What’s New in Safari and WebKit

Session 234

Shloka Kini, Developer Publications
Jason Sandmeyer, Marketing Communications
Agenda

Security

Performance

Rich experience
Security
Use WKWebView
Use WKWebView

UIWebView deprecated
Use WKWebView

UIWebView deprecated

Works on macOS and iOS
Use WKWebView

UIWebView deprecated

Works on macOS and iOS
Use WKWebView

UIWebView deprecated

Works on macOS and iOS

Runs in a separate process from your app
Extending Safari: History
Extending Safari: History

2010: Safari Extensions
Extending Safari: History

2010: Safari Extensions

2014: App Extensibility for macOS and iOS
Extending Safari: History

2010: Safari Extensions

2014: App Extensibility for macOS and iOS
Extending Safari: History

2010: Safari Extensions
2014: App Extensibility for macOS and iOS
2015: Content Blockers for macOS and iOS
Extending Safari: History

2010: Safari Extensions
2014: App Extensibility for macOS and iOS
2015: Content Blockers for macOS and iOS
Extending Safari: History

2010: Safari Extensions
2014: App Extensibility for macOS and iOS
2015: Content Blockers for macOS and iOS
2016: Safari App Extensions for macOS
Extending Safari: History

2010: Safari Extensions
2014: App Extensibility for macOS and iOS
2015: Content Blockers for macOS and iOS
2016: Safari App Extensions for macOS
Legacy Safari Extensions
Legacy Safari Extensions

Safari 12 blocks support for .safariextz extensions outside of the Safari Extensions Gallery
Legacy Safari Extensions

Safari 12 blocks support for .safariextz extensions outside of the Safari Extensions Gallery

New submissions to the Safari Extensions Gallery will be accepted until the end of 2018
Legacy Safari Extensions

Safari 12 blocks support for .safariextz extensions outside of the Safari Extensions Gallery

New submissions to the Safari Extensions Gallery will be accepted until the end of 2018

Transition to a Safari App Extension or Content Blocker
Security

WKWebView

Safari Extensions

Subresource Integrity

Intelligent Tracking Prevention

Automatic Strong Passwords

Security Code AutoFill
Subresource Integrity

Files

HTML

Servers

Your server

Third-party server

User

User’s device
Subresource Integrity

Files

- HTML
- JS
- CSS

Servers

- Your server
- Third-party server

User

- HTTPS
- User’s device
Subresource Integrity

Files
- HTML
- JavaScript (JS)
- CSS

Servers
- Your server
- Third-party server

User

HTTPS

User’s device
Subresource Integrity

Files
- HTML
- JS
- CSS

Servers
- Your server
- Compromised Third-party server

User

User’s device

HTTPS
Subresource Integrity

- Files: HTML, JS, CSS
- Servers: Your server, Compromised Third-party server
- User's device
- HTTPS connection
Subresource Integrity

Files
- HTML
- JS
- CSS

Servers
- Your server
- Compromised
  Third-party server

User's device

User

HTTPS
Subresource Integrity

<script src="https://thirdparty.example/framework.js" integrity="sha384oqVuAfXRKa+R9GqQ8K/ux" crossorigin="anonymous"></script>
Security

WKWebView
Safari Extensions
Subresource Integrity
Intelligent Tracking Prevention
Automatic Strong Passwords
Security Code AutoFill
Cookies can be used in a third-party context. Cookies are partitioned and not persisted in a third-party context. Existing cookies are purged and new cookies are blocked.

Days after the most recent interaction with the website:

- 0 days
- 1 day
- 30 days
Days after the most recent interaction with the website:

- Cookies are partitioned and not persisted in a third-party context
- Existing cookies are purged and new cookies are blocked

0 days
30 days
Days after the most recent interaction with the website

- Cookies are partitioned and not persisted in a third-party context
- Existing cookies are purged and new cookies are blocked

0 days

30 days

Or successful use of the Storage Access API
function makeRequestWithUserGesture() {
    let promise = document.requestStorageAccess();
    promise.then(function () {
        // Storage access was granted.
        // Check whether the user is logged in.
        // If not, do a popup to log the user in.
    }, function () {
        // Storage access was denied.
    });
}
Security

WKWebView
Safari Extensions
Subresource Integrity
Intelligent Tracking Prevention
Automatic Strong Passwords
Security Code AutoFill
## AutoFill for Safari

