What’s New in Safari Extensions

Brian Weinstein, Safari Engineer
Distributing Safari Extensions
Content blocking notifications
Window, tab, and page management
Showing and hiding popovers
Communicating with your app
Safari Extensions allow Mac users to explore the web exactly the way you want. Find helpful information about a web page, display news headlines, or quickly access and share content to your favorite app and services. To get started, choose Preferences from the Safari menu, select the Extensions tab, and click the checkbox to turn on the extension.
Distributing Safari Extensions

Content blocking notifications

Window, tab, and page management

Showing and hiding popovers

Communicating with your app
Distributing Safari Extensions
Content blocking notifications
Window, tab, and page management
Showing and hiding popovers
Communicating with your app
Content Blocking Notifications
Content Blocking Notifications
Content Blocking Notifications

NEW
Content Blocking Notifications
Info.plist

Safari App Extension and Content Blocker must be in the same containing app

```xml
<key>NSExtension</key>
<dict>
    <key>SFSafariAssociatedContentBlockers</key>
    <array>
        <string>com.example.content-blocker</string>
    </array>
    ...
</dict>
```
Content Blocking Notifications

Info.plist

Safari App Extension and Content Blocker must be in the same containing app

```
<key>NSExtension</key>
<dict>
    <key>SFSafariAssociatedContentBlockers</key>
    <array>
        <string>com.example.content-blocker</string>
    </array>
</dict>
```
Content Blocking Notifications
Info.plist

Safari App Extension and Content Blocker must be in the same containing app

```
<key>NSExtension</key>
<dict>
  <key>SFSafariAssociatedContentBlockers</key>
  <array>
    <string>com.example.content-blocker</string>
  </array>
  ...
</dict>
```
Content Blocking Notifications
Info.plist

Safari App Extension and Content Blocker must be in the same containing app

```
<key>NSExtension</key>
<dict>
  <key>SFSafariAssociatedContentBlockers</key>
  <array>
    <string>com.example.content-blocker</string>
  </array>
  ...
</dict>
```
Content Blocking Notifications
Info.plist

Safari App Extension and Content Blocker must be in the same containing app

```
<key>NSExtension</key>
<dict>
    <key>SFSafariAssociatedContentBlockers</key>
    <array>
        <string>com.example.content-blocker</string>
    </array>
... 
</dict>
```
Safari App Extension and Content Blocker must be in the same containing app

```xml
  <key>NSExtension</key>
  <dict>
    <key>SFSafariAssociatedContentBlockers</key>
    <array>
      <string>com.example.content-blocker</string>
    </array>
  </dict>
```
Content Blocking Notifications
Info.plist

Safari App Extension and Content Blocker must be in the same containing app

```
<key>NSExtension</key>
<dict>
  <key>SFSafariAssociatedContentBlockers</key>
  <array>
    <string>com.example.content-blocker</string>
  </array>
... 
</dict>
```
Content Blocking Notifications

Principal Object

Batched notifications

Only URLs with permission

```swift
func contentBlocker(withIdentifier contentBlockerIdentifier: String,
                    blockedResourcesWithURLs urls: [URL], on page: SFSafariPage) {
    // Handle the content blocker notification.
}
```
Distributing Safari Extensions
Content blocking notifications
Window, tab, and page management
Showing and hiding popovers
Communicating with your app
Distributing Safari Extensions
Content blocking notifications
Window, tab, and page management
Showing and hiding popovers
Communicating with your app
Page Navigation Notifications

Can be used to follow redirect chain

URL can be nil for some navigations

```swift
func page(_ page: SFSafariPage, willNavigateTo url: URL?) {
    // Handle the page navigation.
}
```
Demo
Get the visible contents of a page

```javascript
page.getScreenshotOfVisibleArea { image in
  // Image is the visible contents of the page.
}
```
Windows and Tabs
Opening full page web content

Navigate directly from the app extension process

Access to the baseURI in the app extension process

```swift
SFSafariExtension.getBaseURI { baseURI in
    guard let baseURI = baseURI else { return }
    tab.navigate(to: baseURI.appendingPathComponent("full_page_content.html"))
}
```
Windows and Tabs
Opening full page web content

