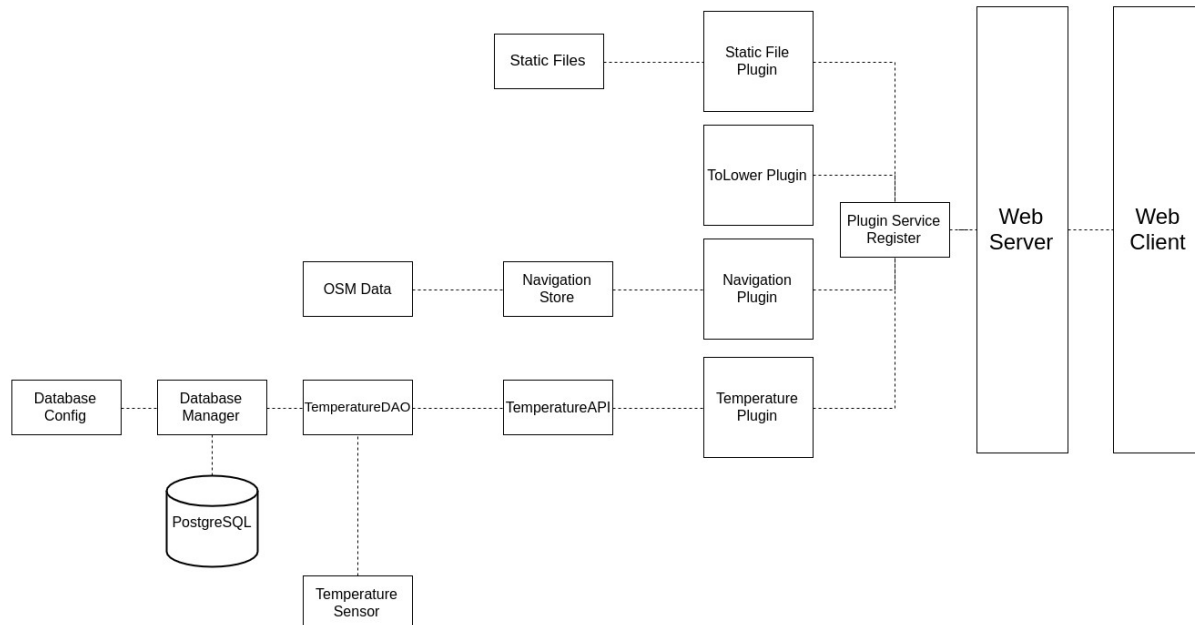


Software Architektur



Benutzerhandbuch – Wie wird die Applikation verwendet

Via HTTP Url kann man die Applikation in einem Web Browser aufrufen. Auf der Startseite sieht man Hauptinformationen über den App. Im Hauptmenu, kann man folgende Kommandos sehen: *ToLower*, *Navigation*, *Temperature*. Die Beschreibung für jedes Plugin finden Sie im Text unter:

1. ToLower Plugin:

Auf dieser Seite sieht man Text-Input Field, wo man den Text schreiben kann. Unter befindet sich ein Button, das eingegene Text in Kleinbuchstaben konvertiert

2. Navigation Plugin:

Mit diesem Plugin kann man sehen, welchen Straßennamen sich im welchen Stadt befindet. Vor man nach Straßennamen suchen kann, muss man zuerst eine Karte neuladen (Es gibt ein Button der das im Hintergrund macht). Dannach kann man im Textarea Straßennamen eingeben und mittels Button nach Name suchen. Als Ausgabe sieht man eine Liste mit allen Stadtnamen.

3. Temperature Plugin:

3.1 Normale Verwendung – Man sieht eine Liste mit Temperatur Ergebnisse (20 Ergebnisse pro Seite). Unter der Liste, sieht man auch zwei Buttons für das nächste oder vorige Seite. Auf der Seite befindet sich auch Datum Picker, womit man ein Datum auswählen kann, um die Temperatur-ergebnisse auslesen zu können.

3.2 REST API – Es gibt auch Unterstützung für REST Requests. Schreibt man z.B.

<http://localhost:8080/GetTemperature/2019/01/01>, bekommt man alle Temperatur-Ergebnisse für das 1. Jänner 2019 im JSON Format.

Lösungsbeschreibung – Wie wurde die Aufgabe gelöst

Zuerst wurden alle angegebene UnitTests implementiert. Für manche Aufgaben, z.B Temperature oder Navigation Plugin, wurde eigene Implementierung eingesetzt. Mehr technischen Informationen finden Sie in diesem Dokument unter "Software Architektur" Abschnitt. Dort wurde gezeigt, wie alle Programmschnitte funktionieren zusammen.

Worauf bin ich stolz

- Verwendung von Singleton Pattern wenn möglich (Managers, Navigation Speicher)
- Verwendung von Data Access Object (DAO) Pattern – Temperature
- Config (Datenbank und Plugins) möglich extern zu ändern -> daher das Neustarten des WebServers benötigt wird

Was würde ich das nächste mal anders machen

- Statt meiner Service Register Implementierung, Java SPI implementieren und verwenden
- Bessere Unterstützung für neue Plugins – mehrere CustomAPIs für Customizires
 - WebClient für Plugins jetzt hard-coded, keine Möglichkeit für das Customizing – in Zukunft wäre es besser mittels Verwendung von Modules (ein Modul – ein Plugin), wo man auch den Client Code schreiben kann, und mit Hilfe des Customizing APIs Client Commands zu registrieren im Core Code-Base (MyWebServer).
- Bei Temp-Sensor, statt *Thread.sleep()*, Events verwenden