<table>
<thead>
<tr>
<th>Input Field</th>
<th>autocomplete =</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>username</td>
</tr>
<tr>
<td>Sign Up password</td>
<td>new-password</td>
</tr>
<tr>
<td>Sign Up / Confirm password</td>
<td>new-password</td>
</tr>
<tr>
<td>Sign In password</td>
<td>current-password</td>
</tr>
</tbody>
</table>
Default Password Format

fregat-Svuzen-pqbie2

20 characters long

Contain upper-case, digits, hyphen, and lower-case

Use passwordrules attribute for custom password rules

Password rules validation tool
Security

- WKWebView
- Safari Extensions
- Subresource Integrity
- Intelligent Tracking Prevention
- Automatic Strong Passwords
- Security Code AutoFill
verify

Please check your messages for a six-digit security code and enter it below.

Submit

From Messages
180605

Done

q w e r t y u i o p
a s d f g h j k l
z x c v b n m
## AutoFill for Safari

<table>
<thead>
<tr>
<th>Input Field</th>
<th>autocomplete =</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>username</td>
</tr>
<tr>
<td>Sign Up password</td>
<td>new-password</td>
</tr>
<tr>
<td>Sign Up / Confirm Password</td>
<td>new-password</td>
</tr>
<tr>
<td>Sign In password</td>
<td>current-password</td>
</tr>
</tbody>
</table>

Automatic Strong Passwords and Security Code AutoFill

WWDC 2018
## AutoFill for Safari

<table>
<thead>
<tr>
<th>Input Field</th>
<th>autocomplete =</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>username</td>
</tr>
<tr>
<td>Sign Up password</td>
<td>new-password</td>
</tr>
<tr>
<td>Sign Up / Confirm Password</td>
<td>new-password</td>
</tr>
<tr>
<td>Sign In password</td>
<td>current-password</td>
</tr>
<tr>
<td>Security Code AutoFill</td>
<td>one-time-code</td>
</tr>
</tbody>
</table>

Automatic Strong Passwords and Security Code AutoFill

WWDC 2018
Security

WKWebView
Safari Extensions
Subresource Integrity
Intelligent Tracking Prevention
Automatic Strong Passwords
Security Code AutoFill
Performance
Performance

Font Collections

font-display Descriptor

Videos in Image Elements

Passive Event Listeners

Async Image Decoding

Beacon API
Font Collections
Similar Unicode fonts downloaded more quickly
Performance

Font Collections
font-display Descriptor
Videos in Image Elements
Passive Event Listeners
Async Image Decoding
Beacon API
Welcome to my Site!

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Photography Portfolio
font-display Desciptor

@font-face {
  font-family: ExampleFont;
  src: url(/path/to/fonts/myfont.woff) format('woff');
  font-weight: 400;
  font-style: normal;
  font-display: fallback;
}
Performance

Font Collections
font-display Descriptor
Videos in image elements
Passive Event Listeners
Async Image Decoding
Beacon API
Videos in Image Elements
MP4 with H.264 encoding

`<img src="explosion.mp4" alt="Color Explosion">`

In CSS background image property:

```
<style>
    body {
        background-image: url("explosion.mp4");
    }
</style>
```
Videos in Image Elements

Fallback images

```html
<picture>
  <source srcset="explosion.mp4" type="video/mp4">
  <img src="explosion.jpg">
</picture>
```
Performance

Font Collections
font-display Descriptor
Videos in Image Elements
Passive event listeners
Async Image Decoding
Beacon API
Passive Event Listeners

Enabled on `document`, `window`, and `body` elements by default

Flag to always allow scrolling

No interruptions

document.addEventListener('touchstart', (e) => {
    ...
}, { passive: true })
Performance

Font collections
font-display descriptor
Videos in image elements
Passive event listeners
Async image decoding
Beacon API
Image Decoding

Synchronous
• Main thread is blocked
• User interactions prevented until all images are decoded

Asynchronous
• On separate thread
• Parallel decoding operations
• User interactions aren't blocked
Image Decoding

Synchronous
- Main thread is blocked
- User interactions prevented until all images are decoded

Asynchronous
- On separate thread
- Parallel decoding operations
- User interactions aren't blocked
- Default on first page load
Async Image Decoding
Async Image Decoding
Async Image Decoding

Markup: Add the `decoding` attribute to an element

JavaScript: Use `HTMLImageElement.decode()`
Performance

Font Collections

font-display Descriptor

Videos in Image Elements

Passive Event Listeners

Async Image Decoding

Beacon API
Beacon API
Asynchronous request on unload event

Sends data on unload without waiting for a response

Smooth browser navigation while waiting for response

Ensures delivery of data while Safari is running
Beacon API
Asynchronous request on unload event

Sends data on unload without waiting for a response

Smooth browser navigation while waiting for response

Ensures delivery of data while Safari is running
Building your first birdhouse.

