

# Configuring Web Applications

A web application is designed to look and behave in a way similar to a native application—for example, it is scaled to fit the entire screen on iOS. You can tailor your web application for Safari on iOS even further, by making it appear like a native application when the user adds it to the Home screen. You do this by using settings for iOS that are ignored by other platforms.

For example, you can specify an icon for your web application used to represent it when added to the Home screen on iOS, as described in [Specifying a Webpage Icon for Web Clip](#). You can also minimize the Safari on iOS user interface, as described in [Changing the Status Bar Appearance and Hiding Safari User Interface Components](#), when your web application is launched from the Home screen. These are all optional settings that when added to your web content are ignored by other platforms.

Read [Viewport Settings for Web Applications](#) for how to set the viewport for web applications on iOS.

## Specifying a Webpage Icon for Web Clip

You may want users to be able to add your web application or webpage link to the Home screen. These links, represented by an icon, are called Web Clips. Follow these simple steps to specify an icon to represent your web application or webpage on iOS.

- To specify an icon for the entire website (every page on the website), place an icon file in PNG format in the root document folder called `apple-touch-icon.png`
- To specify an icon for a single webpage or replace the website icon with a webpage-specific icon, add a link element to the webpage, as in:

```
<link rel="apple-touch-icon" href="/custom_icon.png">
```

In the above example, replace `custom_icon.png` with your icon filename.

- To specify multiple icons for different device resolutions—for example, support both iPhone and iPad devices—add a `sizes` attribute to each link element as follows:

```
<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">
```

The icon that is the most appropriate size for the device is used. See the “Graphics” chapter of *iOS Human Interface Guidelines* for current icon sizes and recommendations.

If there is no icon that matches the recommended size for the device, the smallest icon larger than the recommended size is used. If there are no icons larger than the recommended size, the largest icon is used.

If no icons are specified using a link element, the website root directory is searched for icons with the `apple-touch-icon...` prefix. For example, if the appropriate icon size for the device is 58 x 58, the system searches for filenames in the following order:

1. `apple-touch-icon-80x80.png`
2. `apple-touch-icon.png`

**Note:** Safari on iOS 7 doesn't add effects to icons. Older versions of Safari will not add effects for icon files named with the `-precomposed.png` suffix. See *First Steps: Identifying Your App in iTunes Connect* for details.

## Specifying a Launch Screen Image

On iOS, similar to native applications, you can specify a launch screen image that is displayed while your web application launches. This is especially useful when your web application is offline. By default, a screenshot of the web application the last time it was launched is used. To set another startup image, add a link element to the webpage, as in:

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

In the above example, replace `launch.png` with your startup screen filename. See the “Graphics” chapter of *iOS Human Interface Guidelines* for current launch screen sizes and recommendations.

## Adding a Launch Icon Title

On iOS, you can specify a web application title for the launch icon. By default, the `<title>` tag is used. To set a different title, add a meta tag to the webpage, as in:

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

In the above example, replace `AppTitle` with your title.

## Hiding Safari User Interface Components

On iOS, as part of optimizing your web application, have it use the standalone mode to look more like a native application. When you use this standalone mode, Safari is not used to display the web content—specifically, there is no browser URL text field at the top of the screen or button bar at the bottom

of the screen. Only a status bar appears at the top of the screen. Read [Changing the Status Bar Appearance](#) for how to minimize the status bar.

Set the `apple-mobile-web-app-capable` meta tag to `yes` to turn on standalone mode. For example, the following HTML displays web content using standalone mode.

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

You can determine whether a webpage is displaying in standalone mode using the `window.navigator.standalone` read-only Boolean JavaScript property. For more on standalone mode, see [apple-mobile-web-app-capable](#).

## Changing the Status Bar Appearance

If your web application displays in standalone mode like that of a native application, you can minimize the status bar that is displayed at the top of the screen on iOS. Do so using the `status-bar-style` meta tag.

This meta tag has no effect unless you first specify standalone mode as described in [Hiding Safari User Interface Components](#). Then use the status bar style meta tag, `apple-mobile-web-app-status-bar-style`, to change the appearance of the status bar depending on your application needs. For example, if you want to use the entire screen, set the status bar style to translucent black.

For example, the following HTML sets the background color of the status bar to black:

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

For more on status bar appearance, see the “UI Bars” chapter of *iOS Human Interface Guidelines*.

## Linking to Other Native Apps

Your web application can link to other built-in iOS apps by creating a link with a special URL. Available functionality includes calling a phone number, sending an SMS or iMessage, and opening a YouTube video in its native app if it is installed. For example, to link to a phone number, structure an anchor element in the following format:

```
<a href="tel:1-408-555-5555">Call me</a>
```

For a complete look of these capabilities, see [Apple URL Scheme Reference](#).