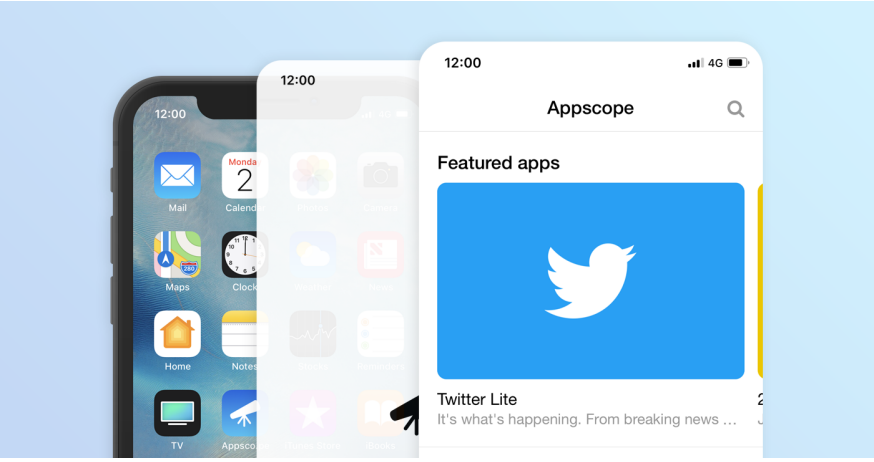
Appscope Follow
Jul 5 · 7 min read

Designing Native-Like Progressive Web Apps For iOS



One of the main characteristics of a Progressive Web App (PWA) is app-likeness. Although your app is technically run in the web browser, you should strive to make it look—and feel—as good as a native application. That includes making it installable on the home screen, adding a custom icon, disabling the address bar, and so on. Unlike Android, for which many native-like features are automatically generated by the Web App Manifest, iOS requires some additional HTML and CSS tricks. Here are seven suggestions on how to make your PWA more native-like on iOS.

1. Make it standalone

There are two ways to make your PWA run as a standalone application (that is, in a new window without the web browser's UI controls) on iOS. The first way is to use the `apple-mobile-web-app-capable` meta tag in the head element of your HTML code with the following code.

```
<meta name="apple-mobile-web-app-capable" content="yes">
```

The second way is to set the `display` property of your Web App Manifest to `standalone` . An example Web App Manifest could look

like the following.

```
{
   "name": "Appscope",
   "display": "standalone",
   "icons": [{
      "src": "icons/icon-192.png",
      "sizes": "192x192"},
   {
      "src": "icons/icon-512.png",
      "sizes": "512x512"}
   ]
}
```

2. Add a custom icon

Unfortunately, iOS does not use the icons specified in the Web App Manifest. Instead, to use a custom icon for all pages of your app, you must provide your icon in PNG format and with the name `apple-touch-icon.png` in the root document folder of your PWA.

If you want to add a specific icon for a *single* page of your app, use the following line in the head element of your HTML code.

```
<link rel="apple-touch-icon" href="single-page-icon.png">
```

