Your App, Your Website, and Safari
Ensuring continuity of user experience

Session 506
Ricky Mondello
Safari and WebKit Engineer
Continuity
Continuity ˌkäntnˈ(y)oʊətē| noun  (pl. continuities)
the unbroken and consistent existence or operation of something over a period of time
Continuity ˌkäntnˈ(y)əʊətē noun  (pl. continuities)  
the unbroken and consistent existence or operation of something over a period of time
Handoff
<table>
<thead>
<tr>
<th>Related Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Adopting Handoff on iOS and OS X</strong></td>
</tr>
<tr>
<td>Mission</td>
</tr>
<tr>
<td>Wednesday 2:00PM</td>
</tr>
</tbody>
</table>
What You Will Learn

How to:

• Use Safari credentials in your iOS apps
• Make Safari AutoFill work better on your website
• Help Safari promote your website
• Make your website consistent across devices
Credentials in Safari
And how your app can use them
Shiny

Hi ricky! Here’s Today’s Shiny:

Change Password  Log out

Log Out  Change Password
Safari AutoFill

Shipping Address

Shipping Address

Jane
Company Name (optional)
555
6948601
1600 Butler Street
Apt, Suite, Bldg. (optional)
94920
Belvedere Tiburon, CA
United States

This is a business address

Shipping Notifications

Shipment notification emails are sent to the Billing Contact. Another recipient email address may be added below.

jappleseed@icloud.com

Apple Shipping Policy

- Signature may be required for delivery
- We do not ship to P.O. boxes
- Delivery estimates below include item preparation
- We do not ship directly to APO/FPO addresses.
Safari AutoFill

Please Enter Your Apple ID and password.

Apple ID
appleseed

Password

Login
Forgot Password?

An Apple ID is required to use this tool.

If you are having trouble logging in, here are ways to get help:
- Find your Apple ID
- Reset your password
- Request assistance with logging in to your Developer Account
- Request assistance with Bug Reporter login issues

Notes: For best performance use Safari on OS X Lion or greater. JavaScript and cookies must be enabled to access the Bug Reporter.
To create your account, you must enter a password in the form below. Your email address and Real Name (if provided) will be shown with changes you make.

**Email Address:** jappleseed@icloud.com  
**Real Name:** Jane Appleseed

Type your password: [Use Safari suggested password: KpN-HM6-SFg-y6P]  
This password will be saved in your iCloud Keychain, so it is available for AutoFill on all your devices.

This account will not be created until 5:30 PDT.

If you do not wish to create an account with this email click the cancel account button below and your details will be forgotten.

Cancel Account
Safari AutoFill
Credentials in Safari
Credentials in Safari

Mac

iPad

iPhone
Credentials in Safari
Credentials in Safari

Shiny

Username
Password

Log In

Forgot Password?
Credentials in Safari

Select a Safari Saved Password to Use With “Shiny”
You can view and remove passwords in the Passwords & Autofill section of Safari settings.

andrew
example.com

Not Now  OK

Forgot Password?
Safari Credentials in Native Apps

Andrew Whalley
Core OS Security Engineering
Password Security
Password Security
Password Security
Password Security

Shiny
Welcome to Shiny. Please log in.
Username
Password
If you don’t have an account, please sign up.

Log in

andrew
afz-zma-9JX-puL
Overview

Select a Safari Saved Password to Use With “Shiny”
You can view and remove passwords in the Passwords & Autofill section of Safari settings.

andrew
example.com

Not Now OK

Forgot Password?
Overview

Associate your app and website
Overview

Associate your app and website

• One file on your website
Overview

Associate your app and website
• One file on your website
• One new entitlement
Overview

Associate your app and website
• One file on your website
• One new entitlement

Simple APIs
Overview

Associate your app and website
• One file on your website
• One new entitlement

Simple APIs
• Ask for credentials
Overview

Associate your app and website
• One file on your website
• One new entitlement

Simple APIs
• Ask for credentials
• Create or update credentials
Associating App and Website
Associating App and Website
Associating App and Website
Associating App and Website
## Associating App and Website

<table>
<thead>
<tr>
<th>App</th>
<th>Website</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shiny App</td>
<td>example.com</td>
<td>Pending</td>
</tr>
<tr>
<td>Foo Client</td>
<td>foo.com</td>
<td>Approved</td>
</tr>
<tr>
<td>Foo Admin App</td>
<td>foo.com</td>
<td>Approved</td>
</tr>
</tbody>
</table>
Associating App and Website

Shiny App

iOS

example.com
Associating App and Website

Shiny App

example.com

example.com
Associating App and Website

Shiny App

example.com
Associating App and Website

Shiny App

iOS

example.com

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

Shiny App

iOS

example.com

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

Shiny App

iOS

example.com
Associating App and Website

Shiny App

iOS

example.com
Associating App and Website

Shiny App

example.com
Associating App and Website

Shiny App

example.com
Associating App and Website

Shiny App

iOS

example.com
Associating App and Website

Shiny App
to
example.com
# Associating App and Website

<table>
<thead>
<tr>
<th>App</th>
<th>Website</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shiny App</td>
<td>example.com</td>
<td>Pending</td>
</tr>
<tr>
<td>Foo Client</td>
<td>foo.com</td>
<td>Approved</td>
</tr>
<tr>
<td>Foo Admin App</td>
<td>foo.com</td>
<td>Approved</td>
</tr>
</tbody>
</table>
## Associating App and Website

