

[App Store Connect API](#) / Xcode Cloud Workflows and Builds

# Xcode Cloud Workflows and Builds

Automate reading Xcode Cloud data, managing workflows, and starting builds.

## Overview

[Xcode Cloud](#) makes it easy to get started with continuous integration and delivery (CI/CD). Using either Xcode or [App Store Connect](#), you can create custom workflows to verify the quality of your apps or frameworks. However, you may need to further automate Xcode Cloud usage and integrate Xcode Cloud into existing back-end systems — often in a corporate context. For example, you may need to:

- Provide decision makers with data about workflows and builds using custom dashboards or reports.
- Integrate with other CI/CD systems that verify changes to back-end services and apps for other platforms.
- Create and manage numerous workflows for multiple apps.

To automate Xcode Cloud usage and integrate it with custom back-end systems, use the App Store Connect API to:

- Read information about Xcode Cloud products, workflows, and builds.
- Manage workflows.
- Start new builds.

Taking advantage of the App Store Connect API, you can combine functionality offered by the Xcode Cloud resources and other App Store Connect resources — for example, TestFlight resources — to create powerful custom tools.

If you're new to using the App Store Connect API, make sure to read the documentation in the Essentials section of [App Store Connect API](#).

# Read Xcode Cloud Data

The App Store Connect API provides resources to access the following Xcode Cloud data:

- Products, workflows, and build environments
- Builds, artifacts, issues, and test results
- Source code management (SCM) providers, Git repositories, pull requests, and Git references

When you've identified the data you need to access, the next step is to identify the Xcode Cloud API endpoints you need to call. This is a customized task that depends on your use case. For example, you could follow these steps to read Xcode Cloud workflow information and display it in a custom dashboard:

1. Access Xcode Cloud product information by reading a list of objects that represent your Xcode Cloud products using the [List All Xcode Cloud Products](#) endpoint.
2. Choose the Xcode Cloud product for which you want to display information in the custom dashboard.
3. Pass the `id` of the Xcode Cloud product you chose to the [List All Workflows for an Xcode Cloud Product](#) endpoint and call it to read a list of workflow objects.
4. Call the [Read Xcode Cloud Workflow Information](#) endpoint for each workflow using the workflow's `id` to access detailed workflow information.
5. Populate your dashboard with the detailed workflow information.

## Start a New Build

Xcode Cloud workflows allow you to configure a custom start condition that queues new builds based on a schedule or in response to branch changes, pull request changes, or tag changes. You can also manually start a new build using Xcode and App Store Connect, but that might not be feasible for large teams or in a corporate context. For example, your company's back-end team likely uses a CI/CD process that frequently deploys changes to staging and production servers. As a result, you need to perform verifications in your app to ensure the app works well with the updated servers. Starting new builds in Xcode Cloud for each back-end change manually or based on a schedule might not be sufficient. To help with these scenarios, use the App Store Connect API to automatically start new Xcode Cloud builds with an API call.

To start a new build:

1. Read data necessary to create a [CiBuildRunCreateRequest](#); for example, access workflow information using endpoints provided by the [Products](#) and [Workflows](#) resources.
2. Create the [CiBuildRunCreateRequest](#) and perform the [Start a Build](#) operation that starts a new build.

# Manage Workflows

Xcode and App Store Connect offer a convenient way to create, update, or delete workflows. However, large teams and companies with many teams often work on multiple apps and frameworks in parallel. As a result, they need to maintain a large number of workflows, making additional automation necessary.

Using endpoints provided by the [Workflows](#) resource, you can:

- Create a new workflow.
- Update an existing workflow.
- Delete an existing workflow.

Depending on what you want to do, you likely need to first read Xcode Cloud data and use it to create the request body to perform the operation. For example, to create a new workflow, you'll need to read Xcode Cloud data using the following resources:

- [Products](#)
- [Repositories](#)
- [Xcode Versions](#)

After you've retrieved all necessary information, use it to create the [CiWorkflowCreate Request](#) and perform the [Create a Workflow](#) operation to create a new workflow.

---

## Topics

### Xcode Cloud Products and Workflows

#### ⌵ Products

Read information about the products Xcode Cloud detected or delete a product and all its associated information.

#### ⌵ Workflows

Manage Xcode Cloud workflows and view workflow details like actions and start conditions.

#### ⌵ macOS Versions

Read macOS version information you configure for an Xcode Cloud workflow.

#### ⌵ Xcode Versions

Read Xcode version information you configure for an Xcode Cloud workflow.

## Build Information

### ⌵ Build Runs

Read detailed build information and start new builds.

### ⌵ Build Actions

Read information about actions you configured for an Xcode Cloud workflow and their related data such as artifacts, issues, or test results.

### ⌵ Artifacts

Read information about artifacts Xcode Cloud creates when it performs a build.

### ⌵ Issues

Read information about issues that occurred when Xcode Cloud performs a build.

### ⌵ Test Results

Read test results for test actions Xcode Cloud performs.

## Source Code Management

### ⌵ Providers

Read information about source code management providers you connected to Xcode Cloud.

### ⌵ Repositories

Read detailed information for each repository Xcode Cloud can access, including Git references and pull requests.

### ⌵ Pull Requests

Read pull request information such as source and destination branches.

### ⌵ Git References

Read information about the canonical reference for a Git branch or tag.