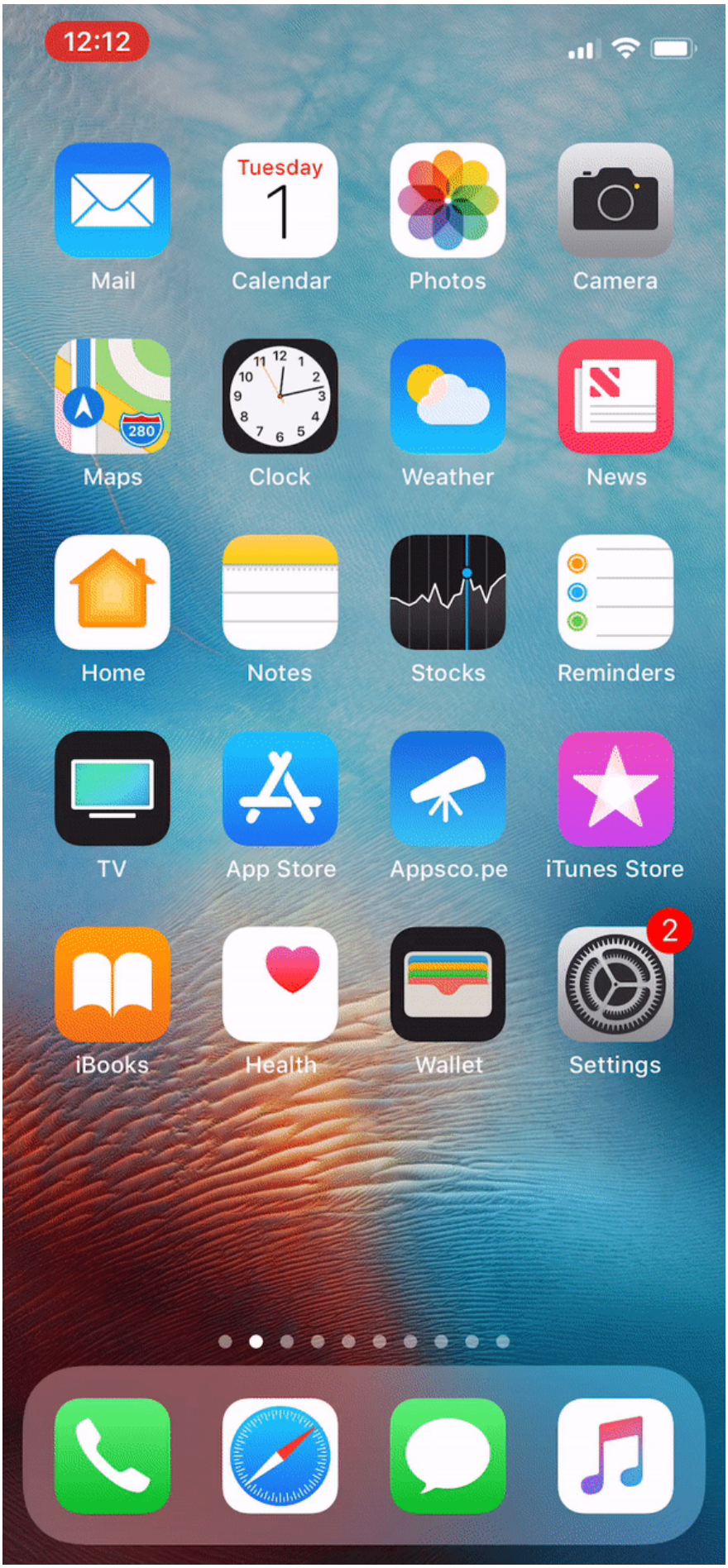
Different iOS devices use different sizes for the home screen icons. To specify an icon for a particular size, use the `sizes` attribute of the link element. If no icon is specified for the recommended icon size for the device, the icon with the smallest size larger than the recommended one is used instead.

The following example, which specifies sizes for the most common iOS devices, is from Apple's Safari Web Content Guide.

```
<link rel="apple-touch-icon" href="touch-icon-iphone.png">
<link rel="apple-touch-icon" sizes="152x152" href="touch-
icon-ipad.png">
<link rel="apple-touch-icon" sizes="180x180" href="touch-
icon-iphone-retina.png">
<link rel="apple-touch-icon" sizes="167x167" href="touch-
icon-ipad-retina.png">
```

Recall that iOS requires opaque icons. Any transparent parts of the icon will be colored black.

3. Add a custom splash screen


12:12
Mail
Tuesday
1
Calendar
Photos
Camera
Maps
Clock
Weather
News
Home
Notes
Stocks
Reminders
TV
App Store
Appsco.pe
iTunes Store
iBooks
Health
Wallet
2
Settings

Splash screen for Appscope

The next step to make your Progressive Web App more native-like is to replace the dull, white launch screen with your own image. To add a custom splash screen, use the following link element.

```
<link rel="apple-touch-startup-image" href="launch.png">
```

In order for this image to show, it is important that its dimensions are the same as those of the device the app is run on. For example, to work on an iPhone X, `launch.png` would have to be of the size 1125 by 2436 pixels. The problem that arises here is that there are multiple iOS devices with different resolutions, and unfortunately we cannot just simply repeat this code multiple times for images of different sizes. Instead, we need to use the `media` attribute to specify which launch image is intended for which device.