<table>
<thead>
<tr>
<th>App</th>
<th>Website</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shiny App</td>
<td>example.com</td>
<td>Approved</td>
</tr>
<tr>
<td>Foo Client</td>
<td>foo.com</td>
<td>Approved</td>
</tr>
<tr>
<td>Foo Admin App</td>
<td>foo.com</td>
<td>Approved</td>
</tr>
</tbody>
</table>
Determining App Intent
Entitlements
Determining App Intent

Entitlements

application-identifier
Determining App Intent

Entitlements

application-identifier
teamid.bundle-identifier
Determining App Intent

Entitlements

application-identifier
teamid.bundle-identifier
YWBN8XTPBJ.com.example.Shiny
Determining App Intent

Entitlements

application-identifier
teamid.bundle-identifier
YWBN8XTPBJ.com.example.Shiny
Determining App Intent

Entitlements

application-identifier
teamid.bundle-identifier
YWBN8XTPBJ.com.example.Shiny
Determining App Intent

Entitlements

application-identifier
teamiid.bundle-identifier
YWBN8XTPBJ.com.example.Shiny
Determining App Intent

Entitlements

application-identifier

teamid.bundle-identifier

YWBN8XTPBJ.com.example.Shiny

com.apple.developer.associated-domains
Determining App Intent

Entitlements

application-identifier
teamid.bundle-identifier
YWBN8XTPBJ.com.example.Shiny

com.apple.developer.associated-domains
service:fully.qualified.domain.name[:port number]
Determining App Intent

Entitlements

application-identifier
teamid.bundle-identifier
YWBN8XTPBJ.com.example.Shiny

com.apple.developer.associated-domains
service:fully.qualified.domain.name[:port number]
webcredentials:example.com
associated-domains Entitlement

Xcode
Associating App and Website
Associating App and Website
Determining Website Intent

https://example.com/apple-app-site-association
Determining Website Intent

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

```json
{
  "webcredentials": {
    "apps": [
      "YWBN8XTPBJ.com.example.Shiny",
      "YWBN8XTPBJ.com.example.ShinyUploader"
    ]
  }
}
```
Determining Website Intent

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

```json
{
  "webcredentials": {
    "apps": [ "YWBN8XTPBJ.com.example.Shiny",
              "YWBN8XTPBJ.com.example.ShinyUploader" ]
  }
}
```
Determining Website Intent

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

{  
    "webcredentials": {  
        "apps": [  
            "YWBN8XTPBJ.com.example.Shiny",
            "YWBN8XTPBJ.com.example.ShinyUploader"
        ]
    }
}
Determining Website Intent

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

{
  "webcredentials": {
    "apps": [ "YWBN8XTPBJ.com.example.Shiny",
              "YWBN8XTPBJ.com.example.ShinyUploader" ]
  }
}
Determining Website Intent

Content-Type must be application/pkcs7-mime
Signed
• Cryptographic Message Syntax
• S/MIME
apple-app-site-association File

Signing the JSON
Signing the JSON

echo '{"webcredentials":{"apps":["YWBN8XTPBJ.com.example.Shiny","YWBN8XTPBJ.com.example.ShinyUploader"]}}' > json.txt
apple-app-site-association File

Signing the JSON

echo '{"webcredentials":{"apps":["YWBN8XTPBJ.com.example.Shiny", "YWBN8XTPBJ.com.example.ShinyUploader"]}}' > json.txt
apple-app-site-association File
Signing the JSON

cat json.txt | openssl smime –sign –inkey example.com.key
–signer example.com.pem
–certfile intermediate.pem
–noattr –nodetach
–outform DER > apple-app-site-association
apple-app-site-association File
Signing the JSON

cat json.txt | openssl smime -sign -inkey example.com.key
   -signer example.com.pem
   -certfile intermediate.pem
   -noattr -nodetach
   -outform DER > apple-app-site-association
apple-app-site-association File

Signing the JSON

cat json.txt | openssl smime -sign -inkey example.com.key
   -signer example.com.pem
   -certfile intermediate.pem
   -noattr -nodetach
   -outform DER > apple-app-site-association
apple-app-site-association File

Signing the JSON

cat json.txt | openssl smime -sign -inkey example.com.key
               -signer example.com.pem
               -certfile intermediate.pem
               -noattr -nodetach
               -outform DER > apple-app-site-association
Associating App and Website
Determining Website Intent

Return results
Determining Website Intent

Return results

<table>
<thead>
<tr>
<th>HTTP Status</th>
<th>Service and Identifier Match</th>
<th>Signature Is Valid</th>
<th>Approval State</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Approved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>Denied</td>
<td></td>
<td></td>
</tr>
<tr>
<td>300-499</td>
<td>n/a</td>
<td>n/a</td>
<td>Denied</td>
</tr>
<tr>
<td>500-599</td>
<td>n/a</td>
<td>n/a</td>
<td>Retry</td>
</tr>
</tbody>
</table>
## Determining Website Intent

Return results

<table>
<thead>
<tr>
<th>HTTP Status</th>
<th>Service and Identifier Match</th>
<th>Signature Is Valid</th>
<th>Approval State</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>Approved</td>
</tr>
</tbody>
</table>
## Determining Website Intent

