EventKit / Creating a recurring event

Article

Creating a recurring event

Set up an event or reminder that repeats.

Overview

Recurring events repeat over a specified interval of time. To make an event recurring, assign it a recurrence rule that describes when the event occurs. Recurrence rules are represented by instances of the `EKRecurrenceRule` class.

Recurrence is applicable to both calendar events and reminders. Unlike with recurring events, only the first incomplete reminder of a recurring set is obtainable. This is true with EventKit as well as the Reminders app. When the reminder is completed, the next reminder in the recurrence set becomes available.

Create a Basic Rule

You can create a recurrence rule with a simple daily, weekly, monthly, or yearly pattern using the `init(recurrenceWith:interval:end:)` method. You provide three values to this method:

- **The recurrence frequency.** This is a value of type `EKRecurrenceFrequency` that indicates whether the recurrence rule is daily, weekly, monthly, or yearly.

- **The recurrence interval.** This is an integer greater than `0` that specifies how often a pattern repeats. For example, if the recurrence rule is a weekly recurrence rule and its interval is `1`, then the pattern repeats every week. If the recurrence rule is a monthly recurrence rule and its interval is `3`, then the pattern repeats every three months.

- **The recurrence end.** This optional parameter is an instance of the `EKRecurrenceEnd` class, which indicates when the recurrence rule ends. The recurrence end can be based on a specific end date or on an amount of occurrences. If you don’t want to specify an end for the recurrence rule, pass `nil`.

Create a Complex Rule

You can create a recurrence rule with a complex pattern using the `init(recurrenceWith:interval:daysOfTheWeek:daysOfTheMonth:monthsOfTheYear:weeksOfTheYear:daysOfTheYear:setPositions:end:)` method. As for a basic recurrence rule, you provide a frequency, interval, and optional end for the recurring event. In addition, you can provide a combination of optional values describing a custom rule, as listed in the table below.

Parameter name	Accepted values	Can be combined with	Example
`days`	An array of `EKRecurrenceDayOfWeek` objects.	All recurrence rules except for daily recurrence rules.	An array containing `EKTuesday` and `EKFriday` objects will create a recurrence that occurs every Tuesday and Friday.
`monthDays`	An array of nonzero `NSNumber` objects ranging from –31 to 31. Negative values indicate counting backward from the end of the month.	Monthly recurrence rules only.	An array containing the values 1 and –1 will create a recurrence that occurs on the first and last day of every month.
`months`	An array of `NSNumber` objects with values ranging from 1 to 12, corresponding to Gregorian calendar months.	Yearly recurrence rules only.	If your originating event occurs on January 10, you can provide an array containing the values 1 and 2 to create a recurrence that occurs every January 10 and February 10.
`weeksOfTheYear`	An array of nonzero `NSNumber` objects ranging from –53 to 53. Negative values indicate counting backward from the end of the year.	Yearly recurrence rules only.	If your originating event occurs on a Wednesday, you can provide an array containing the values 1 and –1 to create a recurrence that occurs on the Wednesday of the first and last weeks of every year. If a specified week does not contain a Wednesday in the current year, as can be the case for the first or last week of a year, the event does not occur.

Parameter name	Accepted values	Can be combined with	Example
daysOfTheYear	An array of nonzero NSNumber objects ranging from –366 to 366. Negative values indicate counting backward from the end of the year.	Yearly recurrence rules only.	You can provide an array containing the values 1 and –1 to create a recurrence that occurs on the first and last day of every year.
setPositions	An array of nonzero NSNumber objects ranging from –366 to 366. Negative values indicate counting backward from the end of the list of occurrences.	All recurrence rules except for daily recurrence rules.	If you provide an array containing the values 1 and –1 to a yearly recurrence rule that has specified Monday through Friday as its value for days of the week, the recurrence occurs only on the first and last weekday of every year.

You can provide values for any number of the parameters in the table. Parameters that don't apply to a particular recurrence rule are ignored. If you provide a value for more than one of the parameters, the recurrence occurs only on days that apply to all provided values.

Once you have created a recurrence rule, you can apply it to a calendar event or reminder with the `addRecurrenceRule(_:)` method of `EKCalendarItem`.

See Also

Recurrence

`class EKRecurrenceDayOfWeek`

A class that represents the day of the week.

`class EKRecurrenceEnd`

A class that defines the end of a recurrence rule.

`class EKRecurrenceRule`

A class that describes the pattern for a recurring event.