Introducing Password AutoFill for Apps
Reducing friction for your users
Session 206

Ricky Mondello, iOS Engineer
Logging in to apps is hard.
In iOS 11, Password AutoFill can help.
Demo
Password AutoFill
Make the QuickType bar appear

Ensure credentials are filled properly

Make QuickType show the right credentials

Authenticating against a third-party service
Make the QuickType Bar Appear
As easy as setting a property
Make the QuickType Bar Appear
Make the QuickType Bar Appear

Requires that you have saved passwords
Make the QuickType Bar Appear

Requires that you have saved passwords

UITextField and UITextView are eligible
Make the QuickType Bar Appear

Requires that you have saved passwords

UITextField and UITextView are eligible

Fields conforming to <UITextInput> are also eligible
Make the QuickType Bar Appear

Requires that you have saved passwords

UITextField and UITextView are eligible

Fields conforming to `<UITextInput>` are also eligible

iOS uses heuristics to determine which fields show the bar
Make the QuickType Bar Appear

Requires that you have saved passwords

UITextField and UITextView are eligible

Fields conforming to `<UITextInput>` are also eligible

iOS uses heuristics to determine which fields show the bar

You can tell iOS to make the bar appear
// UITextInputTraits Protocol
// Indicates the semantic meaning of a text-entry area
optional var textContentType: UITextContentType! { get set }
// UITextInputTraits Protocol
// Indicates the semantic meaning of a text-entry area

optional var textContentType: UITextContentType! { get set }

.name
.namePrefix
.givenName
.middleName
.familyName
.nameSuffix
.nickname
.jobTitle
.organizationName
.typeLocation
.fullStreetAddress
.streetAddressLine1
.streetAddressLine2
.addressCity
.addressState
.addressCityAndState
.sublocality
.countryName
.postalCode
.telephoneNumber
.emailAddress
.URL
.creditCardNumber
// UITextInputTraits Protocol
// Indicates the semantic meaning of a text-entry area

optional var textContentType: UITextContentType! { get set }
Make the QuickType Bar Appear
Use UITextContentType
Make the QuickType Bar Appear
Use UITextContentType

Deploy content types to guarantee AutoFill is available
Make the QuickType Bar Appear
Use UITextContentType

Deploy content types to guarantee AutoFill is available

Can set in code or through Interface Builder
Make the QuickType Bar Appear
Use UITextContentType

Deploy content types to guarantee AutoFill is available

Can set in code or through Interface Builder

UITextContentType.username enables AutoFill for two-screen login
Make the QuickType Bar Appear
Use UITextContentType

Deploy content types to guarantee AutoFill is available

Can set in code or through Interface Builder

UITextContentType.username enables AutoFill for two-screen login

You can combine UITextContentType.username and UIKeyboardType.emailAddress
Make the QuickType Bar Appear
Use UITextContentType

Deploy content types to guarantee AutoFill is available

Can set in code or through Interface Builder

UITextContentType.username enables AutoFill for two-screen login

You can combine UITextContentType.username and UIKeyboardType.emailAddress

UITextContentType.password enables AutoFill for “revealed” passwords
Ensure Credentials are Filled Properly
Validating UI and logging in
Ensure Information Is Filled Properly

What to know
Ensure Information Is Filled Properly

What to know

iOS fills both the user name and password fields
Ensure Information Is Filled Properly

What to know

iOS fills both the user name and password fields

Even if your app prevents changing the first responder
Ensure Information Is Filled Properly

What to know

iOS fills both the user name and password fields

Even if your app prevents changing the first responder

Touch ID will make your app become inactive
Ensure Information Is Filled Properly

Best practices

Don’t tear down login UI when becoming inactive
Ensure Information Is Filled Properly

Best practices

Don’t tear down login UI when becoming inactive

Use “did change” delegate methods or notifications to validate UI
Ensure Information Is Filled Properly

Best practices

Don’t tear down login UI when becoming inactive

Use “did change” delegate methods or notifications to validate UI

UITextFieldTextDidChange: NSNotification.Name

UITextViewTextDidChange: NSNotification.Name

protocol UITextViewDelegate { optional public func textViewDidChange(…) }

protocol UITextViewInputDelegate { public func textDidChange(…) }
Ensure Information Is Filled Properly

Best practices

Don’t tear down login UI when becoming inactive

Use “did change” delegate methods or notifications to validate UI

UITextFieldTextDidChange: NSNotification.Name

UITextViewTextDidChange: NSNotification.Name

protocol UITextViewDelegate { optional public func textViewDidChange(…) }

protocol UITextInputDelegate { public func textDidChange(…) }

Read information from your text field’s text property
Make QuickType Show the Right Credentials

Associating your app and website
Associating Your App and Website

iOS
Associating Your App and Website

example.com
Associating Your App and Website

https://example.com/.well-known/apple-app-site-association
Associating Your App and Website

https://example.com/.well-known/apple-app-site-association
Associating Your App and Website

iOS

JSON

{ }
Associating Your App and Website

iOS

JSON
Demo

Integrating with Password AutoFill

Maureen Daum, iOS Engineer
Example Inc.
Apple Developer Program

People
Send invitations to your development team so they can take advantage of membership resources.