<table>
<thead>
<tr>
<th>HTTP Status</th>
<th>Service and Identifier Match</th>
<th>Signature Is Valid</th>
<th>Approval State</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>✔</td>
<td>✔</td>
<td>Approved</td>
</tr>
<tr>
<td>200</td>
<td>✗</td>
<td>✗</td>
<td>Denied</td>
</tr>
<tr>
<td>200</td>
<td>✗</td>
<td>✗</td>
<td>Approved</td>
</tr>
</tbody>
</table>
## Determining Website Intent

<table>
<thead>
<tr>
<th>HTTP Status</th>
<th>Service and Identifier Match</th>
<th>Signature Is Valid</th>
<th>Approval State</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>☑</td>
<td>☑</td>
<td>Approved</td>
</tr>
<tr>
<td>200</td>
<td>✗</td>
<td>✗</td>
<td>Denied</td>
</tr>
<tr>
<td>300-499</td>
<td>n/a</td>
<td>n/a</td>
<td>Denied</td>
</tr>
</tbody>
</table>
## Determining Website Intent

Return results

<table>
<thead>
<tr>
<th>HTTP Status</th>
<th>Service and Identifier Match</th>
<th>Signature Is Valid</th>
<th>Approval State</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>✓</td>
<td>✓</td>
<td>Approved</td>
</tr>
<tr>
<td>200</td>
<td>✗</td>
<td>✗</td>
<td>Denied</td>
</tr>
<tr>
<td>300-499</td>
<td>n/a</td>
<td>n/a</td>
<td>Denied</td>
</tr>
<tr>
<td>500-599</td>
<td>n/a</td>
<td>n/a</td>
<td>Retry</td>
</tr>
</tbody>
</table>
Native Apps with Accounts
Native Apps with Accounts
Native Apps with Accounts

1. Check for Saved Credentials
2. Prompt User
3. Save Credentials
4. Log In
Native Apps with Accounts

Flow:
1. Check for Saved Credentials
2. Prompt User
3. Save Credentials
4. Log In
Native Apps with Accounts

1. Check for Saved Credentials
2. Prompt User
3. Save Credentials
4. Log In
Native Apps with Accounts

Check for Saved Credentials

Prompt User

Save Credentials

Log In
Native Apps with Accounts

1. Log In
2. Check for Saved Credentials
3. Prompt User
4. Save Credentials
5. Log In
Native Apps with Accounts

Check for Saved Credentials

Prompt User

Save Credentials

Log In
Native Apps with Accounts

Check for Saved Credentials

- Prompt User
- Save Credentials
- Log In
Native Apps with Accounts

1. Check for Saved Credentials
2. Prompt User
3. Save Credentials
4. Log In
Native Apps with Accounts

- Check for Saved Credentials
- Prompt User
- Save Credentials
- Log In
Native Apps with Accounts

- Check for Saved Credentials
- Prompt User
- Save Credentials
- Log In
Native Apps with Accounts

1. Check for Saved Credentials
2. Check Safari for Credentials
3. Prompt User
4. Save Credentials
5. Log In
Native Apps with Accounts

- Check for Saved Credentials
  - Check Safari for Credentials
    - Prompt User
      - Save Credentials
        - Log In
Native Apps with Accounts

1. Check for Saved Credentials
2. Check Safari for Credentials
3. Prompt User
4. Save Credentials
5. Log In
Native Apps with Accounts

1. Check for Saved Credentials
2. Check Safari for Credentials
3. Prompt User
4. Save Credentials
5. Log In

- Select a Safari Saved Password to Use With "Shiny"
  - You can view and remove passwords in the Passwords & Autofill section of Safari settings.
  - Example Password: andrew@example.com

- Not Now
- OK
Native Apps with Accounts

Log In

- Check for Saved Credentials
- Check Safari for Credentials
- Prompt User
- Save Credentials

- Log In

Check Safari for Credentials
Native Apps with Accounts

1. Check for Saved Credentials
2. Check Safari for Credentials
3. Prompt User
4. Save Credentials
5. Log In
New APIs
New APIs

SecRequestSharedWebCredential
SecAddSharedWebCredential
SecCreateSharedWebCredentialPassword
New APIs

SecRequestSharedWebCredential
SecAddSharedWebCredential
SecCreateSharedWebCredentialPassword
SecRequestSharedWebCredential(CFSTR("example.com"),
    CFSTR("lordsecurity@icloud.com"),
    ^(CFArrayRef credentials, CFErrorRef error) {

});
SecRequestSharedWebCredential(CFSTR("example.com"),
    CFSTR("lordsecurity@icloud.com"),
    ^(CFArrayRef credentials, CFErrortRef error) {
        
    });
Obtaining Safari Credentials

SecRequestSharedWebCredential(CFSTR("example.com"),
   CFSTR("lordsecurity@icloud.com"),
   ^(CFArrayRef credentials, CFErrorRef error) {

});
SecRequestSharedWebCredential(CFSTR("example.com"),
   CFSTR("lordsecurity@icloud.com"),
   ^(CFArrayRef credentials, CFErrroRef error) {

});
Obtaining Safari Credentials

SecRequestSharedWebCredential(NULL,
   NULL,
   ^(CFArrayRef credentials, CFErrorRef error) {
        
   });
SecRequestSharedWebCredential(NULL, // use domains from entitlement
                                        NULL,
                                        ^(CFArrayRef credentials, CFErrorRef error) {

                                        });
Obtaining Safari Credentials

SecRequestSharedWebCredential(NULL, // use domains from entitlement
    NULL, // find all matching usernames
    ^(CFArrayRef credentials, CFErrorRef error) {

});
SecRequestSharedWebCredential(NULL,  // use domains from entitlement
                              NULL,  // find all matching usernames
                              ^(CFArrayRef credentials, CFErrorRef error) {
                              });
Obtaining Safari Credentials

SecRequestSharedWebCredential(NULL,       // use domains from entitlement
                                NULL,       // find all matching usernames
                                ^(CFArrayRef credentials, CFErrorRef error) {
                                
                                });
Obtaining Safari Credentials

SecRequestSharedWebCredential

Select a Safari Saved Password to Use With "Shiny"
You can view and remove passwords in the Passwords & Autofill section of Safari settings.

