

## Anhang 1: Spezifikation

Alle genannten Funktionen gelten nur für Termine, welche in OpenOLAT mit dem internen Editor erstellt wurden. Importierte Kalender können in OpenOLAT nicht editiert werden.

Termine können auch über das REST API von einem externen Programm in OpenOLAT angelegt werden. Diese Termine gelten ebenfalls als „interne Termine“. Wahlweise kann konfiguriert werden, ob die so erstellten Termine durch den Benutzer verändert werden dürfen oder nicht.

### 1: Löschen einzelner Termine eines Serientermins

Die folgenden Arbeiten werden vorgenommen:

- Implementation Benutzerschnittstelle, um einzelne Termine eines Serientermins zu löschen
- Implementation Benutzerschnittstelle, um einzelne Termine und alle folgende Termine eines Serientermins zu löschen
- Anpassung Speicherroutine, um gelöschte Einzeltermine beim Verschieben eines Serientermins korrekt zu verschieben
- Erweitern der Kalender Darstellungsroutine für Unterstützung von mehreren gelöschten Terminen eines Serientermins

### Option 2: Verändern einzelner Termine eines Serientermins

Neben der einfachen Ausnahmeregel von Serienterminen ist es gemäss ical Standard auch möglich, einzelne Termine eines Serientermins zu verändern. Eine Veränderung wäre z.B. dass für einen Einzeltermin der Serie der Durchführungsort, die Uhrzeit, das Datum, die Beschreibung oder der Titel verändert würde.

Anpassungen einzelner Termine sind technisch völlig anders gespeichert als eine einfache „Ausnahmeregel“. Während die Ausnahmen Eigenschaften des Serientermins sind, werden für die Ausnahmen neue Termine angelegt, die jedoch die selbe UID wie der Serientermin haben und zusätzlich eine sogenannte Recurrence-ID besitzen.

Die Problematik ist nun, dass OpenOLAT in der aktuellen Version davon ausgeht, dass die UID der Termine in jedem Fall eindeutig ist. Die Erweiterung der Spezifikation mit Unterstützung der Recurrence-ID ist nicht implementiert. Die UID des Termins ist zentral in der Handhabung des gesamten Kalenders, von der Benutzerschnittstelle bis zur Speicherung. Um Recurrence-ID unterstützen zu können ist daher ein umfassenderes Refactoring im Kalendercode durchzuführen.

Die folgenden Arbeiten werden vorgenommen:

- Implementation Recurrence-ID in Event und Kalender Datenmodell
- Erweiterung Manager Klassen für korrekte Behandlung von Events mit identischer UID und unterschiedlicher Recurrence-ID
- Anpassung Benutzerschnittstelle bei drag&drop Aktionen einzelner Termine eines Serientermins, Abspeichern als neue Events mit eigener Recurrence-ID
- Anpassung Benutzerschnittstelle um Events mit identischer UID und unterschiedlicher Recurrence-ID korrekt darzustellen
- Anpassung Benutzerschnittstelle um im Termin Editorfenster einzelne Termine eines Serientermins individuell ändern zu können, Abspeichern als neue Events mit eigener Recurrence-ID
- Anpassung Speicherroutine, um veränderte Einzeltermine bei späterem Löschvorgang korrekt in Ausnahmen auf dem Serientermin umzuwandeln
- Anpassung Speicherroutine, um die Recurrence-ID von veränderte Einzeltermine beim Verschieben eines Serientermins korrekt anzupassen