

[App Clips](#) / Associating your App Clip with your website

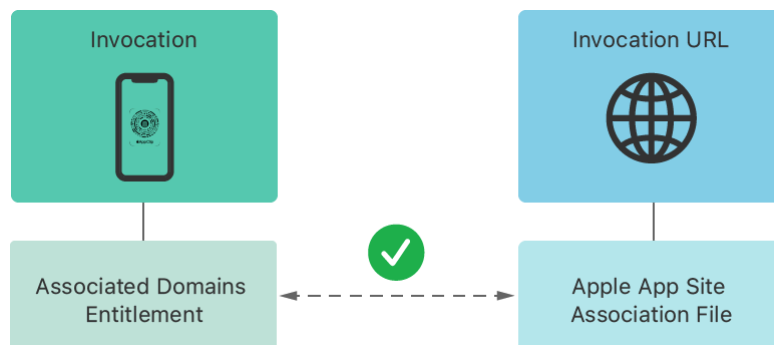
Article

Associating your App Clip with your website

Enable the system to verify your App Clip to support invocations from your website and devices running iOS 16.3 or earlier.

Overview

An App Clip gives people quick access to a particular workflow in your app, even when a person hasn't installed your app. NFC readers, App Clip Codes, or QR codes define an *invocation URL* that specifies which App Clip, or workflow within your full app, the system needs to run. If you want to support invocations from your website or support iOS 16.3 and earlier, enable the system to verify your App Clip. The system's verification checks that the App Clip includes the URL in its code signature as the Associated Domains Entitlement, which cites the invocation URL's domain. The system also verifies that the server of the domain agrees to launch the App Clip, by citing the App Clip within an Apple App Site Association (AASA) file that it hosts.



Note

If you don't support iOS 16.3 and earlier, don't plan to support all invocations, and don't have a website to associate with your App Clip, you may be able to keep your invocations to the default URL that App Store Connect generates. For more information, refer to [Configuring App Clip experiences](#).

To associate your app and App Clip with your website:

- Specify your invocation URL's domain within an [Associated Domains Entitlement](#) on both your app and App Clip targets in Xcode.
- Add or modify an AASA file on the domain's server.

The system verifies that both the entitlement and the configuration in the AASA file match before it permits the invocation of the App Clip. App Store Connect also verifies the match when you create an App Clip experience; for more information, refer to [Set up an App Clip experience](#).

Tip

You may already be familiar with the Associated Domains Entitlement if your app supports [Handoff](#) or [universal links](#). If you're new to using this entitlement and universal links, read [Allowing apps and websites to link to your content](#) and [Supporting universal links in your app](#). For additional information about the Associated Domains Entitlement — including cache policies — read [Supporting associated domains](#).

Add the Associated Domains entitlement

To associate your App Clip with your website, you must add the [Associated Domains Entitlement](#) to the app and the App Clip targets.

First, open your project in Xcode; then, in your project settings, enable the Associated Domains capability to add the [Associated Domains Entitlement](#).

Second, for each URL that launches your App Clip or full app, add its domain to the Associated Domains capability using the pattern: `appclips:<fully qualified domain>`. For example, add `appclips:example.com` or `appclips:appclip.example.com`. Make sure to only include the desired subdomain and the top-level domain. Don't include a trailing slash (/), wildcard (*), or path and query components in the URL. For more information, refer to [Supporting associated domains](#).

Make changes to your server

In addition to adding the [Associated Domains Entitlement](#) to your Xcode project, you need to make changes to your server to associate your App Clip with your server and allow the system to verify the URL that tries to invoke your App Clip.

First, create an AASA file as described in [Supporting associated domains](#). Next, add an entry for the App Clip with the `appclips` key to the file.

The following code shows the content to add. Note how the value for the `apps` key is an array that contains the app identifier of the App Clip. In many cases, the array contains only one entry. However, it can contain entries for multiple App Clips.

```
{
  "appclips": {
    "apps": [ "ABCDE12345.com.example.MyApp.Clip" ]
  }
  ...
}
```

Important

For apps that detect App Clip Codes in AR, add an entry for the parent app identifier. For more information, refer to [Interacting with App Clip Codes in AR](#).

Then, add the AASA file to your website's `.well-known` directory. If you previously added an AASA file to your server, add the entry for the `appclips` key to the existing file.

Note

If you plan to use multiple invocation URLs with different domains, remember to add an AASA file to each domain's `.well-known` directory. In addition, remember to add each domain to the [Associated Domains Entitlement](#).

Finally, to make sure the system can validate the association between your App Clip and the AASA file on your server, check your server's configuration and make sure it allows AASA-Bot and CFNetwork as user agents.

Check the validation status of your App Clip

App Store Connect verifies the AASA file configuration of your App Clip after you've uploaded a build to App Store Connect and created an App Clip experience. To check the verification status:

1. Open [App Store Connect](#) in your browser and navigate to a build's details page.
2. Click View Status in the App Clip section to show the domain validation status. It shows the validation status for each domain that's associated with your App Clip.

For example, you could configure the default App Clip experience to use `https://example.com` as its invocation URL and configure an advanced App Clip experience to use `https://appclip.example.com`. In this example, you'd place an AASA file in the well-known directories for each URL's domain, and App Store Connect would show the verification status for both domains.

The Cache Status column shows the validation status for your App Clip as the system performs the validation on people's devices. As you develop your App Clip, you may make frequent changes to your AASA file. To check the verification status in real time, click Load Debug Status in the modal view that shows the verification status of your App Clip. If a configuration error occurs, App Store Connect shows information about the error in the Debug Status column.

For more information, refer to [WWDC20: What's New in App Store Connect](#).

See Also

Launch



Responding to invocations

Add code to respond to invocations and offer a focused launch experience.



Supporting invocations from your website and the Messages app

Display a Smart App Banner and the App Clip card on your website that people tap to launch your App Clip, and add support for invocations from the Messages app.



Confirming a person's physical location

Add code to quickly confirm a person's physical location while respecting their privacy.



Launching another app's App Clip from your app

Enable people to launch another app's App Clip from your app with App Clip links and offer a rich preview of it with the Link Presentation framework.

`class` `APActivationPayload`

Information that's passed to an App Clip on launch.

`NSAppClip`

A collection of keys that an App Clip uses to get additional capabilities.