

TeachKit Zwischenpräsentation

Session 602

Patrick Niepel Teachify Engineer
Marcel Hagmann Teachify Engineer
Philipp Knoblauch Teachify Engineer

TeachKit

Ziel

Lehrer

- Klassen/Fächer/Aufgabenblätter/Aufgaben anlegen

- Schüler nach Gruppen einteilen

- Schüler zu einem Fach hinzufügen/löschen

- Statistiken seiner Schüler (% richtig/falsch und welcher Schüler)

Schüler

- Fach teilnehmen/austreten

- Push Nachricht, bei neuem Aufgabenblatt

- Bearbeiten der Aufgaben in Form von Spielen

Lehrer/Schüler

- Zugriff auf Leaderboard

Realisierung über CloudKit

TeachKit

Ziel

Lehrer

Klassen/Fächer/Aufgabenblätter/Aufgaben anlegen

Schüler nach Gruppen einteilen

Schüler zu einem Fach hinzufügen/löschen

Statistiken seiner Schüler (% richtig/falsch und welcher Schüler)

Schüler

Fach teilnehmen/austreten

Push Nachricht, bei neuem Aufgabenblatt

Bearbeiten der Aufgaben in Form von Spielen

Lehrer/Schüler

Zugriff auf Leaderboard

Realisierung über CloudKit

TeachKit

Ziel

Lehrer

Klassen/Fächer/Aufgabenblätter/Aufgaben anlegen

Schüler nach Gruppen einteilen

Schüler zu einem Fach hinzufügen/löschen

Statistiken seiner Schüler (% richtig/falsch und welcher Schüler)

Schüler

Fach teilnehmen/austreten

Push Nachricht, bei neuem Aufgabenblatt

Bearbeiten der Aufgaben in Form von Spielen

Lehrer/Schüler

Zugriff auf Leaderboard

Realisierung über CloudKit

 Umgesetzt

 Probleme

TeachKit

Ziel

Lehrer

Klassen/Fächer/Aufgabenblätter/Aufgaben anlegen

Schüler nach Gruppen einteilen

Schüler zu einem Fach hinzufügen/löschen

Statistiken seiner Schüler (% richtig/falsch und welcher Schüler)

Schüler

Fach teilnehmen/austreten

Push Nachricht, bei neuem Aufgabenblatt

Bearbeiten der Aufgaben in Form von Spielen

Lehrer/Schüler

Zugriff auf Leaderboard

Realisierung über CloudKit



Umgesetzt



Probleme



Diskussion

TeachKit

Ziel

Lehrer

Klassen/Fächer/Aufgabenblätter/Aufgaben anlegen

Schüler nach Gruppen einteilen

Schüler zu einem Fach hinzufügen/löschen

Statistiken seiner Schüler (% richtig/falsch und welcher Schüler)