Navigate directly from the app extension process

Access to the baseURI in the app extension process

```swift
SFSafariExtension.getBaseURI { baseURI in
  guard let baseURI = baseURI else { return }
  tab.navigate(to: baseURI.appendingPathComponent("full_page_content.html"))
}
```
Windows and Tabs
Opening full page web content

Navigate directly from the app extension process

Access to the baseURI in the app extension process

```
SFSafariExtension.getBaseURI { baseURI in
  guard let baseURI = baseURI else { return }
  tab.navigate(to: baseURI.appendingPathComponent("full_page_content.html"))
}
```
Windows and Tabs
Opening full page web content

Navigate directly from the app extension process

Access to the baseURI in the app extension process

```swift
SFSafariExtension.getBaseURI { baseURI in
    guard let baseURI = baseURI else { return }
    let fullPage = baseURI.appendingPathComponent("full_page_content.html")
    tab.navigate(to: fullPage)
}
```
Windows and Tabs
Opening full page web content

Navigate directly from the app extension process

Access to the baseURI in the app extension process

```swift
SFSafariExtension.getBaseURI { baseURI in
    guard let baseURI = baseURI else { return }
    tab.navigate(to: baseURI.appendingPathComponent("full_page_content.html"))
}
```
**Windows and Tabs**

Opening full page web content

Navigate directly from the app extension process

Access to the baseURI in the app extension process

```swift
SFSafariExtension.getBaseURI { baseURI in
    guard let baseURI = baseURI else { return }
    tab.navigate(to: baseURI.appendingPathComponent("full_page_content.html"))
}
```
Windows and Tabs
Opening full page web content

Navigate directly from the app extension process

Access to the baseURI in the app extension process

```swift
SFSafariExtension.getBaseURI { baseURI in
    guard let baseURI = baseURI else { return }
    tab.navigate(to: baseURI.appendingPathComponent("full_page_content.html"))
}
```
Navigate directly from the app extension process

Access to the baseURI in the app extension process

```swift
SFSafariExtension.getBaseURI { baseURI in
    guard let baseURI = baseURI else { return }
    tab.navigate(to: baseURI.appendingPathComponent("full_page_content.html"))
}
```
Windows and Tabs
Accessing all windows and tabs

Iterate through all open windows and tabs

```swift
SFSafariApplication.getAllWindows { windows in
    for window in windows {
        window.getAllTabs { tabs in /* … */ }
    }
}
page.getContainingTab { tab in
    tab.getContainingWindow { window in /* … */ }
}
```
Windows and Tabs
Accessing all windows and tabs

Iterate through all open windows and tabs

```swift
SFSafariApplication.getAllWindows { windows in
    for window in windows {
        window.getAllTabs { tabs in /* … */ }
    }
}
page.getContainingTab { tab in
    tab.getContainingWindow { window in /* … */ }
}
```
Iterate through all open windows and tabs

```
SFSafariApplication.getAllWindows { windows in
    for window in windows {
        window.getAllTabs { tabs in /* … */ }
    }
}

page.getContainingTab { tab in
    tab.getContainingWindow { window in /* … */ }
}
```
Iterate through all open windows and tabs

```swift
SFSafariApplication.getAllWindows { windows in
    for window in windows {
        window.getAllTabs { tabs in /* … */ }
    }
}
page.getContainingTab { tab in
    tab.getContainingWindow { window in /* … */ }
}
```
Iterate through all open windows and tabs

```swift
SFSafariApplication.getAllWindows { windows in
    for window in windows {
        window.getAllTabs { tabs in /* … */ }
    }
}
page.getContainingTab { tab in
    tab.getContainingWindow { window in /* … */ }
}
```
Windows and Tabs
Accessing all windows and tabs

Iterate through all open windows and tabs
Access containing windows and tabs

SFSafariApplication.getAllWindows { windows in
    for window in windows {
        window.getAllTabs { tabs in /* ... */ }
    }
}

page.getContainingTab { tab in
    tab.getContainingWindow { window in /* ... */ }
}
Windows and Tabs
Accessing all windows and tabs

