

[Assignables](#) / `UserIdentity`

Protocol

UserIdentity

Types conforming to this protocol can act as user identities for editors of a document.

iOS 17.4+ | iPadOS 17.4+ | Mac Catalyst 17.4+ | visionOS

```
protocol UserIdentity : Decodable, Encodable, Hashable, Sendable
```

Topics

Inspecting an identity

```
var stringRepresentation: String
```

String representation of this user identity for display or debugging purposes.

Required

```
var typeID: String
```

A unique type identifier for this user identity.

Required

Setting the scope

```
func scope<R>(() throws -> R) rethrows -> R
```

Sets the user identity for document-related operations that occur within the closure passed in.

```
func scope<R>(() async throws -> R) async rethrows -> R
```

Sets the user identity for document-related operations that occur within the async closure passed in.

Getting a type eraser

```
func eraseToAnyUserIdentity() -> AnyUserIdentity
```

Wraps this user identity with a type eraser.

Required Default implementation provided.

```
typealias As
```

An alias for [UserIdentityFactory](#) for convenience.

Relationships

Inherits From

Decodable

Encodable

Equatable

Hashable

Sendable

SendableMetatype

Conforming Types

AnonymousUserIdentity, AnyUserIdentity, StringUserIdentity

See Also

Identity

```
struct AnonymousUserIdentity
```

A user identity for unknown editors.

```
struct AnyUserIdentity
```

A user identity that performs type erasure by wrapping another user identity.

`struct StringUserIdentity`

A user identity defined by a string.

`class UserIdentityTypeRegistry`

A registry for user identity types. Assignable documents and document elements store user identity data as `Data` objects. In order for that data to be deserialized, the type to deserialize it as needs to be known to `UserIdentityTypeRegistry`. Without registration of the user identity, custom types won't be deserializable.

`enum UserIdentityFactory`

A type that contains helpers for creating user identity objects.