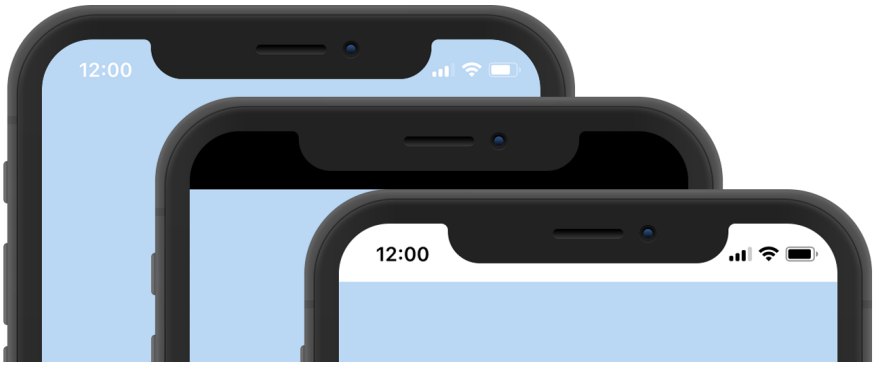
Add the following code to the head element of your PWA to support custom splash screens for the different iOS devices.

```
<!-- iPhone X (1125px x 2436px) -->
<link rel="apple-touch-startup-image" media="(device-width:
375px) and (device-height: 812px) and (-webkit-device-pixel-
ratio: 3)" href="/apple-launch-1125x2436.png">

<!-- iPhone 8, 7, 6s, 6 (750px x 1334px) -->
<link rel="apple-touch-startup-image" media="(device-width:
375px) and (device-height: 667px) and (-webkit-device-pixel-
ratio: 2)" href="/apple-launch-750x1334.png">

<!-- iPhone 8 Plus, 7 Plus, 6s Plus, 6 Plus (1242px x
2208px) -->
<link rel="apple-touch-startup-image" media="(device-width:
414px) and (device-height: 736px) and (-webkit-device-pixel-
ratio: 3)" href="/apple-launch-1242x2208.png">

<!-- iPhone 5 (640px x 1136px) -->
<link rel="apple-touch-startup-image" media="(device-width:
320px) and (device-height: 568px) and (-webkit-device-pixel-
ratio: 2)" href="/apple-launch-640x1136.png">

<!-- iPad Mini, Air (1536px x 2048px) -->
<link rel="apple-touch-startup-image" media="(device-width:
768px) and (device-height: 1024px) and (-webkit-device-
pixel-ratio: 2)" href="/apple-launch-1536x2048.png">
```

```
<!-- iPad Pro 10.5" (1668px x 2224px) -->
<link rel="apple-touch-startup-image" media="(device-width:
834px) and (device-height: 1112px) and (-webkit-device-
pixel-ratio: 2)" href="/apple-launch-1668x2224.png">

<!-- iPad Pro 12.9" (2048px x 2732px) -->
<link rel="apple-touch-startup-image" media="(device-width:
1024px) and (device-height: 1366px) and (-webkit-device-
pixel-ratio: 2)" href="/apple-launch-2048x2732.png">
```

If you need help setting up splash screens for your PWA, check out the Splash Screen Generator at Appscope.

4. Change the status bar



Status bars with settings black-translucent, black, and default

You may also customize the iOS status bar (the area along to upper edge of the screen displaying the time and battery status) of your PWA. In order to do this, you must use the `apple-mobile-web-app-status-bar-style` meta tag in the head element of your code.

```
<meta name="apple-mobile-web-app-status-bar-style"
content="default">
```

Unfortunately, the number of ways to customize the status bar is fairly limited, but Apple offers three distinct settings for the `content` attribute.

- `default` results in a white status bar with black text and symbols.
- `black` results in a black status bar and black text and symbols, making it appear completely black. If you do not use the status bar

meta tag, this is what status bar will look like.

- `black-translucent` results in white text and symbols, and the status bar will take the same background color as the body element of your web app.

5. Give it a short name

The title of your PWA will be shown below its launch icon on the home screen. To avoid truncation, this title should not be longer than 12 characters, although it ultimately comes down to the width of the characters used (for example, the letter *w* is wider than the letter *i*.) If the original name of your PWA does not fit below the icon, you may assign a short version of the name.

One way to specify a short name for your PWA is to use the `apple-mobile-web-app-title` meta tag with the following line in the head element of your code.

```
<meta name="apple-mobile-web-app-title" content="Appscope">
```

Another way is to use the `short_name` property in your Web App Manifest. An example Web App Manifest could look like the following.

