EventKit / Creating events and reminders
Article

Creating events and reminders

Create and modify events and reminders in a person's database.

Overview

Once you have permission to access a person's Calendar and Reminder data, you can create, display, and edit events and reminders.

Create Events

Create a new event with the `init(eventStore:)` method of the `EKEvent` class.

You can edit the details of a new event or an event you previously fetched from the Calendar database by setting the event's corresponding properties. Some of the details you can edit include:

- The event's title with the `title` property.
- The event's start and end dates with the `startDate` and `endDate` properties.
- The calendar with which the event is associated with the `calendar` property.
- The alarms associated with the event with the `alarms` property (see "Setting an alarm" for more details).
- The event's recurrence rule, if it is a repeating event, with the `recurrenceRules` property (see "Creating a recurring event" for more details).

> Note
> In iOS, you have the option of letting users modify event data with the event view controllers provided in the EventKit UI framework. For information on how to use these event view controllers, see EventKit UI.

Save and Delete Events

> Important
> If your app modifies a user's Calendar database, it must get confirmation from the user before doing so. An app should never modify the Calendar database without specific instruction from the user.

Save your changes to the Calendar database with the `EKEventStore` method `save(: span:commit:)`. If you want to remove an event from the Calendar database, use the `EKEventStore` method `remove(: span: commit:)`. Whether you are saving or removing an event, implementing the respective method automatically syncs your changes with the calendar the event belongs to (CalDAV, Exchange, and so on).

If you are saving a recurring event, your changes can apply to all future occurrences of the event by specifying `EKSpan.futureEvents` for the span parameter of the `save(: span:commit:)` method. Likewise, you can remove all future occurrences of an event by specifying `EKSpan.futureEvents` for the span parameter of the `remove(_:span:commit:)` method.

> Note
> If you pass NO to the `commit` parameter, make sure that you later invoke the `commit()` method to permanently save your changes.

Create Reminders

Reminders are tasks that may be tied to a specific time or location. They are similar to calendar events, but can be marked complete and may not necessarily span an exact period of time.

Because `EKReminder` inherits from `EKCalendarItem`, you can perform the same methods on a reminder as you would on an event, such as adding an alarm with `addAlarm(:)` or setting a recurrence rule with `add RecurrenceRule(:)`.

> Important
> If your iOS app links on macOS and you need to access Reminders data, be sure to include the `NSRemindersUsageDescription` key in your Info.plist file.

You can create reminders using the `init(eventStore:)` class method. The `title` and `calendar` properties are required. The calendar for a reminder is the list with which it is grouped.

Like events, reminders can trigger time-based or location-based alarms to alert the user of a certain task. Read "Setting an Alarm" for more information on how to attach alarms to calendar items.

Save and Delete Reminders

To save a reminder to the Calendar database, call the `save(commit:)` method. To remove an event, call the `remove( :commit:)` method. The `title` and `calendars` properties must explicitly be set before you save the reminder.

> Note
> Just like when saving or removing events, make sure that if you pass NO to the `commit` parameter, you later invoke the `commit()` method to save your changes.

Edit Reminders

To associate a start date or due date with a reminder, use the `startDateComponents` and `dueDate Components` properties. To complete a reminder, set the `completed` property to YES, which automatically sets `completionDate` to the current date.

> Important
> If your app modifies a user's Calendar database, it must get confirmation from the user before doing so. An app should never modify the Calendar database without specific instruction from the user.

See Also

Events and reminders

⊟ Retrieving events and reminders Fetch events and reminders from the Calendar database.

≡ Updating with notifications Register for notifications about changes and keep your app up to date.

Managing Location-Based Reminders Add, fetch, complete, remove, and sort location-based reminders in your app.

class EKEvent

A class that represents an event in a calendar.

class EKReminder

A class that represents a reminder in a calendar.