
Vorname	Nachname	Beruf	Dauer
Marten	Meißner	Fachinformatiker AE	07.22-07.24

Wochenbericht KW 45. (2023.11.06. - 12.)

Vor dem Umbau Projekt, haben wir jedem Termin Viemodell ein eigenes „AppointmentTime“ Objekt gegeben um dort Typensicherheit, exklusive Methoden zu nutzen.

Dadurch das wir jetzt ein Objekt für den Bauplan einen wiederkehrenden Termin haben, haben wir mehr Optionen, bzw. ist es einfacher hier mit einer Methode einen oder mehrere Termine über einen gewissen Zeitraum abzufragen.

Mit der bisherigen Lösung konnten wir jedes Datenmodell mit nur einem bestimmten Tag abfragen.

Dies lief so ab, dass ein initialer Tag bekamen und eine Regelung wie sich ein Tag wiederholte.

Da jedes Viemodell unserer Termine immer nur eine Datei enthielt, haben wir sehr viele Objekte erzeugt für einen gewissen Tag.

Dies hatte verschiedene Nachteile, zu einem wurden für jeden weiteren Termin ein ganzes Objekt erschaffen, mit 15 identischen Eigenschaften und nur einer neu angepassten Eigenschaft.

Neben den vielen Eigenschaften, die übernommen wurden, musste zudem mehr Speicher verschwendet.

Das führte dazu, dass es schwer wurde einzelne Termine zu filtern, da jeder neue Termin die gleiche ID hatte, die eigentlich einzigartig sein sollte.

Da wir für jedes Mal, wenn ein Termin eine Wiederholung hatte, ein neues VM gebaut wurde, war es schwierig einen größeren Zeitpunkt abzufragen.

Wenn wir einen ganzen Monat abfragten und im Ausschlussverfahren alle Tage ausfilterten, an dem dieser Termin nicht stattfand, um dann am übrig gebliebenen Tag den richtigen Tag zu finden.

Diese Verfahren waren sehr ineffizient.

Um einen Termin anhand einer Wiederholungsregel zu bauen, wurde für jedes Datenmodell der ganze Monat abgefragt, ob er stattfinden sollte, dies führt dazu, dass wir für jedes Datenmodell 30 Tage ablehnten und dies für 30 Tage wiederholten.

Diese Vorgehensweise hat eine Big O Notation von $O(m^n)$ für jedes Element in der einen List müssen so oft abfragen, wie wir Tage im Monat haben, die wir ablehnen.

Nun erstellen wir aber nur noch ein Viewmodell für jedes Datenmodell, somit sind alle Viewmodelle wieder einzigartig, können leichter gefunden und zugewiesen werden.

In der Bloc Datei brauche ich jetzt nur noch eine Liste von DMs und ein Datum oder ein Monat welche ich abfragen möchte.

Nach dem Refactoring habe ich jetzt ein Objekt, das alle Termine von sich selber kennt und einzigartig und keine redundanten Informationen zusätzlich speichern muss.

Jetzt dienen die Termine VMs nur noch zum Darstellen der Informationen.

Kontrolliert am: _____ Unterschrift : _____