[Check Box]
andrew@example.com

[Button] Not Now
[Button] OK
SecRequestSharedWebCredential(NULL, // use domains from entitlement
    NULL, // find all matching usernames
    ^(CFArrayRef credentials, CFErrorRef error) {

});
Obtaining Safari Credentials

SecRequestSharedWebCredential(NULL,       // use domains from entitlement
                               NULL,       // find all matching usernames
                             ^(CFArrayRef credentials, CFErrorRef error) {

                           });
Obtaining Safari Credentials

SecRequestSharedWebCredential(NULL, // use domains from entitlement
NULL, // find all matching usernames
^(CFArrayRef credentials, CFErrorRef error) {

});
Obtaining Safari Credentials

SecRequestSharedWebCredential(NULL,       // use domains from entitlement
                              NULL,       // find all matching usernames
                              ^(CFArrayRef credentials, CFErrorRef error) {
                                    
                                      });
Obtaining Safari Credentials

SecRequestSharedWebCredential(NULL, // use domains from entitlement
    NULL,       // find all matching usernames
    ^(CFArrayRef credentials, CFErrorRef error) {

        if (!error && CFArrayGetCount(credentials)) {
            CFDictionaryRef credential = CFArrayGetValueAtIndex(credentials, 0);
            NSString *username = CFDictionaryGetValue(credential, kSecAttrAccount);
            NSString *password = CFDictionaryGetValue(credential, kSecSharedPassword);
            dispatch_async(dispatch_get_main_queue(), ^{
                [serverManager attemptLoginWithUsername:username password:password];
            });
        } else {
            dispatch_async(dispatch_get_main_queue(), ^{
                [self showLoginUI];
            });
        }
    });
Obtaining Safari Credentials

SecRequestSharedWebCredential(NULL, // use domains from entitlement
    NULL, // find all matching usernames
    ^(CFArrayRef credentials, CFErrorRef error) {
        if (!error && CFArrayGetCount(credentials)) {
            CFDictionaryRef credential = CFArrayGetValueAtIndex(credentials, 0);
            NSString *username = CFDictionaryGetValue(credential, kSecAttrAccount);
            NSString *password = CFDictionaryGetValue(credential, kSecSharedPassword);
            dispatch_async(dispatch_get_main_queue(), ^{
                [serverManager attemptLoginWithUsername:username password:password];
            });
        } else {
            dispatch_async(dispatch_get_main_queue(), ^{
                [self showLoginUI];
            });
        }
    });
Obtaining Safari Credentials

SecRequestSharedWebCredential(NULL, // use domains from entitlement
NULL, // find all matching usernames
^(CFArrayRef credentials, CFErrorRef error) {

if (!error && CFArrayGetCount(credentials)) {
    CFDictionaryRef credential = CFArrayGetValueAtIndex(credentials, 0);
    NSString *username = CFDictionaryGetValue(credential, kSecAttrAccount);
    NSString *password = CFDictionaryGetValue(credential, kSecSharedPassword);
    dispatch_async(dispatch_get_main_queue(), ^{
        [serverManager attemptLoginWithUsername:username password:password];
    });
} else {
    dispatch_async(dispatch_get_main_queue(), ^{
        [self showLoginUI];
    });
}

}));
Obtaining Safari Credentials

SecRequestSharedWebCredential(NULL,       // use domains from entitlement
NULL,       // find all matching usernames
^(CFArrayRef credentials, CFErrorRef error) {

    if (!error && CFArrayGetCount(credentials)) {
        CFDictionaryRef credential = CFArrayGetValueAtIndex(credentials, 0);
        NSString *username = CFDictionaryGetValue(credential, kSecAttrAccount);
        NSString *password = CFDictionaryGetValue(credential, kSecSharedPassword);
        dispatch_async(dispatch_get_main_queue(), ^{
            [serverManager attemptLoginWithUsername:username password:password];
        });
    } else {
        dispatch_async(dispatch_get_main_queue(), ^{
            [self showLoginUI];
        });
    }
});
SecRequestSharedWebCredential(NULL, // use domains from entitlement
    NULL, // find all matching usernames
    ^(CFArrayRef credentials, CFErrorRef error) {

        if (!error && CFArrayGetCount(credentials)) {
            CFDictionaryRef credential = CFArrayGetValueAtIndex(credentials, 0);
            NSString *username = CFDictionaryGetValue(credential, kSecAttrAccount);
            NSString *password = CFDictionaryGetValue(credential, kSecSharedPassword);
            dispatch_async(dispatch_get_main_queue(), ^{
                [serverManager attemptLoginWithUsername:username password:password];
            });
        } else {
            dispatch_async(dispatch_get_main_queue(), ^{
                [self showLoginUI];
            });
        }
    });
Obtaining Safari Credentials

