SwiftData / Query

Structure

Query

A type that fetches models using the specified criteria, and manages those models so they remain in sync with the underlying data.

SwiftData | SwiftUI | iOS 17.0+ | iPadOS 17.0+ | Mac Catalyst 17.0+ | macOS 14.0+ | tvOS 17.0+ | visionOS 1.0+ | watchOS 10.0+

```
@MainActor @preconcurrency
struct Query<Element, Result> where Element : PersistentModel
```

Mentioned in

Preserving your app’s model data across launches

Topics

Creating a query

`init(FetchDescriptor<Element>, animation: Animation)`

Create a query with a SwiftData fetch descriptor.

`init(filter: Predicate<Element>?, sort: [SortDescriptor<Element>], animation: Animation)`

Create a query with a predicate, and a list of sort descriptors.

`init<Value>(filter: Predicate<Element>?, sort: KeyPath<Element, Value?>, order: SortOrder, animation: Animation)`

Creates a query with a predicate, a key path to a property for sorting, and the order to sort by.

`init<Value>(filter: Predicate<Element>?, sort: KeyPath<Element, Value>, order: SortOrder, animation: Animation)`

Creates a query with a predicate, a key path to a property for sorting, and the order to sort by.

`init(FetchDescriptor<Element>, transaction: Transaction?)`

Create a query with a SwiftData fetch descriptor.

`init(filter: Predicate<Element>?, sort: [SortDescriptor<Element>], transaction: Transaction?)`

Create a query with a predicate, and a list of sort descriptors.

`init<Value>(filter: Predicate<Element>?, sort: KeyPath<Element, Value>, order: SortOrder, transaction: Transaction?)`

Create a query with a predicate, a key path to a property for sorting, and the order to sort by.

`init<Value>(filter: Predicate<Element>?, sort: KeyPath<Element, Value?>, order: SortOrder, transaction: Transaction?)`

Create a query with a predicate, a key path to a property for sorting, and the order to sort by.

Getting query configuration

`var modelContext: ModelContext`

Current model context `Query` interacts with.

`var fetchError: (any Error)?`

An error encountered during the most recent attempt to fetch data.

Accessing the value

`var wrappedValue: Result`

The most recent fetched result from the Query.

Relationships

Conforms To

`DynamicProperty`, `Sendable`, `SendableMetatype`

See Also

Model fetch

{} Filtering and sorting persistent data

Manage data store presentation using predicates and dynamic queries.

`macro Query()`

Fetches all instances of the attached model type.

☰ Additional query macros

Supplementary macros that enable you to narrow query results and tell SwiftData how to sort and order those results.

`struct FetchDescriptor`

A type that describes the criteria, sort order, and any additional configuration to use when performing a fetch.