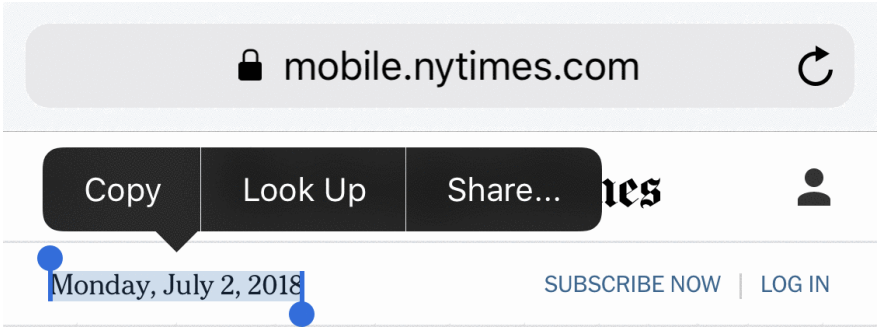
```
{
   "name": "Little Alchemy 2",
   "short_name": "Alchemy 2",
   "icons": [{
      "src": "/public/icons/icon-192x192.png",
      "sizes": "192x192"},
    {
      "src": "/public/icons/icon-512x512.png",
      "sizes": "512x512"}
   ]
}
```

6. Disable selection, highlighting, and callouts

On default, the iOS web browser adds certain interactive effects for texts and links that native applications do not have. Therefore, to make your PWA feel more native-like—and less like a website or document—

you may disable (or at least partially disable) these effects. The following subsections target the three most common types of effects.

6.1 Disable text selection

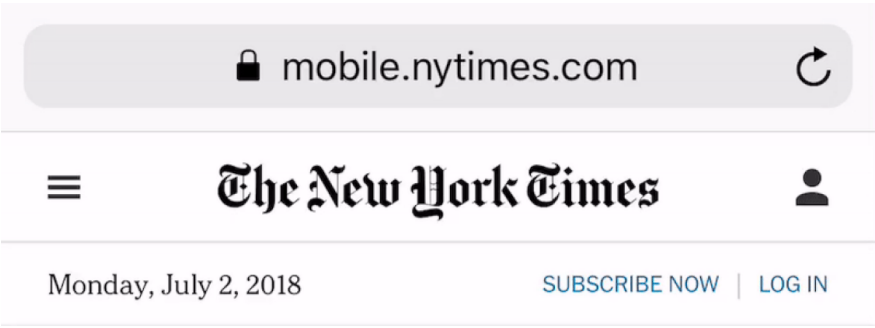


Text-selection on New York Times

Just like most native iOS applications do not allow text selection, you may disable this feature in your PWA. To do this, set the `-webkit-user-select` CSS property to `none` for the elements you do not want selectable. To turn off text selection completely, assign the property to your body element.

```
body {
   -webkit-user-select: none;
}
```

