생체 인증, 디바이스 PIN 또는 보안 키를 통한 원활한 사용자 경험을 제공합니다.

패스키 활성화 방법

앱에서 패스키를 활성화하려면:

1. [Civic Auth Dashboard](#)로 이동합니다
2. 사이드바에서 **Application Options**로 이동합니다
3. **Enable passkeys** 섹션을 찾습니다
4. 패스키 설정을 **ON**으로 전환합니다
5. 사용자가 언제 패스키 생성을 요청받을지 구성합니다



프롬프트 빈도 옵션

사용자가 패스키 생성을 요청받을 빈도를 로그인 횟수 또는 마지막 로그인 이후의 시간에 따라 결정할 수도 있습니다.

패스키란 무엇인가?

비밀번호를 입력하는 대신, 사용자는 다음과 같은 간단하고 안전한 동작을 통해 인증합니다:

- 생체 인식 스캔(지문, Face ID) 또는 디바이스 PIN
- 물리적 보안 키 탭(e.g., YubiKey)
- 비밀번호 관리자(e.g., Bitwarden)를 통한 인증

인증 흐름

패스키는 여러 인증 요소를 단일 단계에서 결합하여 **내재적 다중 요소 인증(MFA)**을 제공합니다:

- **소유한 것:** 패스키가 저장된 디바이스
- **존재/지식:** 생체 인증(지문/얼굴) 또는 디바이스 PIN

사용자는 다음을 수행할 수 있습니다:

- 패스키를 사용하여 이메일/소셜 로그인 단계를 건너뛰고 더 빠르게 접근
- 항상 전통적인 로그인 방식으로 폴백
- 패스키가 없을 경우 선호하는 주요 인증 방식을 계속 사용

이러한 유연성은 사용자가 잠기는 일이 없도록 보장하며, 내장된 MFA를 통해 향상된 보안을 제공합니다.

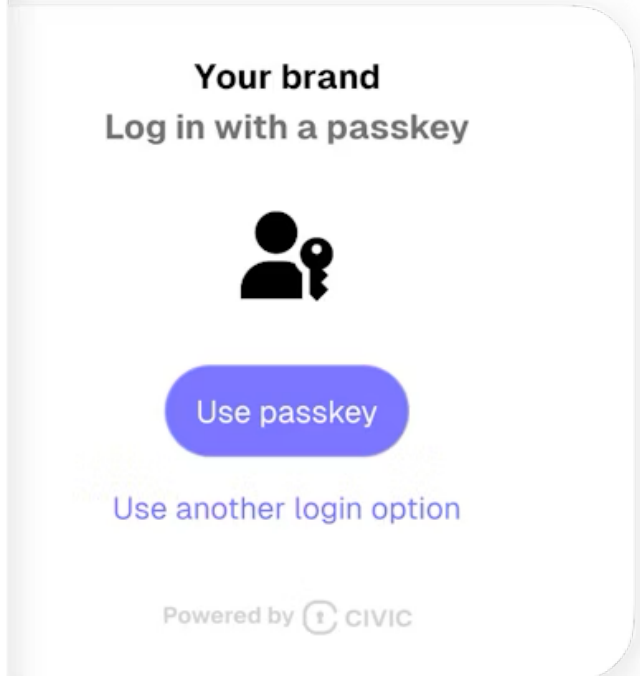
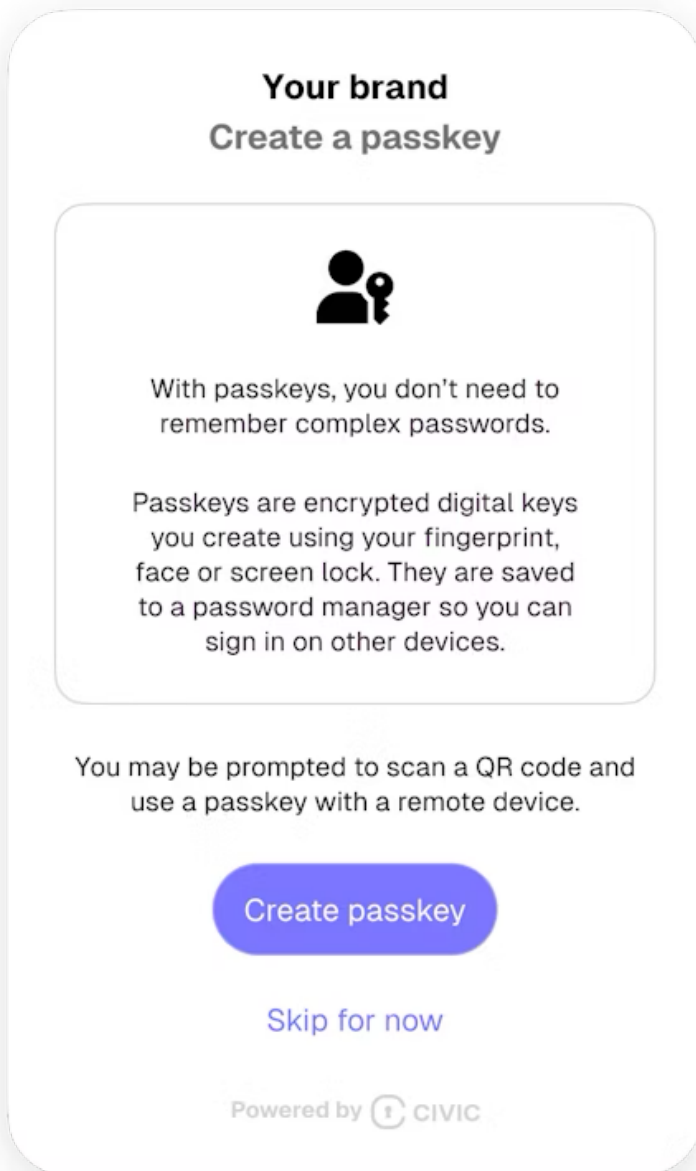
주요 이점