SecRequestSharedWebCredential(NULL,       // use domains from entitlement
                           NULL,       // find all matching usernames
                           ^(CFArrayRef credentials, CFErrorRef error) {

    if (!error && CFArrayGetCount(credentials)) {
        CFDictionaryRef credential = CFArrayGetValueAtIndex(credentials, 0);
        NSString *username = CFDictionaryGetValue(credential, kSecAttrAccount);
        NSString *password = CFDictionaryGetValue(credential, kSecSharedPassword);
        dispatch_async(dispatch_get_main_queue(), ^{
            [serverManager attemptLoginWithUsername:username password:password];
        });
    } else {
        dispatch_async(dispatch_get_main_queue(), ^{
            [self showLoginUI];
        });
    }
});
Obtaining Safari Credentials

SecRequestSharedWebCredential(NULL, // use domains from entitlement
    NULL, // find all matching usernames
    ^(CFArrayRef credentials, CFErrorRef error) {

    if (!error && CFArrayGetCount(credentials)) {
        CFDictionaryRef credential = CFArrayGetValueAtIndex(credentials, 0);
        NSString *username = CFDictionaryGetValue(credential, kSecAttrAccount);
        NSString *password = CFDictionaryGetValue(credential, kSecSharedPassword);
        dispatch_async(dispatch_get_main_queue(), ^{
            [serverManager attemptLoginWithUsername:username password:password];
        });
    } else {
        dispatch_async(dispatch_get_main_queue(), ^{
            [self showLoginUI];
        });
    }
});
Obtaining Safari Credentials

SecRequestSharedWebCredential(NULL, // use domains from entitlement
    NULL, // find all matching usernames
    ^(CFArrayRef credentials, CFErrorRef error) {

        if (!error && CFArrayGetCount(credentials)) {
            CFDictionaryRef credential = CFArrayGetValueAtIndex(credentials, 0);
            NSString *username = CFDictionaryGetValue(credential, kSecAttrAccount);
            NSString *password = CFDictionaryGetValue(credential, kSecSharedPassword);
            dispatch_async(dispatch_get_main_queue(), ^{
                [serverManager attemptLoginWithUsername:username password:password];
            });
        } else {
            dispatch_async(dispatch_get_main_queue(), ^{
                [self showLoginUI];
            });
        }
});
Telling Safari About Credentials
Telling Safari About Credentials

SecAddSharedWebCredential(CFSTR("example.com"),
    username,
    password,
    ^(CFErrorRef error) {
        NSLog(@"%@", error);
    });
Telling Safari About Credentials

SecAddSharedWebCredential(CFSTR("example.com"),
   username,
   password,
   ^(CFErrorRef error) {
       NSLog(@"%@", error);
   });
Telling Safari About Credentials

SecAddSharedWebCredential(CFSTR("example.com"),
    username,
    password,
    ^(CFErrorRef error) {
        NSLog(@"%@", error);
    });
Telling Safari About Credentials

SecAddSharedWebCredential(CFSTR("example.com"),
    username,
    password,
    ^(CFErrorRef error) {
        NSLog(@"%@", error);
    });
Telling Safari About Credentials

SecAddSharedWebCredential(CFSTR("example.com"),
    username,
    password,
    ^(CFErrorRef error) {
        NSLog(@"%@", error);
    });
SecAddSharedWebCredential(CFSTR("example.com"),
    username,
    password,
    ^(CFErrorRef error) {
        NSLog(@"%@", error);
    });
Telling Safari About Credentials

Updating a Credential

SecAddSharedWebCredential(CFSTR("example.com"),
    username,
    newPassword,
    ^(CFErrorRef error) {
        NSLog(@"%@", error);
    });
Telling Safari About Credentials

Updating a Credential

SecAddSharedWebCredential(CFSTR("example.com"),
    username,
    newPassword,
    ^(CFErrorRef error) {
        NSLog(@"%@", error);
    });
Telling Safari About Credentials

Deleting a Credential

SecAddSharedWebCredential(CFSTR("example.com"),
    username,
    NULL,
    ^(CFErrorRef error) {
        NSLog(@"%@", error);
    });
SecAddSharedWebCredential(CFSTR("example.com"),
    username,
    NULL,
    ^(CFErrorRef error) {
        NSLog(@"%@", error);
    });
Generating a Password

CFStringRef suggestedPwd = SecCreateSharedWebCredentialPassword();
Generating a Password

CFStringRef suggestedPwd = SecCreateSharedWebCredentialPassword();

JVA-9cx-ZfT-c4H
k6S-kgS-c7E-hhK
2H4-mVC-dx3-Vbv
9uB-VC9-52L-9H5
meR-7CK-d3B-em9
mMJ-8LN-Wf4-Jsb
wu4-PGe-gEQ-xTE
645-3bY-7CG-3mn
5B8-T8E-rRG-EtB
ESB-naN-PrF-s3r
wN4-NUq-dHB-oU9
Z7k-n3r-nqB-oNP
Error Conditions

- Missing entitlement errSecParam "No domain provided"
- Missing entitlement errSecMissingEntitlement "example.com not found in com.apple.developer.associated-domains entitlement"
- Domain not approved errSecItemNotFound "no matching items found"
- No saved credentials errSecItemNotFound "no matching items found"
<table>
<thead>
<tr>
<th>Condition</th>
<th>CFErrorGetCode()</th>
<th>CFErrorCopyDescription()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing entitlement</td>
<td>errSecParam</td>
<td>&quot;No domain provided&quot;</td>
</tr>
<tr>
<td>&quot;example.com not found in com.apple.developer.associated-domains entitlement&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Domain not approved</td>
<td>errSecItemNotFound</td>
<td>&quot;no matching items found&quot;</td>
</tr>
<tr>
<td>No saved credentials</td>
<td>errSecItemNotFound</td>
<td>&quot;no matching items found&quot;</td>
</tr>
</tbody>
</table>
## Error Conditions