As spring turns to summer, the fine feathered friends in your yard will appreciate a birdhouse. It’s fun, easy, and a great way to dip your toe into the world of woodworking!

Prepare your materials

See the list of tools and materials below. Make sure you have everything you need! Measure according to the blueprint. As long as the sides are even, feel free to make changes! Be creative!
Performance

Font collections
font-display descriptor
Videos in image elements
Passive event listeners
Async image decoding
Beacon API
Rich Experience
Rich Experience

Drag and drop
Payment Request API
Service Worker API
Fullscreen API on iPad
AR Quick Look + Safari
watchOS
Drag and Drop Improvements

Directory upload

Read and write support for MIME types (HTML, plaintext, and URLs)

iOS: `DataTransfer.getData` and `DataTransfer.setData`
Rich Experience

• Drag and drop
• Payment Request API
• Service Worker API
• Fullscreen API on iPad
• AR Quick Look + Safari
• watchOS
Payment Request API + Apple Pay
Payment Request API + Apple Pay

Support for Apple Pay using Payment Request API
Support for Apple Pay using Payment Request API

Still need to support Apple Pay specifics
Payment Request API + Apple Pay

Support for Apple Pay using Payment Request API

Still need to support Apple Pay specifics

Apple Pay JS features
- Granular error handling
- Cobranded card support
- Phonetic names
Rich Experience

Drag and drop
Payment Request API
Service Worker API
Fullscreen API on iPad
AR Quick Look + Safari
watchOS
Service Worker API

Great offline experience
- Cache interactions
- Intercept requests

Multiple tabs share one instance

Persistent store of resources
Rich Experience

• Drag and drop
• Payment Request API
• Service Worker API
• Fullscreen API on iPad
• AR Quick Look + Safari
• watchOS
The Earth is our home. There's no place like home. Oh... what a world it is! Our Earth is a jewel shimmering in the sequined velvet black void. Play the video above in fullscreen mode and marvel.
MY EARTH BLOG

The Earth is our home. There's no place like home. Oh... what a world it is! Our Earth is a jewel shimmering in the sequined velvet black void. Play the video above in fullscreen mode and marvel.
Fullscreen API for iPad
CSS Environment Variables

fullscreen-inset-top
fullscreen-auto-hide-delay
Fullscreen API for iPad
CSS Environment Variables

fullscreen-inset-top
fullscreen-auto-hide-delay
Fullscreen API for iPad
CSS Environment Variables

fullscreen-inset-top
fullscreen-auto-hide-delay
Rich Experience

Drag and drop
Payment Request API
Service Worker API
Fullscreen API on iPad
AR Quick Look + Safari
watchOS
AR Quick Look + Safari

```html
<a rel="ar" href="myfile.usdz">
  <img src="myimagefallback.jpg">
</a>
```
Integrating Apps and Content with AR Quick Look

WWDC 2018
Rich Experience

- Drag and drop
- Payment Request API
- Service Worker API
- Fullscreen API on iPad
- AR Quick Look + Safari
- watchOS
WebKit is the web browser engine used by Safari, Mail, App Store, and many other apps on macOS, iOS, and Linux. Get started contributing code, or reporting bugs.

Web developers can follow development, check feature status, download Safari Technology Preview to try out the latest web technologies, and report bugs.

Release Notes for Safari Technology Preview 56
Safari Technology Preview Release 56 is now available for download for macOS Sierra and macOS

Visualizing Layers in Web Inspector
Recent releases of Safari Technology Preview contain a new experimental feature for Web
WebKit

A fast, open source web browser engine.

WebKit is the web browser engine used by Safari, Mail, App Store, and many other apps on macOS, iOS, and Linux. Get started contributing code, or reporting bugs.

Web developers can follow development, check feature status, download Safari Technology Preview to try out the latest web technologies, and report bugs.
WebKit

A fast, open source web browser engine.

WebKit is the web browser engine used by

webkit.org
Demo

Jason Sandmeyer, Marketing Communications
Rich Experience

Drag and drop
Payment Request API
Service Worker API
Fullscreen API on iPad
AR Quick Look + Safari
watchOS
More Information

https://developer.apple.com/wwdc18/234