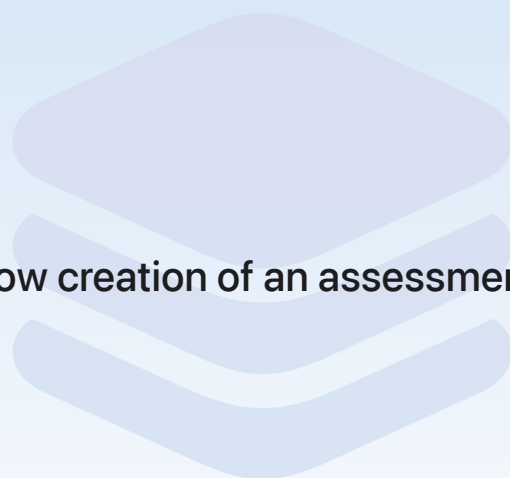


Framework

Assignables

A framework that contains wrappers for a PDF to allow creation of an assessment and student work on that assessment.

iOS 15.4+ | iPadOS 15.4+ | Mac Catalyst 15.4+



Overview

An AssignableDocument is a document that contains a base PDF and stores added markup, annotations, questions, and scoring options for that PDF.

This framework focuses on the student and teacher experience. Teachers can assign the document to a student by creating an AssignedWorkDocument from the AssignableDocument. In addition to those data types, Assignables also provides SwiftUI view classes to enable editing of those documents in your own views.

Once a student completes their work, they can return it to the teacher to be scored and to receive comments on their work.

Authorship is important at each of these steps, so Assignables supports attribution of changes to any document using types that conform to UserIdentity.

Since teachers may collaborate on making an assignable or scoring student works, or students might collaborate on a document, both document types support merging copies of the document with any other copy of the same document.

Topics

Assignable document

```
struct AssignableDocument
```

An assignable document is an augmented PDF that allows teachers to mark up the PDF with the intention of students taking the assessment.

`struct AssignedWorkDocument`

An assigned work document is a document that contains taker and scorer markup specific to a taker. It also contains a copy of the assignable document upon which it is based.

`protocol Assignable`

Documents conforming to this protocol can be assigned to a user.

Configuration

`protocol AssignableDocumentConfiguration`

A type that specifies the options for an assignable document.

`protocol AssignedWorkDocumentConfiguration`

A type that specifies the score of a document.

Presentation

`struct AssignableView`

SwiftUI View to display an AssignableView.

`struct AssignedWorkDocumentView`

SwiftUI View to display an AssignedWorkDocument

Document elements

`protocol DocumentElement`

Represents an element that is contained within a document. Such elements can have identifiers that uniquely identify them within a document.

`struct BasicDocumentElementID`

A default implementation for a document element identifier.

`protocol DocumentElementID`

An identifier for an element in a document.

`protocol AssignableDocumentElement`

An element of an AssignableDocument.

`protocol AssignedWorkDocumentElement`

An element of an AssignedWorkDocument.

Mergeable document

`protocol MergeableDocument`

Documents conforming to this protocol can merge several copies of the document into a single document.

`struct MergeablePartsContainerPartID`

The ID of a part in a `MergeablePartsContainer`.

`protocol MergeableDocumentPage`

Types conforming to this protocol indicate that they are a page in a MergeableDocument conforming object.

`protocol MergeablePartsContainer`

Objects conforming to this protocol allow merging in other replicas of themselves or merging in individual parts of themselves.

`struct DocumentThumbnail`

A structure that contains an image of an entire page or a portion of a page and the ID of the page the image is from.

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 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.