

[StoreKit](#) / Requesting App Store reviews

Sample Code

Requesting App Store reviews

Implement best practices for prompting users to review your app in the App Store.

[Download](#)

iOS 17.0+ | iPadOS 17.0+ | macOS 14.0+ | Xcode 15.0+

Overview

Present your users with a request for an App Store review using [RequestReviewAction](#) to get feedback on your app. Be aware that the system displays the review prompt to a user a maximum of three times within a 365-day period. You can determine when and where your app displays the prompt to request a review. Consider the best places and conditions in your app to show the request, and when it might be appropriate to delay it. Here are some best practices:

- Make the request at a time that doesn't interrupt what someone is trying to achieve in your app, for example, at the end of a sequence of events that they successfully complete.
- Avoid showing a request for a review immediately when a user launches your app, even if it isn't the first time it launches.
- Avoid requesting a review as the result of a user action.

Also, remember that people can disable requests for reviews from ever appearing on their device.

Present the review request

This sample project demonstrates a simulated three-step process. A person taps the Start Process button, and then taps Continue Process twice, after which the app presents a Process Completed scene. The request for review appears from this scene only.

In addition, the app and its use meets the following conditions before displaying the prompt:

- The app hasn't shown a review prompt for a version of the app bundle that matches the current bundle version. This ensures that a person doesn't receive a prompt to review the same version of an app multiple times.
- The person successfully completes the three-step process at least four times. This number is arbitrary and developers can choose something that fits well with how many times someone is likely to complete a process in their apps.
- A person must pause on the Process Completed scene for a few seconds. This requirement limits the possibility of the prompt interrupting them before they move to a different task in the app.

To present a review request, the app reads the `requestReview` environment value to get an instance of `RequestReviewAction` and calls it as a function:

```
@Environment(\.requestReview) private var requestReview
```

The conditions above exist purely to delay the call to `requestReview`, so days, weeks, or even months can elapse without the app prompting a user for a review.

```
/// Presents the rating and review request view after a two-second delay.
private func presentReview() {
    Task {
        // Delay for two seconds to avoid interrupting the person using the app.
        try await Task.sleep(for: .seconds(2))
        await requestReview()
    }
}
```

Techniques to delay the call are valuable because they cause an app to show a review request when people are more experienced at using the app and can provide better feedback.

```
/*
    The lastVersionPromptedForReview property stores the version of the app that last
    The app presents the rating and review request view if the person completed the
    its current version is different from the version that last prompted them for re
*/
if processCompletedCount >= 4, currentAppVersion != lastVersionPromptedForReview {
    presentReview()

    // The app already displayed the rating and review request view. Store this current
    lastVersionPromptedForReview = currentAppVersion
}
```

```
}
```

In the following code, the app stores the usage data that delays the review request in [App Storage](#):

```
/// An identifier for the three-step process the person completes before this app c
@AppStorage("processCompletedCount") var processCompletedCount = 0

/// The most recent app version that prompts for a review.
@AppStorage("lastVersionPromptedForReview") var lastVersionPromptedForReview = ""
```

In other apps, there might be more appropriate on-device storage options. For more information about best practices for requesting reviews, see [Ratings and reviews](#) in Human Interface Guidelines.

Manually request a review

To enable a person to initiate a review as a result of an action in the UI, the sample code uses a deep link to the App Store page for the app with the query parameter `action=write-review` appended to the URL:

```
// Replace the YOURAPPSTOREID value below with the App Store ID for your app.
// You can find the App Store ID in your app's product URL.
let url = "https://apps.apple.com/app/idYOURAPPSTOREID?action=write-review"

guard let writeReviewURL = URL(string: url) else {
    fatalError("Expected a valid URL")
}

openURL(writeReviewURL)
```

See Also

Reviews

```
struct RequestReviewAction
```

An instance that tells StoreKit to request an App Store rating or review, if appropriate.

```
class SKStoreReviewController
```

An object that controls the process of requesting App Store ratings and reviews from customers.

Deprecated