<table>
<thead>
<tr>
<th>Condition</th>
<th>CFErrorGetCode()</th>
<th>CFErrorCopyDescription()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing entitlement</td>
<td>errSecParam</td>
<td>&quot;No domain provided&quot;</td>
</tr>
<tr>
<td>Domain not approved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No saved credentials</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Error Conditions

<table>
<thead>
<tr>
<th>Condition</th>
<th>CFErrorGetCode()</th>
<th>CFErrorCopyDescription()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing entitlement</td>
<td>errSecParam</td>
<td>&quot;No domain provided&quot;</td>
</tr>
<tr>
<td>Missing entitlement</td>
<td>errSecMissingEntitlement</td>
<td>&quot;example.com not found in com.apple.developer.associated-domains entitlement&quot;</td>
</tr>
</tbody>
</table>
## Error Conditions

<table>
<thead>
<tr>
<th>Condition</th>
<th>CFErrorGetCode()</th>
<th>CFErrorCopyDescription()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing entitlement</td>
<td>errSecParam</td>
<td>&quot;No domain provided&quot;</td>
</tr>
<tr>
<td>Missing entitlement</td>
<td>errSecMissingEntitlement</td>
<td>&quot;example.com not found in com.apple.developer.associated-domains entitlement&quot;</td>
</tr>
<tr>
<td>Domain not approved</td>
<td>errSecItemNotFound</td>
<td>&quot;no matching items found&quot;</td>
</tr>
</tbody>
</table>
## Error Conditions

<table>
<thead>
<tr>
<th>Condition</th>
<th>CFErrorGetCode()</th>
<th>CFErrorCopyDescription()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing entitlement</td>
<td>errSecParam</td>
<td>&quot;No domain provided&quot;</td>
</tr>
<tr>
<td>Missing entitlement</td>
<td>errSecMissingEntitlement</td>
<td>&quot;example.com not found in com.apple.developer.associated-domains entitlement&quot;</td>
</tr>
<tr>
<td>Domain not approved</td>
<td>errSecItemNotFound</td>
<td>&quot;no matching items found&quot;</td>
</tr>
<tr>
<td>No saved credentials</td>
<td>errSecItemNotFound</td>
<td>&quot;no matching items found&quot;</td>
</tr>
</tbody>
</table>
Demo
Putting it into practice
Safari Credentials in Native Apps
Credentials on the Web

Ricky Mondello
Safari and WebKit Engineer
Expressing Intent
Expressing Intent on the Web
The Password Lifecycle

No Account -> Created Account -> Logged In
Logged Out

Logged Out can change to Change Password
Change Password can change to Logged In
Logged In can change to Created Account
Created Account can change to No Account
Test Your Website with Safari AutoFill
Testing Your Site and Safari AutoFill

The password lifecycle
Testing Your Site and Safari AutoFill

Creating an account

- No Account
- Created Account
- Logged In
- Logged Out
- Change Password
Testing Your Site and Safari AutoFill

Creating an account

- No Account
- Created Account
- Logged In
- Change Password

Logged Out

Flow diagram showing the transitions between account statuses.
Testing Your Site and Safari AutoFill
Logging in with saved credentials

No Account ➔ Created Account ➔ Logged In ➔ Change Password ➔ Logged Out ➔ No Account
Testing Your Site and Safari AutoFill

Logging in with saved credentials

- No Account
- Created Account
- Logged In
- Change Password
- Logged Out

Created Account
Logged In
Logged Out
Testing Your Site and Safari AutoFill

Changing a password

Diagram:
- No Account → Created Account → Logged In → Change Password → Logged Out
- Logged Out can transition to No Account or Created Account
- Logged In can transition to Change Password or Logged Out
Testing Your Site and Safari AutoFill
Changing a password

- No Account
- Created Account
- Logged In
- Logged Out
- Change Password
Testing Your Site and Safari AutoFill

Resetting a password
Testing Your Site and Safari AutoFill
Resetting a password
Testing Your Site and Safari AutoFill

Logging in for the first time

No Account -> Created Account -> Logged In -> Change Password

Logged Out
Testing Your Site and Safari AutoFill

Logging in for the first time

- No Account → Created Account → Logged In → Change Password
- Logged Out ↔ Logged In
- Logged Out ↔ Created Account
- Created Account ↔ No Account
Testing Your Site and Safari AutoFill

What to do if something doesn’t work

No Account → Created Account → Logged In → Change Password → Logged Out

Logged Out → Created Account → Logged In

Logged In ↔ Change Password

Logged Out ↔ Created Account
Your Forms’ Intent
Your Forms’ Intent

New values for the `autocomplete` attribute

```html
<input type="text" autocomplete="username">

<input type="password" autocomplete="current-password">

<input type="password" autocomplete="new-password">
```
Logging in

<form id="login" action="login.php" method="post">
  <input type="text" autocomplete="username">
  <input type="password" autocomplete="current-password">
  <input type="submit" value="Log in">
</form>
Sign up

<form id="signup" action="signup.php" method="post">
  <input type="text" autocomplete="username">
  <input type="password" autocomplete="new-password">
  <input type="submit" value="Sign up">
</form>
Signing up
With confirmation

