Ein Bild, das Schrift, Grafiken, Logo, Grafikdesign enthält.

Automatisch generierte Beschreibung

Nepomuk Crhonek

Noser Young AG/Accenture

Arbeitsjournal

A purple arrow on a black background

Description automatically generated

KW13/2025

|  |  |  |
| --- | --- | --- |
| Wochentag | Tätigkeit | Zeit in Minuten |
| Montag  24.03.25 TBZ | Eng Bewertung besprochen  Mat Prüfung geschrieben  ABU An der Präsentation von Buch gearbeitet  **Total** | 90  90  180  **360** |
| Dienstag  25.03.2025  TBZ | Modul 319 An SQL-Aufgaben gearbeitet  Modul 231 gearbeitet an Webshop Projekt **Total** | 180  180  **360** |
| Mittwoch  26.03.2025  NY | Beginn üK295 Info-Input zugehört  Input zu Spring (Boot) zugehört  An den Spring (Boot) Aufgaben gearbeitet  Input zu JPA angehört  **Total** | 33  107  323  41  **504** |
| Donnerstag  27.03.2025  NY | An den Backend Aufgaben weitergearbeitet  Input zu Error Handling erhalten  Input zu Testing angehört  **Total** | 376  72  56  **504** |
| Freitag  28.03.2025  NY | Input zu Custom Queries und Security Theory erhalten  Projektauftrag erhalten  Am Projektauftrag weitergearbeitet  **Total** | 52  13  439  **504** |

Wochenrückblick

Diese Woche musste ich mein Deutsch Buch zurückbringen da es abgelaufen ist. Da ich das Buch gelesen habe sollte es einfach sein, die Präsentation über das Buch nächste Woche vorstellen. Nächste Woche werden wir auch andere wichtige Prüfungen haben. Ich fühle mich aber relativ sicher. Wir mussten diese Woche sehr viel dazu lernen um gut mit Spring (Boot) klar zu kommen.

Reflexion

Ich habe diese Woche sehr viel neues gelernt im üK 295 wie zum Beispiel das grundlegende Verständnis von REST-APIs und HTTP-Statuscodes. Die Bedeutung der verschiedenen Statuscode-Kategorien (1xx, 2xx, 3xx, 4xx, 5xx) ist nun klarer für mich. Besonders wichtig war das Verständnis der REST-Architektur mit ihren Prinzipien wie Client-Server-Architektur, Zustandslosigkeit und einheitliche Schnittstelle.

Die dreischichtige Architektur (Controller-Service-Repository) in Spring Boot ist nun für mich verständlicher. Ich kann jetzt besser nachvollziehen, wie die verschiedenen Komponenten zusammenarbeiten, und wie Dependency Injection durch Annotationen wie @Autowired implementiert wird.

Das Arbeiten mit Spring Boot hat mir gezeigt, wie viel boilerplate Code uns durch das Framework erspart bleibt. Annotationen wie @RestController, @Service und @Repository machen die Entwicklung effizienter und strukturierter.

Auch die Fehlerbehandlung mit @ControllerAdvice und dem GlobalExceptionHandler war ein wichtiger Lernpunkt. Die Möglichkeit, Exceptions zentral zu verwalten, verbessert die Codequalität erheblich.

Die zentrale Fehlerbehandlung mit @ControllerAdvice und dem GlobalExceptionHandler verbessert die Codequalität erheblich, indem sie den fehlerbehandelnden Code von der Geschäftslogik trennt und dadurch sowohl die Wartbarkeit als auch die Lesbarkeit des Codes steigert. Durch diese Struktur werden zudem Fehlerantworten konsistent formatiert und redundanter Code vermieden, da die Fehlerbehandlungslogik nur an einem zentralen Ort implementiert und gewartet werden muss.

Die Arbeit mit UUIDs als Primärschlüssel statt Auto-Increment-IDs war neu für mich. Ich verstehe jetzt die Vor- und Nachteile beider Ansätze und wann UUIDs besser geeignet sind, besonders in verteilten Systemen.

Stimmung der Woche

Mir geht es gut. Mit Kollegen auch alles gut.