Schüler

Fach teilnehmen/austreten

Push Nachricht, bei neuem Aufgabenblatt

Bearbeiten der Aufgaben in Form von Spielen

Lehrer/Schüler

Zugriff auf Leaderboard

Realisierung über CloudKit



Umgesetzt



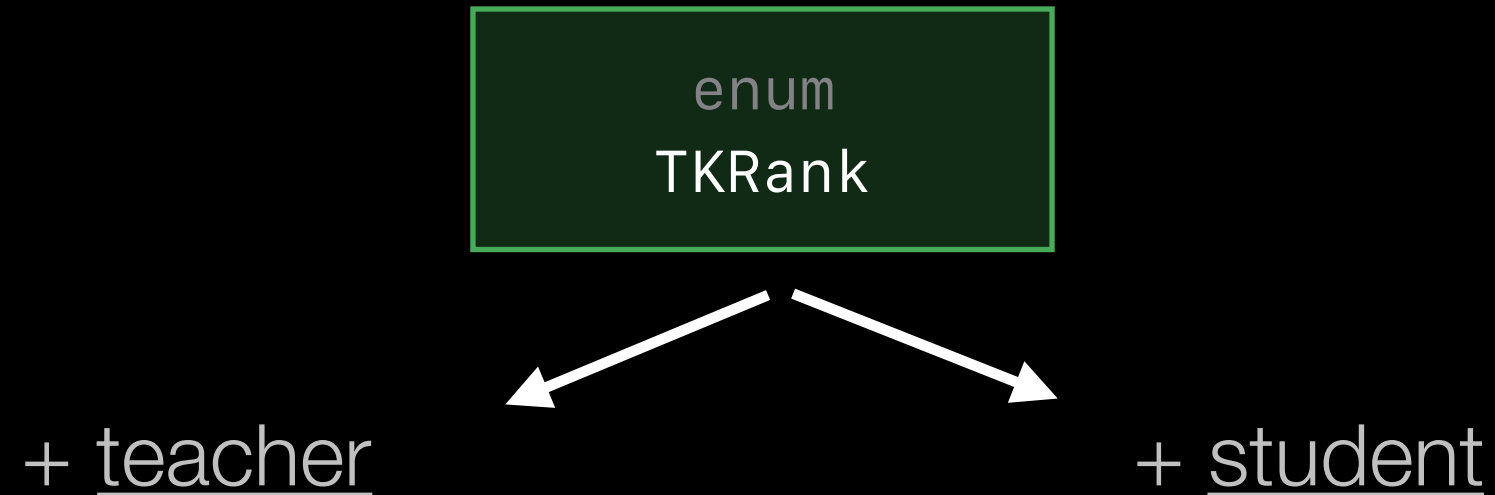
Probleme



Diskussion

TeachKit

Initialisierung - Student und Lehrer



- private Database
- kann ab TKClass zugreifen

- shared Database
- kann ab TKSubject zugreifen

TeachKit

Initialisierung - Student und Lehrer

- TKClassController, TKSubjectController, TKDocumentController, TKExerciseController, TKSolutionController müssen wie folgt initialisiert werden.
- Grund: .student muss die (Shared) CKRecordZone gefetched werden

```
let classCtrl = TKClassController()
classCtrl.initialize(withRank: rank) { succeed in
    print("Class init -> \(succeed)")
}
```


TeachKit

Fetch

- wenn tkClass == nil —> fetched alle Subjects
- fetchSortOptions —> die Reihenfolge im Array wird beachtet
- completion:
 - Option 1: Array mit Subjects aus der Cloud & error = nil
 - Option 2: leeres Array & TKError

```
func fetchSubject(forClass tkClass: TKClass? = nil,  
                 withFetchSortOptions fetchSortOptions: [TKFetchSortOption] = [],  
                 completion: @escaping ([TKSubject], TKError?) -> ())
```

TeachKit


Update


- subject —> das Fach das aktualisiert werden soll
- completion:
 - Option 1: Aktualisiertes Subject & error = nil
 - Option 2: subject = nil & TKError

```
func update(subject: TKSubject,  
            completion: @escaping (TKSubject?, TKError?) -> ())
```

TeachKit

Update

-  das aktualisierte Object muss durch das im Model ausgetauscht werden!
- sonst kann es später nicht erneut geupdated werden.

```
func update(subject: TKSubject,  
            completion: @escaping (TKSubject?, TKError?) -> ())  
{ updatedSubject, error in  
  
    if let updatedSubject = updatedSubject {  
        self.subject = updatedSubject //   
    }  
  
}
```

TeachKit

Delete

- subject —> das Fach das gelöscht werden soll

```
func delete(subject: TKSubject,  
            completion: @escaping (TKSubject?, TKError?) -> ())
```

TeachKit

Add

- subject —> das Fach das in die Cloud geladen werden soll
- tkClass —> das TKClass-Parent-Objekt

```
func add(subject: TKSubject,  
        toTKClass tkClass TKClass,  
        completion: @escaping (TKSubject?, TKError?) -> ())
```

-  gleiche Bedienung wie bei Update!

TeachKit

Create

- tkClass —> das Fach das in die Cloud geladen werden soll

```
func create(tkClass: TKClass,  
            completion: @escaping (TKSubject?, TKError?) -> ())
```

-  gleiche Bedienung wie bei Update!