<form id="signup" action="signup.php" method="post">
  
  <input type="text" autocomplete="username">
  
  <input type="password" autocomplete="new-password">
  
  <input type="password" autocomplete="new-password">
  
  <input type="submit" value="Sign up">
</form>
Changing One’s Password
While logged in

<form id="change-password"
action="update-user.php" method="post">
  
  <input type="text" autocomplete="username"
        readonly value="rmondello">
  
  <input type="password"
        autocomplete="current-password">
  
  <input type="password"
        autocomplete="new-password">
  
  <input type="submit" value="Change Password">
</form>
Changing One’s Password
While logged in

<form id="change-password"
action="update-user.php" method="post">
  <input type="text" autocomplete="username"
  readonly value="rmondello">
  <input type="password"
  autocomplete="current-password">
  <input type="password"
  autocomplete="new-password">
  <input type="submit" value="Change Password">
</form>
Changing One’s Password
While logged in

<form id="change-password"
    action="update-user.php" method="post">
    <input type="hidden" autocomplete="username"
        value="rmondello">

    <input type="password"
        autocomplete="current-password">

    <input type="password"
        autocomplete="new-password">

    <input type="submit"
        value="Change Password">
<form>

Current Password

New Password

Change Password
Changing One’s Password

After forgetting the password

<form id="change-password"
action="update-user.php" method="post">

<input type="text" autocomplete="username"
readonly value="rmondello">

<input type="password"
autocomplete="new-password">

<input type="submit"
value="Change Password">
</form>
## Your Forms’ Intent

**The autocomplete attribute**

<table>
<thead>
<tr>
<th><strong>Username</strong></th>
<th><strong>Email</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>current-password</td>
<td>street-address</td>
</tr>
<tr>
<td>new-password</td>
<td>city</td>
</tr>
<tr>
<td>name-full</td>
<td>state</td>
</tr>
<tr>
<td>given-name</td>
<td>province</td>
</tr>
<tr>
<td>middle-name</td>
<td>postal-code</td>
</tr>
<tr>
<td>family-name</td>
<td>country</td>
</tr>
<tr>
<td>birthday-year</td>
<td>phone</td>
</tr>
<tr>
<td>birthday-month</td>
<td>phone-area-code</td>
</tr>
<tr>
<td>birthday-day</td>
<td>organization</td>
</tr>
</tbody>
</table>
Testing Your Site and Safari AutoFill

No Account → Created Account → Logged In → Change Password

Logged Out → Logged In → Change Password

Logged Out → Logged In → Change Password
Declaring a State Transition

History API

history.pushState

history.replaceState
Declaring a State Transition

History API

```javascript
history.pushState(data, null, "page.html");

history.replaceState(data, null, "page.html");
```
iCloud
iCloud Calendar
iCloud Mail
Twitter
iCloud
Declaring a State Transition

History API
Declaring a State Transition

History API

```javascript
history.pushState({ data: "yourData" }, null);
```
Safari and Your Website’s Content
Iconography and metadata
Iconography

Favicon

In-page markup:

```html
<head>
  <link rel="icon"
       type="image/png"
       href="http://www.example.com/favicon.png">
</head>
```

Standard path:

/favicon.ico
Iconography

Apple Touch Icon

Used on iOS home screen

In-page markup:

```
<head>
  <link rel="apple-touch-icon" sizes="152x152"
  href="/apple-touch-icon.png" />
</head>
```

See *Configuring Web Applications* on the Safari Web Content Guide
Apple Touch Icon
Iconography
Apple Touch Icon
Iconography
Apple Touch Icon
Iconography
Reading List
Magical thinking is the attribution of causal relationships between actions and events where scientific consensus says that there are none. In religion, folk religion, and superstition beliefs, the correlation posited is often between religious ritual, prayer, sacrifice, or the observance of a taboo, and an expected benefit or recompense. In clinical psychology, magical thinking can cause a patient to experience fear (characterizing various acts or having unreal thoughts, or having an associated emotion or belief). Magical thinking has been found in a wide range of individuals, including children, adults, and the elderly. It is a type of causal reasoning or causal illusion that leads to meaningful associations of supposed causes and effects in the environment. Magical thinking is often characterized by a belief in the power of thought to change the environment, or by a belief that thoughts can influence the environment.
Your Site in Shared Links

Apple Hot News
Introducing the WebKit FTL JIT
Just a decade ago, Javascript – the programming language used to drive web page interactions – was thought to be too slow for serious application deve...

Apple Hot News
Teachers Use iPad to Enhance Learning in Special Education Across the Globe
For special education teachers around the world, iPad opens up a world of possibilities. Meet teachers at three schools — Special School Podleady in...

Apple Hot News
2014 NHL Playoffs: Teams Turning to High-Tech Analysis During Games
Sports Illustrated writes about how teams competing in the NHL’s Stanley Cup playoffs are using iPad to help. They’re analyzing video, visualizing the...

Apple Hot News
Apple Updates MacBook Air
Apple today updated the MacBook Air lineup with faster processors and lower prices. Now starting at $899, the ultra-thin notebooks feature powerful In...

