

---

Vorname	Nachname	Beruf	Dauer
Marten	Meißner	Fachinformatiker AE	07.22-07.24

---

## Wochenbericht KW 42. (2023.10.16. - 22.)

---

Einer der Fehler, die ich dem letzten großen merge unserer App feststelle, war das wir immer noch nicht die Korrekte DateTime Objekte erstellt haben.

Wir bekommen über die eingelesene URL von Microsoft eine ICalendar(.ics) Datei, diese hat verschiedene Zeit und Datumsangaben.

Wir bekommen einmal einen String, der ungefähr so aussieht, 'TZID=W. Europe Standard Time 20230403T114500' wir bekommen einen Key z. B. West. Europe. Standart, nach etwas Recherche stellte ich fest, dass eigentlich die CEST(Central Europe Summer Time) gemeint war, gefolgt von einem Datum, Uhrzeit String.

Dieser String bestimmt den ersten Zeitpunkt diesen Termin z. B. „20230403T1145“, bedeutet 3.4.2023 um 11:45, diese Zeit enthält alle Informationen zu einem DateTime Objekt.

Das Datum und Uhrzeit mit einer Zeitzonen anpassung, in unserem Beispiel wären es zwei stunden aufaddiert, da es die Zentral europäische Sommerzeit handelt. Dann stellte ich fest das, wenn ich eine DateTime Objekt in Dart versuche zu formatieren z.B. „20230403T1145“, erhalte ich daraus eine Coordinated Universal Time(UTC), dieser Wert ist aber nicht korrekt, da beim übersetzten dieses Strings aus einer Zeit ohne Zeitzonen Anpassung oder einer Markierung automatisch diesen String als UTC Zeit Objekt übersetzt.

Dadurch erhalte ich immer eine falsche Zeit, dies bereinigte ich, indem ich im Internet mir eine Map(Assoziatives Array) suchte, in der die Zeitzonen Namen mit den jeweils zugehörigen Zeitabweichungen zur UTC Zeit als Wert hinterlegt wurde.

Mithilfe dieser Liste subtrahierte ich die Zeitverschiebung vom Timestring, um eine bereinigte Zeit Objekt zu erhalten.

Neben dieser Eingangsinformation erhielten wir auch ein anderes Format, was zu Beginn einfacher aussah, aber auch wieder ein anderes Problem ergab.

In diesem Fall erhielt ich einen String, der ungefähr so aussieht „Value:2023-10-03“.

Da ein Datetime Objekt immer eine Uhrzeit enthält, wird hier beim Formatieren zu jedem Datum automatisch die früheste Zeit an diesem Datum hinzugefügt. Somit wird aus „2023-10-03“ ein DateTime Objekt mit folgender Information: DateTime(2023-10-03-00-00-00.000).

Wenn ich ein DateTime Objekt in meiner App nutzen möchte, muss ich vorher wissen, dass alle Zeiten nicht meiner Lokalen Zeitzone entsprechen, dadurch muss ich immer, wenn ich ein Datum oder eine Zeit ausgeben möchte, ihr die Lokale Zeitzone Verschiebung hinzufügen muss.

Wenn ich aber immer eine Lokale Zeitzone Verschiebung hinzufüge, erhalte ich diese auch bei ganztägigen Terminen wie einen Feiertag, dadurch fängt dieser Feiertag in meiner Lokale Zeitzone eine bis zwei Stunden später an.

Dieses Problem löste ich jetzt erstmal so, dass ich, wenn ein solches Objekt ausgegeben wird, vorher abzufragen, ob es zu diesem Termin eine konkrete Zeit gibt oder ein Ganztagestermin handelt, in diesem Fall gebe ich einen andern String zurück wie einen leeren String.

Nach mehreren Versuchen stellte ich fest, dass dies die beste Methode ist, die Zeiten korrekt anzeigen zu lassen.

Damit andere Programmierer nicht den gleichen Denkfehler machten, fing ich an diese Eigenschaft in den Dokumentationen der einzelnen Klassen mit aufzunehmen, sodass andere Entwickler dies direkt nachlesen können.

Kontrolliert am: \_\_\_\_\_ Unterschrift : \_\_\_\_\_