

**23.04. - 29.04.**

Versucht Server Lokal zu starten Virtuelle Maschinen eingerichtet

**30.04. - 06.05.**

Versucht Lokal Services zum laufen zu kriegen

- mit bash Scripte geschrieben
- Fehler versucht zu beheben
- Eigene Telegram Bot eingerichtet

**07.05. - 13.05.**

Eingelesen in Services um zu verstehen:

- Wie sind sie aufgebaut?
- Wie funktioniert der Datentransfer?
- Welche Strukturen können verändert werden um es zum laufen zu bekommen?

Passwort der VM umgangen, mit Kais Anleitung VM erfolgreich gestartet

**14.05. - 20.05.**

Python Kenntnisse aufgefrischt Kais Code versucht zu verstehen, leider ohne erfolg Neue Projekt/Microservice Ideen ausgedacht bzw. erweitert

**20.05. - 27.05.**

Wiki erweitert

**02.06.**

- 14 - 15:15 Uhr → Besprechung mit der Gruppe über Zwischenbericht

**03.06.**

- 17 - 19:30 Uhr → Zwischenbereich erweitert

**04.06.**

- 10 - 12:30 Uhr → Zwischenbericht erweitert
- 14 - 15 Uhr → letzte Besprechung über Zwischenbericht vor Abgabe / Präsentation

**09.06.**

- 14 - 18 Uhr → Cache Anforderungen / Technologien / Kriterien / Usecases & Userstories bearbeitet

**15.06.**

- 12 - 17 Uhr → Cache Doku durchgelesen / Alle Server mit testbot gestartet / Kriterien an registry und Weather service verfasst wie:

- Weather Format Umschreiben: Tagesformat muss kürzer verfasst werden → Grand ohne komma, Windgeschw. von 0.27m/s bis 1m/s, Luftfeuchtigkeit auch nur von bis, Nicht alle 3 sondern 6 stunden und wenn ja nur dann wenn beuth auch auf hat, nicht tag jedesmal angeben und uhrzeit ohne minuten und sekunden, versuchen emojis zu benutzen für wolken und so
- Weather cache objekt erweitern um value: ttl und cachename, cache wird nach name gespeichert und bekommt instruktionen von service nicht selber erstellen

## **16.06.**

- 10 - 15 Uhr → Einlesen in dden derzeitig existierenden Weather Service + einlesen in die OpenWeatherAPI

## **18.06.**

- 13 - 15 Uhr → Verschiedene Testanfragen an die OpenWeatherApi gestellt. Zusammenhänge von Daten wie Forecast, Weather, Onecall, History finden um optimale Requests zu erstellen, damit Datentransfer optimaler läuft

## **19.06.**

- 11 - 16 Uhr → Emojis können nicht mit klartext an TelegramBot geschickt werden, muss aktuell herausfinden, wie ich sie schicken kann, damit sie angezeigt werden. 7 Tages Anfrage an OpenWeatherAPI  
<https://api.openweathermap.org/data/2.5/onecall?lat=52.544500&lon=13.353260&exclude=hourly,current,minutely&units=metric&lang=de> um nur daily response zu bekommen, sowie andere sind übner onecall besser geeignet. Leider geht q=Berlin,de nicht. Api will bei onecall lon und lat, warum auch immer.

## **22.06.**

- 10 - 15 Uhr → Emojis gehen nur als icon direkt in der Nachricht / Minuten werden jetzt auch mitgeschickt. Date wurde umgebaut / Cache wurde angepasst / Vorbereitung für mehr implementierungen wie z.B. Methode fürs abfragen von Stündlicher Zeit / eventueller Zeitraum, als anfrage( Rasa müsste angepasst werden )

## **23.06. - Vortrag**

- Nachrichten wurden Angepasst( ca. 20 Stunden )
  - Kurz aber informativ
  - Gleiches Design bei alles Anfragen
  - Icons wurden Hinzugefügt, ähnlich wie bei der genutzten Api
  - mehrere Testläufe und Anpassungen, vor allem bei relativer Größe des Telegram Messenger
  - Wohnort wurde in Nachrichten mit eingefügt
- 2. Anfragen wurden Um Zeit erweitert( ca. 30 Stunden)
  - Requests werden jetzt auch mit der eingegebenen Zeit berücksichtigt, sprich wetter morgen um 14 uhr wird mit gezählt
  - diverse Fehleinschätzungen wurden entfernt
  - Zeitrechnung von UTC zu GMT wurde angepasst und optimiert
  - Zeitzone unabhängig geschrieben, bei Beibehalten von Date Format, sprich kein cutten in String oder so ähnlich
- 3. History Anfrage wurde hinzugefügt( ca. 11 Stunden)

- es ist nun möglich 5 Tage in die Vergangenheit zu fragen
- 4. Wetteranfrage wurde erweitert( ca. 5 Stunden )
  - es wird nun ein neuer Api call gemacht, womit nicht nur 5 sondern 7 Tage in die Zukunft gefragte werden kann
- 5. Diverse Bug fixes, forallem im Format der Nachricht, aber auch im Zeitrechnungsbereich( ca. 3 Stunden)
- 6. Nach umstellung des Servers neuanpassung ans neue Backend(ca. 2 Stunden )
- 7. alter Cache wurde entfernt und in Registry eingebettet( ca. 20 Minuten )
  - Statt 6 Stunden werden alle Calls nun 30 min gespeichert
- 8. Erstellung des Vortrages( Eigener Part )( ca. 5 Stunden )
  - Erstellung der Folien
  - Erstellung des Skriptes, was gesagt werden soll
  - Grobe Planung, was wer sagen soll( Zeitplanung, Arbeitsverteilung )
- 9. Diverse schriftliche Rücksprachen über Discord, auch Meetings( ca. 7 - 10 Stunden )

Nutzungshinweis: Auf dieses vorliegende Schulungs- oder Beratungsdokument (ggf.) erlangt der Mandant vertragsgemäß ein nicht ausschließliches, dauerhaftes, unbeschränktes, unwiderrufliches und nicht übertragbares Nutzungsrecht. Eine hierüber hinausgehende, nicht zuvor durch *datenschutz-maximum* bewilligte Nutzung ist verboten und wird urheberrechtlich verfolgt.