TeachKit

Sharing

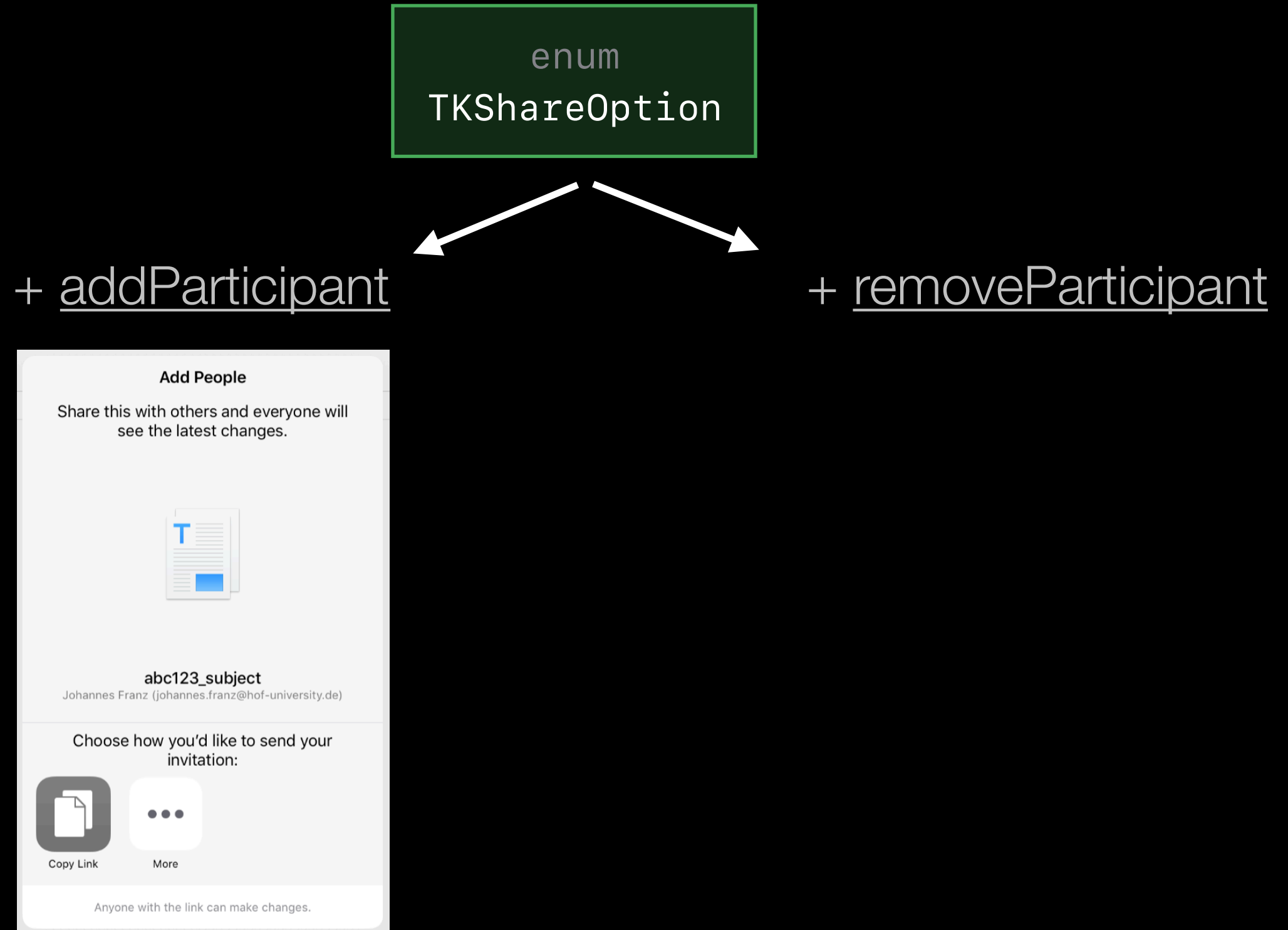
- es kann nur TKSubject geteilt werden, alles darunter wird automatisch geteilt

```
func createCloudSharingController(forSubject subject: TKSubject,  
    withShareOption shareOption: TKShareOption,  
    completion: @escaping (UIViewController?, TKError?) -> ())
```

- UIViewController —> ViewController der nur noch presented werden muss

TeachKit

Sharing



TeachKit

Probleme - Subscriptions

Class

CKQuerySubscription

The abstract base class for CloudKit subscriptions.

Overview

Note

CKQuerySubscription is not supported in a [sharedCloudDatabase](#).

TeachKit

Diskussion

Schüler nach Gruppen einteilen

Fragen?

Update/Add/Create

inout?

More Information

Documentation

TeachKit Resources

developer.teachify.com/teachkit

Technical Support

Teachify Developer Forums

devforums.teachify.com

Developer Technical Support

developer.teachify.com/support/technical

General Inquiries

teachkit@teachify.com