Certificates, Identifiers & Profiles
Manage the certificates, identifiers, profiles, and devices you need to develop and distribute apps.

iTunes Connect
Publish and manage your apps on the App Store with iTunes Connect.
Example Inc.
Apple Developer Program

People
Send invitations to your development team so they can take advantage of membership resources.

Certificates, Identifiers & Profiles
Manage the certificates, identifiers, profiles, and devices you need to develop and distribute apps.

iTunes Connect
Publish and manage your apps on the App Store with iTunes Connect.
Setup and configure services for this App ID.

ID: com.example.Shiny
Name: Shiny

Enable | Service
--- | ---
[ ] | App Groups (Disabled)
[ ] | Apple Pay (Disabled)
[ ] | Associated Domains (Enabled)
[ ] | Data Protection (Disabled)

Sharing and Permissions:
- Complete Protection
- Protected Unless Open
- Protected Until First User Authentication
{
    "webcredentials": {
        "apps": [ "E5336VM85F.com.example.Shiny" ]
    }
}
Associated Domains File

Where to serve the file

https://example.com/.well-known/apple-app-site-association

https://example.com/apple-app-site-association
Debugging Associated Domains
swcd Console output when running your app

swcd Request for 'https://wrong.example.com/apple-app-site-association' denied: 404/0x194 Not Found
Debugging Associated Domains

swcd Console output when running your app

swcd  Request for 'https://wrong.example.com/apple-app-site-association' denied: 404/0x194 Not Found

swcd  Updated app ID 'E5336VM85F.ios.Shiny', domain 'shiny.example.com', flags 0x0 < > -> 0x2 < SiteApproved > on check
Viewing, Changing, and Adding Saved Passwords
Touch ID for “Settings”
View Passwords
Cancel
apple.com — rmondello
apple.com
example.com — rmondello
shiny.example.com
webkit.org — rmondello
webkit.org
Add Password
User name: rmondello
Password: 5Ts-C2M-XDn-576

WEBSITES

shiny.example.com
Authenticating Against a Third-Party Service
Use Safari View Controller
Example 3rd Party Service

rmondello

Log In

I forgot my password

I am new customer, sign me up!

Example 3rd Party Service is a registered trademark of Sample LLC, a subsidiary of Examples Inc.
What's New in Safari View Controller
Summary
Summary

Friction when logging in can lead users to abandon your app.
Summary

Friction when logging in can lead users to abandon your app

Password AutoFill uses heuristics to work automatically in some apps
Summary

Friction when logging in can lead users to abandon your app

Password AutoFill uses heuristics to work automatically in some apps

You can guarantee that it works by
Summary

Friction when logging in can lead users to abandon your app.

Password AutoFill uses heuristics to work automatically in some apps.

You can guarantee that it works by:

• Testing your app.
Summary

Friction when logging in can lead users to abandon your app.

Password AutoFill uses heuristics to work automatically in some apps.

You can guarantee that it works by:

• Testing your app
• Adopting `UITextContentType.username` and `UITextContentType.password`
Summary

Friction when logging in can lead users to abandon your app.

Password AutoFill uses heuristics to work automatically in some apps.

You can guarantee that it works by:

- Testing your app.
- Adopting `UITextContentType.username` and `UITextContentType.password`.
- Adopting the webcredentials Associated Domains service.
More Information

## Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>What’s New in Safari View Controller</td>
<td>Executive Ballroom</td>
<td>Thursday 10:00AM</td>
</tr>
<tr>
<td>The Keys to a Better Text Input Experience</td>
<td>Grand Ballroom B</td>
<td>Friday 11:00AM</td>
</tr>
<tr>
<td>Introducing Safari View Controller</td>
<td>WWDC 2015</td>
<td></td>
</tr>
<tr>
<td>Labs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Safari, WebKit, and Password AutoFill Lab</strong></td>
<td>Technology Lab D</td>
<td>Wed 10:00AM–12:00PM</td>
</tr>
<tr>
<td><strong>Safari, WebKit, and Password AutoFill Lab</strong></td>
<td>Technology Lab B</td>
<td>Wed 5:10PM–6:10PM</td>
</tr>
<tr>
<td><strong>Safari, WebKit, and Password AutoFill Lab</strong></td>
<td>Technology Lab K</td>
<td>Thurs 11:00AM–1:00PM</td>
</tr>
</tbody>
</table>