- **보안 강화:** WebAuthn 표준을 기반으로 구축된 패스키는 피싱 및 서버 측 데이터 유출에 강력하게 대응합니다
- **간편한 경험:** 복잡한 비밀번호를 생성, 기억 또는 입력할 필요가 없습니다
- **디바이스 간 동기화:** 패스키는 플랫폼 계정(Apple ID, Google 계정)을 통해 디바이스 간 동기화됩니다
- **더 빠른 로그인:** 자격 증명을 입력하지 않고 빠르게 인증할 수 있습니다
- **내장 MFA:** 추가 단계 없이 내재적인 다중 요소 인증을 제공합니다

사용자 경험

패스키가 활성화되면, 사용자가 경험하게 될 사항은 다음과 같습니다:

1. **최초 설정:** 이메일 또는 소셜 로그인 후, 사용자는 더 빠른 미래 접근을 위해 패스키 생성을 요청받습니다
2. **이후 로그인:** 사용자는 패스키로 인증하여 즉시 접근할 수 있습니다
3. **폴백 옵션:** 사용자는 언제든지 패스키 인증을 건너뛰고 전통적인 로그인 방식을 사용할 수 있습니다



브라우저 호환성

현재 브라우저 지원 및 호환성 매트릭스:

- WebAuthn API Browser Compatibility - WebAuthn 지원에 대한 실시간 호환성 데이터
- Passkeys.dev Device Support - 플랫폼 및 브라우저별 패스키 지원에 대한 상세 정보
- FIDO Alliance Certified Products - 인증된 인증기 목록

주요 호환성 리소스

- MDN WebAuthn API Documentation - 브라우저 관련 주석이 포함된 종합적인 API 참조
- WebAuthn.io - 현재 브라우저에서 패스키 지원을 테스트할 수 있는 인터랙티브 데모
- Passkeys.dev - 개발자 리소스 및 구현 가이드

중요한 구현 참고 사항

크로스 플랫폼 동기화:

- Apple 패스키는 iCloud Keychain을 통해 동기화됩니다 (Apple 생태계 전용)
- Google 패스키는 Google Password Manager를 통해 동기화됩니다 (Chrome/Android)
- Windows Hello 패스키는 일반적으로 디바이스에 종속됩니다

애플리케이션에서 **추가적인 코드 변경은 필요하지 않습니다** – Civic Auth가 모든 패스키 구현을 자동으로 처리합니다. 포함되는 기능은 다음과 같습니다:

- WebAuthn 자격 증명 생성 및 검증
- 크로스 플랫폼 호환성
- 폴백 처리
- 오류 관리

사용자가 패스키로 로그인했던 기존 방식으로 로그인했다면, 애플리케이션은 동일한 사용자 인증 토큰을 계속 수신합니다.