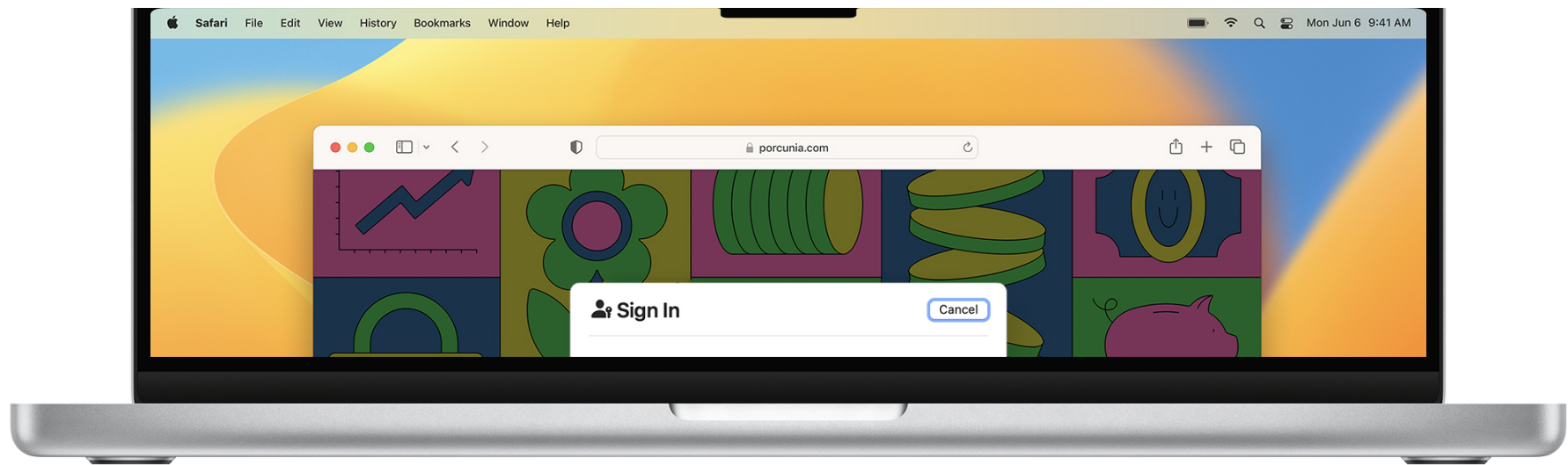# Passkeys

Passkeys can be synced using external providers, and you can create groups to share passwords and passkeys. In managed environments, passkeys support Managed Apple IDs, including syncing via iCloud Keychain, and access controls let people easily restrict how passkeys are shared and synced.

✦ What's new

Passkeys are a replacement for passwords that are more secure, easier to use, and can't be phished. They offer faster sign-in, fewer password resets, and reduced support costs. Use the new automatic passkey upgrade API to create a passkey when someone signs in to your app and let them know that the passkey was saved — all without interrupting their flow.

Watch the latest video ⊙

Streamlined sign-in, without passwords

Saving and using a passkey is quick and easy with one-step account creation and sign-in using Face ID or Touch ID. There's no need to create or manage passwords. Because passkeys are

synced with iCloud Keychain, they're available across Apple devices. You can even use your iPhone to sign in to apps and websites on non-Apple devices.

Next-generation account security

Based on FIDO Alliance and W3C standards, passkeys replace passwords with cryptographic key pairs. These key pairs profoundly improve security.

Strong credentials. Every passkey is strong. They're never guessable, reused, or weak.

Safe from server leaks. Because servers only keep public keys, servers are less valuable targets for hackers.

Safe from phishing. Passkeys are intrinsically linked with the app or website they were created for, so people can never be tricked into using their passkey to sign in to a fraudulent app or website.

In iCloud Keychain, passkeys are end-to-end encrypted, so even Apple can't read them. A passkey ensures a strong, private relationship between a person and your app or website.

Works alongside passwords

Since signing in with passkeys uses AutoFill and Face ID or Touch ID for biometric verification, the transition to passkeys is seamless. This lets people use passkeys alongside passwords, so you don't need to adjust your sign-in page based on credential type. You'll use the new Authentication Services API to add passkeys, creating sign-in flows that are familiar to users.

Get more insights into passkeys

Spotlight on developers who brought passkeys to their apps ›

Top 10 questions and answers on passkey integration ›

Browse the latest documentation

View sample code

Watch the latest passkeys videos

Ask questions on the forums

Platforms	Topics & Technologies	Resources	Programs
iOS	Accessibility	Documentation	Apple Developer Program
iPadOS	Accessories	Tutorials	Apple Developer Enterprise Program
macOS	App Extensions	Downloads	App Store Small Business Program
tvOS	App Store	Forums	MFi Program
visionOS	Audio & Video	Videos	News Partner Program
watchOS	Augmented Reality		Video Partner Program
	Business	**Support**	Security Bounty Program
Tools	Design	Support Articles	Security Research Device Program
Swift	Distribution	Contact Us	
SwiftUI	Education	Bug Reporting	**Events**
Swift Playgrounds	Fonts	System Status	Meet with Apple Experts
TestFlight			Apple Developer Centers

Xcode	Games		App Store Awards
Xcode Cloud	Health & Fitness	**Account**	Apple Design Awards
SF Symbols	In-App Purchase	Apple Developer	Apple Developer Academies
	Localization	App Store Connect	Entrepreneur Camp
	Maps & Location	Certificates, IDs, & Profiles	WWDC
	Machine Learning	Feedback Assistant	
	Open Source		
	Security		
	Safari & Web		

Get the Apple Developer app.

Light   Dark   Auto