Documentation   Language: Swift

Contacts / CNContactStore

Class

CNContactStore

The object that fetches and saves contacts, groups, and containers from the user’s Contacts database.

iOS 9.0+ | iPadOS 9.0+ | Mac Catalyst 13.1+ | macOS 10.11+ | visionOS 1.0+ | watchOS 2.0+

```
class CNContactStore
```

Mentioned in

Accessing the contact store

Overview

The `CNContactStore` object represents the user’s contacts store database, and you use it to fetch information from that database and save changes back to it. There are a few recommended ways you can implement fetch and save requests in your app:

- Fetch only the properties that you need for contacts.
- When fetching all contacts and caching the results, first fetch all contacts identifiers, then fetch batches of detailed contacts by identifiers as required.
- To aggregate several contacts fetches, first collect a set of unique identifiers from the fetches. Then fetch batches of detailed contacts by those unique identifiers.
- If you cache the fetched contacts, groups, or containers, you need to refetch these objects (and release the old cached objects) when `CNContactStoreDidChange` is posted.

Because `CNContactStore` fetch methods perform I/O, it’s recommended that you avoid using the main thread to execute fetches.

Topics

Requesting access to the user’s contacts

```
func requestAccess(for: CNEntityType, completionHandler: (Bool, (any Error)?) -> Void)
```

Requests access to the user’s contacts.

```
class func authorizationStatus(for: CNEntityType) -> CNAuthorizationStatus
```

Returns the current authorization status to access the contact data.

```
enum CNAuthorizationStatus
```

An authorization status the user can grant for an app to access the specified entity type.

```
enum CNEntityType
```

The entities the user can grant access to.

Fetching contacts

```
func enumerateContacts(with: CNContactFetchRequest, usingBlock: (CNContact, UnsafeMutablePointer<ObjCBool>) -> Void) throws
```

Returns a Boolean value that indicates whether the enumeration of all contacts matching a contact fetch request executes successfully.

```
func unifiedMeContactWithKeys(toFetch: [any CNKeyDescriptor]) throws -> CNContact
```

Fetches the unified contact that’s the *me* card.

```
func unifiedContact(withIdentifier: String, keysToFetch: [any CNKeyDescriptor]) throws -> CNContact
```

Fetches a unified contact for the specified contact identifier.

```
func unifiedContacts(matching: NSPredicate, keysToFetch: [any CNKeyDescriptor]) throws -> [CNContact]
```

Fetches all unified contacts matching the specified predicate.

Fetching groups and containers

```
func defaultContainerIdentifier() -> String
```

Returns the identifier of the default container.

```
func groups(matching: NSPredicate?) throws -> [CNGroup]
```

Fetches all groups matching the specified predicate.

```
func containers(matching: NSPredicate?) throws -> [CNContainer]
```

Fetches all containers matching the specified predicate.

Fetching change history info

```
var currentHistoryToken: Data?
```

The current history token.

Saving changes

```
func execute(CNSaveRequest) throws
```

Executes a save request and returns success or failure.

Responding to contact store changes

```
static let CNContactStoreDidChange: NSNotification.Name
```

Posted when changes occur to the contact store.

Relationships

Inherits From

`NSObject`

Conforms To

`CVarArg`
`CustomDebugStringConvertible`
`CustomStringConvertible`
`Equatable`
`Hashable`

`NSObjectProtocol`

See Also

Essentials

Accessing the contact store

Request permission from the person to read and write their contact data.

{} Accessing a person’s contact data using Contacts and ContactsUI

Allow people to grant your app access to contact data by adding the Contact access button and Contact access picker to your app.

`NSContactsUsageDescription`

A message that tells people why the app is requesting access to their contacts.

`com.apple.developer.contacts.notes`

A Boolean value that indicates whether the app may access the notes in contact entries.