Setting up HealthKit

Set up and configure your HealthKit store.

Overview

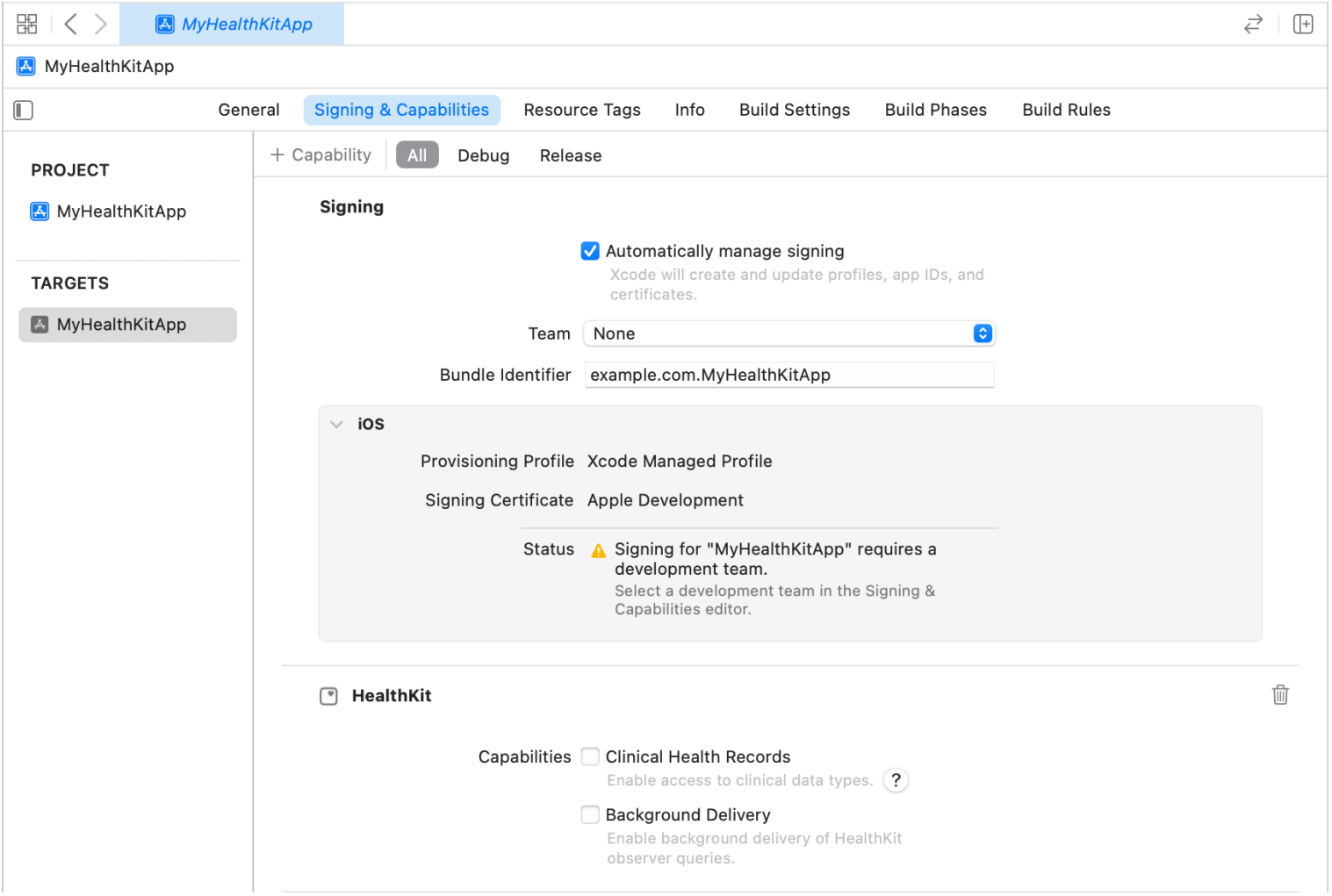
Before using HealthKit, you must perform the following steps:

1. Enable HealthKit in your app.
2. Ensure HealthKit is available on the current device.
3. Create your app's HealthKit store.
4. Request permission to read and share data.

The following sections describe the first three steps in detail. For more information on requesting authorization, see Authorizing access to health data. For a practical example of how to set up and use HealthKit, see Build a workout app for Apple Watch.

Enable HealthKit

Before you can use HealthKit, you must enable the HealthKit capabilities for your app. In Xcode, select the project and add the HealthKit capability. Only select the Clinical Health Records checkbox if your app needs to access the user's clinical records. App Review may reject apps that enable the Clinical Health Records capability if the app doesn't actually use the health record data. For more information, see Accessing Health Records.



For a detailed discussion about enabling capabilities, see Configuring HealthKit access.

When you enable the HealthKit capabilities on an iOS app, Xcode adds HealthKit to the list of required device capabilities, which prevents users from purchasing or installing the app on devices that don't support HealthKit.

If HealthKit isn't required for the correct operation of your app, delete the `healthkit` entry from the "Required device capabilities" array. Delete this entry from either the Target Properties list on the app's Info tab or from the app's `Info.plist` file.

> **Note**
>
> The `healthkit` entry isn't used by watchOS apps.

For more information on required device capabilities, see the `UIRequiredDeviceCapabilities`.

Ensure HealthKit's availability

Call the `isHealthDataAvailable()` method to confirm that HealthKit is available on the user's device.

```
if HKHealthStore.isHealthDataAvailable() {
    // Add code to use HealthKit here.
}
```

Call this method before calling any other HealthKit methods. If HealthKit isn't available on the device (for example, on iPadOS 16 or earlier, or macOS), other HealthKit methods fail with an `errorHealthDataUnavailable` error. If HealthKit is restricted (for example, in an enterprise environment), the methods fail with an `errorHealthDataRestricted` error.

Create the HealthKit store

If HealthKit is both enabled and available, instantiate an `HKHealthStore` object for your app as shown:

```
let healthStore = HKHealthStore()
```

You need only a single HealthKit store per app. These are long-lived objects; you create the store once, and keep a reference for later use.

Topics

Entitlements

`HealthKit Entitlement`

A Boolean value that indicates whether the app may request user authorization to access health and activity data that appears in the Health app.

`HealthKit Capabilities Entitlement`

Health data types that require additional permission.

Information property list keys

`NSHealthUpdateUsageDescription`

A message to the user that explains why the app requested permission to save samples to the HealthKit store.

`NSHealthShareUsageDescription`

A message that explains to people why the app requests permission to read samples from the HealthKit store.

`NSHealthRequiredReadAuthorizationTypeIdentifiers`

The clinical record data types that your app must get permission to read.

`NSHealthClinicalHealthRecordsShareUsageDescription`

A message to the user that explains why the app requested permission to read clinical records.

See Also

Essentials

- About the HealthKit framework
  Learn about the architecture and design of the HealthKit framework.
- Authorizing access to health data
  Request permission to read and share data in your app.
- Protecting user privacy
  Respect and safeguard your user's privacy.
- HealthKit updates
  Learn about important changes to HealthKit.
- HealthKitUI
  Display user interface that enables a person to view and interact with their health data.