6.2 Disable highlighting



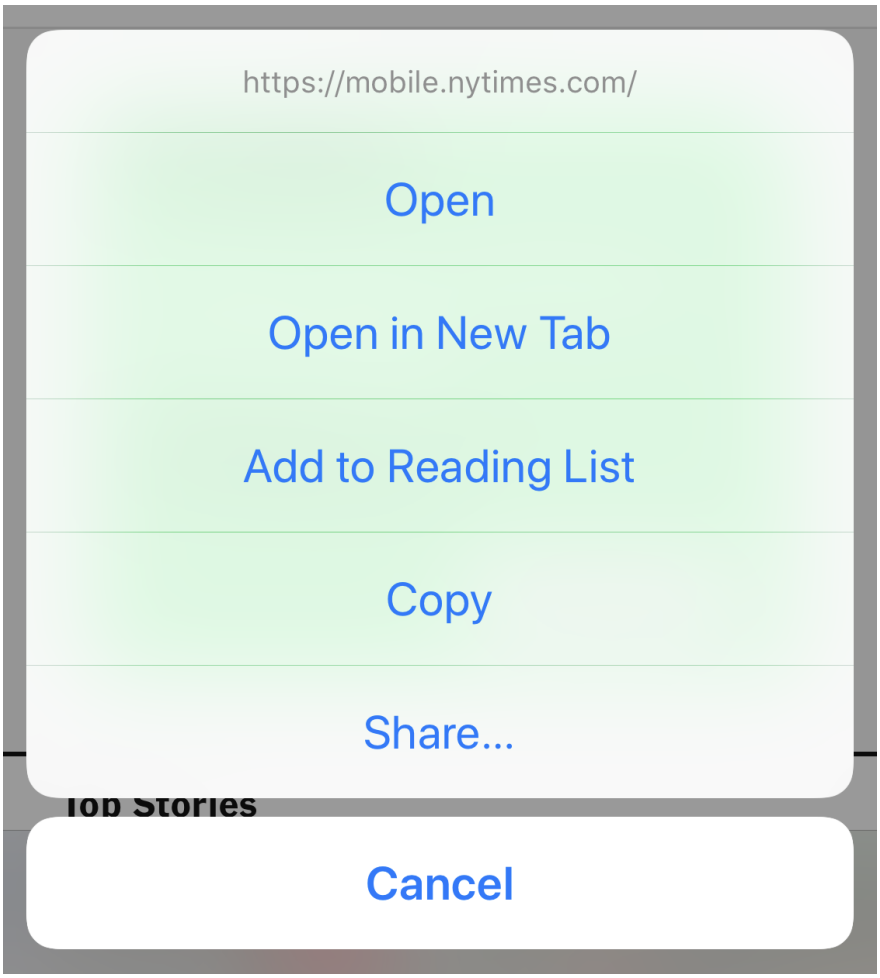
Link highlighting on New York Times

When tapping a link in the iOS web browser, a grey box appears around the element. Although there is no simple way of disabling this effect, you may change the highlighting color to transparent, and thus effectively make it disappear. To do this for your PWA, set the `-webkit-`

`tap-highlight-color` property to `transparent` for the desired elements (or assign it to the body element to disable link highlighting for all elements.)

```
body {
   -webkit-tap-highlight-color: transparent;
}
```

6.3 Disable callouts

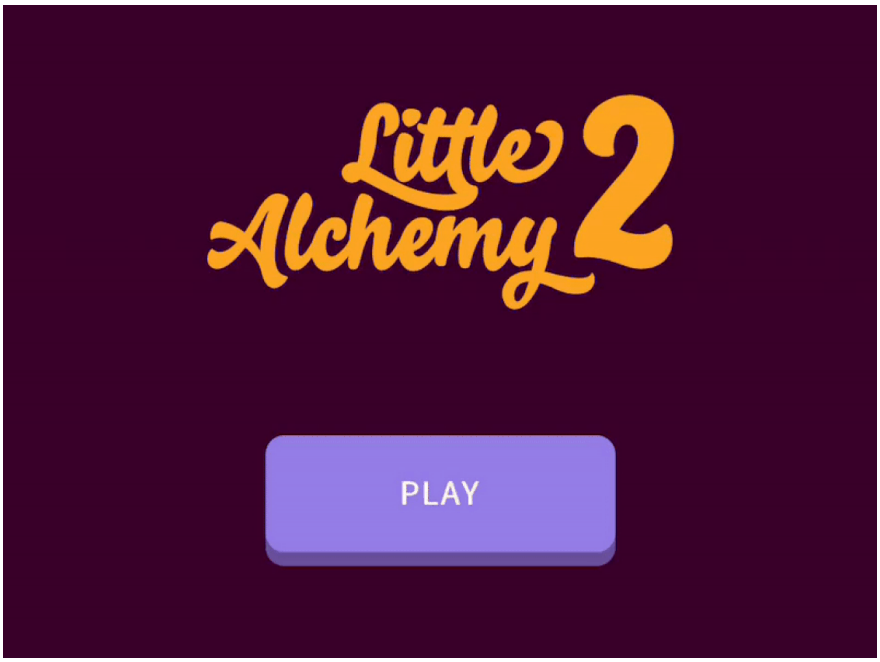


Link callout on New York Times

If you tap and hold on an element in the iOS browser, it will open a callout menu (like the one above.) To disable this effect on iOS, set the `-webkit-touch-callout` property to `none` for the desired elements. Again, to disable the effect for all elements, assign the property to your body element.

```
body {
   -webkit-touch-callout: none;
}
```

7. Enable tap effects



Tap effect (:active) on Little Alchemy 2

Instead of using the default gray highlighting when tapping a link, you may add your own on-tap effects. By including the `ontouchstart` attribute in the body tag of your code and keeping its value empty, links and other elements will display their :hover and :active effects when tapped. Use the following code and play around with different :hover and :active styles to find the effects that work best for your PWA.

```
<html>
   <head>
      ...
   </head>
   <body ontouchstart="">
      ...
   </body>
</html>
```

. . .