Apple Hot News
Apple Reports Second-Quarter Results
Apple has announced financial...
Your Site in Shared Links
Surfin' Safari

Introducing the WebKit FTL JIT

Just a decade ago, JavaScript – the programming language used to drive web page interactions – was thought to be too slow for serious application deve...
Your Site in Shared Links
Reading List
Expressing Intent on the Web
Reading List

Better titles and descriptions through metadata

<head>
  <title>Surfin' Safari - Archive » Introducing the WebKit FTL JIT</title>
</head>
Reading List

Better titles and descriptions through metadata

<head>
<title>Surfin' Safari - Archive » Introducing the WebKit FTL JIT</title>

<meta property="og:title" content="Introducing the WebKit FTL JIT" />
<meta property="og:description" content="Just a decade ago, JavaScript – the programming..." />
</head>
Reading List

Better titles and descriptions through metadata

<head>
	<title>Surfin' Safari - Archive » Introducing the WebKit FTL JIT</title>

	<meta property="og:title" content="Introducing the WebKit FTL JIT" />
	<meta property="og:description" content="Just a decade ago, JavaScript – the programming…" />
</head>

Surfin' Safari
Introducing the WebKit FTL JIT
Just a decade ago, JavaScript – the programming language used to drive web page interactions – was thought to be too slow for serious application dev...
Just a decade ago, JavaScript – the programming...
Metadata Standards

Resources

OpenGraph
http://ogp.me

The Dublin Core® Metadata Initiative
http://dublincore.org

schema.org
http://schema.org
The Sticky Mobile Website
The Sticky Mobile Website

The Sticky Mobile Website

Generic Blog

Lorem ipsum dolor sit amet


Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet

The Sticky Mobile Website

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam nec
lacus condimentum, feugiat nisl vitae, fermentum eros. Mauris
placerat tincidunt orci ut aliquet. Etiam malesuada magna nibh, ac
elementum eros tincidunt quis. Cras et lorem eget purus elementum
ullamcorper. Proin leo nulla, vulputate id enim suscipit, tempor
semper metus. Praesent sodales dui eu nisi egestas ultricies. Fusce
sollicitudin eleifend nisi, tristique porta justo sollicitudin porta.
Aliquam a ornare leo. Interdum et malesuada fames ac ante ipsum
primis in faucibus. Aliquam eget adipiscing erat. Proin blandit enim
mollis leo lobortis, eu ullamcorper eros interdum. Vestibulum
condimentum convallis dui, at auctor nisi molestie non.
Pellentesque nunc diam, pharetra et mi a, scelerisque mattis leo.
Fusce tempor libero id accumsan euismod. Maecenas bibendum
felis eget dui imperdiet, id congue nunc sodales.

Aenean vitae pulvinar orci, sed pellentesque felis. Donec mattis
dapibus eros condimentum pretium. Curabitur non lacinia felis. In
mattis elit ac diam tempor volutpat. Duis elementum sem justo.
The Sticky Mobile Website

GET example.com/…

example.com

m.example.com

Desktop Layout

Mobile Layout
The Sticky Mobile Website

GET example.com/...

example.com

GET m.example.com/...

m.example.com

Desktop Layout

Mobile Layout
The Sticky Mobile Website
Relevant to Handoff
The Sticky Mobile Website

GET example.com/...

example.com

GET m.example.com/...

m.example.com

Desktop Layout

Mobile Layout
The Sticky Mobile Website

GET example.com/… → example.com → Desktop Layout

GET m.example.com/… → m.example.com → Mobile Layout
Responsive Design
One site for everyone

GET example.com/…

Responsive Layout
Responsive Design
One site for everyone
Related Sessions

- Designing Responsive Web Experiences

<p>| Marina | Friday 10:15AM |</p>
<table>
<thead>
<tr>
<th>Related Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Media for the Web</td>
</tr>
<tr>
<td>Nob Hill</td>
</tr>
<tr>
<td>Tuesday 11:30AM</td>
</tr>
</tbody>
</table>
Continuity |ˈkæntn(y)əʊətə| noun (pl. continuities) the unbroken and consistent existence or operation of something over a period of time
Summary

Use Safari credentials in your apps
Test Safari AutoFill with your website
Use the autocomplete attribute
Deploy high-quality website icons
Express intent with metadata
Unify mobile and desktop websites
More Information

Evangelism
evangelism@apple.com

Developer Technical Support
http://developer.apple.com/contact

Apple Developer Forums
http://devforums.apple.com
## Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Media for the Web</td>
<td>Nob Hill</td>
<td>Tuesday 11:30AM</td>
</tr>
<tr>
<td>Keychain and Authentication with Touch ID</td>
<td>Nob Hill</td>
<td>Wednesday 10:15AM</td>
</tr>
<tr>
<td>Adopting Handoff on iOS and OS X</td>
<td>Mission</td>
<td>Wednesday 2:00 PM</td>
</tr>
<tr>
<td>Designing Responsive Web Experiences</td>
<td>Marina</td>
<td>Friday 10:15AM</td>
</tr>
<tr>
<td>Lab</td>
<td>Location</td>
<td>Time</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>----------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Security Lab</td>
<td>Core OS Lab B</td>
<td>Wednesday 11:30AM</td>
</tr>
<tr>
<td>Safari and WebKit Lab</td>
<td>Media Lab B</td>
<td>Wednesday 4:30PM</td>
</tr>
<tr>
<td>Safari and WebKit Lab</td>
<td>Media Lab B</td>
<td>Thursday 2:00PM</td>
</tr>
<tr>
<td>Security and Privacy Lab</td>
<td>Core OS Lab B</td>
<td>Thursday 3:15PM</td>
</tr>
</tbody>
</table>