

Stundenplan–App für iOS

Dokumentation, Spezifikation, Konstruktion

Daniel Glaser, Stefan Scharrer

06.07.2017

Inhaltsverzeichnis

1	Calendar Interface	1
1.1	Einleitung	1
1.2	CalendarController	2
1.2.1	createCalendar	2
1.2.2	removeCalendar	2
1.2.3	createAllEvents	2
1.2.4	updateAllEvents	2
1.2.5	removeAllEvents	3
1.2.6	CalendarRoutine	3
1.3	CalendarInterface	4
1.3.1	createCalenderIfNeeded	4
1.3.2	removeCalendar	4
1.3.3	createEvent	4
1.3.4	updateEvent	4
1.3.5	removeEvent	4
1.3.6	saveIDs	4
1.4	CalendarData	5

1 Calendar Interface

1.1 Einleitung

Im folgenden Diagramm ist die Architektur und Integration des Calendar Interface in die Stundenplan App dargestellt.

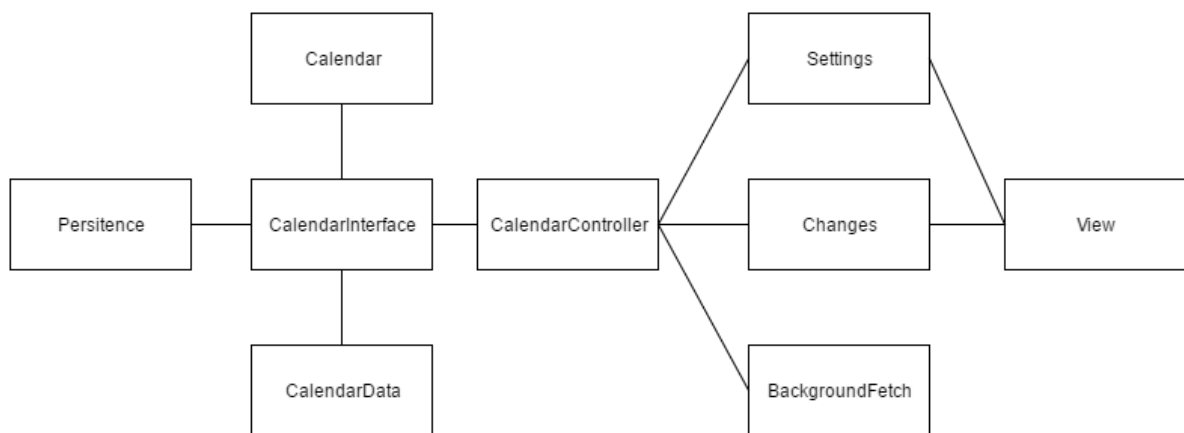


Abbildung 1.1: CalendarInterface Diagramm

Das Calendar Interface besteht aus folgenden drei Klassen:

- CalendarController
- CalendarInterface
- CalendarData

Im Folgenden wird auf die Klassen genauer eingegangen.

1.2 CalendarController

Der CalendarController ist der Controller für das Calendar Interface. Über ihm wird auf das Interface zugegriffen.

Im Folgenden wird auf die wichtigsten Methoden eingegangen:

- createCalendar()
- removeCalendar()
- createAllEvents()
- updateAllEvents()
- removeAllEvents()
- CalendarRoutine()

Jede Methode überprüft ob die Berechtigung gewährt wurde und reagiert entsprechend.

1.2.1 createCalendar

Erzeugt den Kalender falls die Berechtigung vorhanden ist. Falls er erzeugt wurde werden die Events mit createAllEvents in den Kalender geschrieben. Als Rückgabewert gibt er den Berechtigungsstatus zurück. Es gibt drei Rückgabewerte:

- authorized
Bedeutet das der Nutzer die Berechtigung erteilt hat und der Kalender angelegt wurde oder bereits angelegt war.
- notDetermined
Bedeutet das die Berechtigung gerade abgefragt wird.
- denied
Bedeutet das die Berechtigung verweigert wurden.

1.2.2 removeCalendar

Löscht den Kalender mit Hilfe des CalendarInterface.

1.2.3 createAllEvents

Erzeugt die aus den Vorlesungen die Events, die anschließend mit Hilfe des CalendarInterface in den Kalender geschrieben werden. Falls der Kalender noch nicht erzeugt wurde wird er angelegt. Die EventID's werden persistent gespeichert.

1.2.4 updateAllEvents

Ermittelt welche Art von Änderung vorliegt und passt die Events mit Hilfe des CalendarInterface entsprechend an. Die EventID's werden persistent gespeichert.

1.2.5 removeAllEvents

Holt sich für die übergebenden Vorlesungen die EventID's und löscht die entsprechenden Events mit Hilfe des CalendarInterface. Die Änderungen an den EventID's werden persistent gespeichert.

1.2.6 CalendarRoutine

Aktualisiert die Vorlesungen im Kalender. Dazu holt sie sich die abgewählten und neu ausgewählten Vorlesungen und übergibt sie der removeAllEvents oder createAllEvents-Methode.

1.3 CalendarInterface

Dies ist die Klasse die direkt auf den Kalender zugreift.

Im Folgenden wird auf die wichtigsten Methoden eingegangen:

- `createCalendarIfNeeded()`
- `removeCalendar()`
- `createEvent()`
- `updateEvent()`
- `removeEvent()`
- `saveIDs()`

1.3.1 createCalendarIfNeeded

Erzeugt den Kalender falls er nicht bereits vorhanden ist. Gibt einen Boolean-Wert zurück ob der Kalender angelegt wurde.

1.3.2 removeCalendar

Löscht den Kalender falls dieser vorhanden ist. Gibt einen Boolean-Wert zurück ob das Löschen erfolgreich war.

1.3.3 createEvent

Schreibt das übergebende Event in den Kalender.

1.3.4 updateEvent

Das zu der EventID dazugehörige Event wird mit den entsprechend Werten des übergebenen Event angepasst.

1.3.5 removeEvent

Löscht das übergebene Event. Gibt einen Boolean-Wert zurück ob das Löschen erfolgreich war.

1.3.6 saveIDs

Speichert die EventIDs der Vorlesungen und Änderungen.

1.4 CalendarData

Die CalendarData-Klasse dient dazu, die EventIDs der Vorlesungen und Änderungen getrennt in Dictionarys zu speichern. Die Dictionarys bestehen aus einer Zuordnung von einer ID einer Vorlesung zu mehreren EventIDs.