

[Assignables](#) / AnyUserIdentity

Structure

AnyUserIdentity

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

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

```
struct AnyUserIdentity
```

Topics

Creating a user identity

```
init<T>(T)
```

Initializes this type eraser with a user identity to wrap.

```
init(from: any Decoder) throws
```

Creates a new instance by decoding from the given decoder.

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.

Inspecting an identity

```
var stringRepresentation: String
```

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

```
var typeId: String
```

A unique type identifier for this user identity.

```
enum Error
```

Error type for this user identity.

Instance Methods

```
func encode(to: any Encoder) throws
```

Encodes this value into the given encoder.

```
func hash(into: inout Hasher)
```

Hashes the essential components of this value by feeding them into the given hasher.

Comparing identities

```
static func == (AnyUserIdentity, AnyUserIdentity) -> Bool
```

Returns a Boolean value indicating whether two values are equal.

Relationships

Conforms To

Decodable

Encodable

Equatable

Hashable

Sendable

SendableMetatype

UserIdentity

See Also

Identity

`protocol` `UserIdentity`

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

`struct` `AnonymousUserIdentity`

A user identity for unknown editors.

`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.