Iterate through all open windows and tabs

Access containing windows and tabs

```swift
SFSafariApplication.getAllWindows { windows in
    for window in windows {
        window.getAllTabs { tabs in /* … */ }
    }
}

page.getContainingTab { tab in
    tab.getContainingWindow { window in /* … */ }
}
```
Iterate through all open windows and tabs

Access containing windows and tabs

```swift
SFSafariApplication.getAllWindows { windows in
    for window in windows {
        window.getAllTabs { tabs in /* … */ }
    }
}

page.getContainingTab { tab in
    tab.getContainingWindow { window in /* … */ }
}
```
Windows and Tabs
Accessing all windows and tabs

Iterate through all open windows and tabs

Access containing windows and tabs

```swift
SFSafariApplication.getAllWindows { windows in
    for window in windows {
        window.getAllTabs { tabs in /* … */ }
    }
}
page.getContainingTab { tab in
    tab.getContainingWindow { window in /* … */ }
}
```
Iterate through all open windows and tabs

Access containing windows and tabs

Pinned tabs have a nil window

```swift
SFSafariApplication.getAllWindows { windows in
    for window in windows {
        window.getAllTabs { tabs in /* … */ }
    }
}

page.getContainingTab { tab in
    tab.getContainingWindow { window in /* … */ }
}
```
Distributing Safari Extensions
Content blocking notifications
Window, tab, and page management
Showing and hiding popovers
Communicating with your app
Distributing Safari Extensions
Content blocking notifications
Window, tab, and page management
Showing and hiding popovers
Communicating with your app
Popovers

Programmatically show and dismiss

```swift
window.getToolbarItem { toolbarItem in
    toolbarItem?.showPopover()
}

// From within SFSafariExtensionViewController
self.dismissPopover()
```
Demo
Distributing Safari Extensions
Content blocking notifications
Window, tab, and page management
Showing and hiding popovers
Communicating with your app
Communicating with Your App
Communicating with Your App
Communicating with Your App
Communicating with Your App

App Group

NSXPCConnection

UserDefaults(suiteName:)
Communicating with Your App

Sending a message to your extension

Extension must be enabled

```swift
SFSafariApplication.dispatchMessage(withName: "message",
    toExtensionWithIdentifier: "com.extension.identifier",
    userInfo: 
        ["key": "value"],
    completionHandler: { error in /* ... */ })
```
Communicating with Your App

SFSafariExtensionHandling
.messageReceivedFromContainingApp
func messageReceivedFromContainingApp(withName messageName: String,
        userInfo: [String: Any]? = nil) {
    // Handle the message from your containing app.
}
Communicating with Your App

SFSafariApplication.dispatchMessage

SFSafariExtensionHandling.messageReceivedFromContainingApp

NSUserDefaults(suiteName:)

NSXPCConnection

App Group
Communicating with Your App

- SFSafariApplication.dispatchMessage
- SFSafariExtensionHandling.messageReceivedFromContainingApp

App Group

- NSXPCConnection
- UserDefaults(suiteName:)
Communicating with Your App

- **SFSafariApplication.dispatchMessage**
- **SFSafariExtensionHandling.messageReceivedFromContainingApp**
- **NSUserDefaults(suiteName:)**
Communicating with Your App

SFSafariApplication.dispatchMessage

SFSafariExtensionHandling.messageReceivedFromContainingApp

App Group

NSUserDefaults(suiteName:)

NSXPCConnection

UserDefaults(suiteName:):
Distributing Safari Extensions
Content blocking notifications
Window, tab, and page management
Showing and hiding popovers
Communicating with your app
Distributing Safari Extensions
Content blocking notifications
Window, tab, and page management
Showing and hiding popovers
Communicating with your app
Summary

Distribute through the Mac App Store or as a notarized app

Associate your Content Blocker with your Safari App Extension

Adopt App Groups to communicate between your Safari Extension and app