

praxisarbeit m165

Dokumentation nach IPERKA



Davud Ponjevic

Inhaltsverzeichnis

[1 Informieren 2](#_Toc168312947)

[1.1 Ausgangsituation 2](#_Toc168312948)

[1.2 Anforderungen 2](#_Toc168312949)

[2 Planen 3](#_Toc168312950)

[2.1 Trello 3](#_Toc168312951)

[3 Entscheiden 4](#_Toc168312952)

[4 Realisieren 5](#_Toc168312953)

[4.1 Ergebnis 5](#_Toc168312954)

[4.1.1 Swagger 5](#_Toc168312955)

[4.1.2 Postman 10](#_Toc168312956)

[4.2 Datenmodel 11](#_Toc168312957)

[5 Kontrollieren 12](#_Toc168312958)

[6 Auswerten 13](#_Toc168312959)

[6.1 Fazit 13](#_Toc168312960)

# Informieren

Im ersten Schritt von IPERKA, dem "Informieren", suche ich nach Informationen über meine Arbeit. Ich sammle Dokumente, spreche mit Kollegen und suche nach relevanten Quellen. Dabei achte ich darauf, dass die Informationen aktuell und zuverlässig sind. Ich berücksichtige auch externe Einflüsse wie Gesetzesänderungen oder Branchentrends. Das Ziel ist, eine solide Basis zu schaffen, um die Anforderungen meiner Aufgabe zu verstehen und effizient umzusetzen.

## Ausgangsituation

Die Firma Jetstream-Service führt als KMU in der Wintersaison Skiservicearbeiten durch und hat in den letzten Jahren grosse Investitionen in eine durchgängige digitale Auftragsanmeldung und Verwaltung, bestehend aus einer datenbankbasierender Web-Anmeldung und Auftragsverwaltung getätigt.

Aufgrund guter Auftragslage hat sich die Geschäftsführung für eine Diversifizierung mit Neu-eröffnungen an verschiedenen Standorten entschieden.

Die bis anhin eingesetzte relationale Datenbank genügt den damit verbundenen Ansprüchen an Datenverteilung und Skalierung nicht mehr. Um einerseits den neuen Anforderungen gerecht zu werden sowie anderseits Lizenzkosten einzusparen, soll im Backend der Anwendung die Datenbank auf ein NoSQL Datenbanksystem migriert werden.

## Anforderungen

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 1 Anforderungen

# Planen

Im zweiten Schritt von IPERKA, dem "Planen", erstelle ich einen klaren Plan für meine Arbeit. Ich setze mir konkrete Ziele, teile die Aufgaben in kleine Schritte auf und überlege mir, wie viel Zeit ich für jeden Schritt benötige. Dabei berücksichtige ich mögliche Herausforderungen und denke über Lösungen nach. Zusätzlich identifiziere ich die Ressourcen, die ich für die Umsetzung meiner Pläne benötige, sei es Material oder Unterstützung von Kollegen. Durch die Priorisierung von Aufgaben und die Erstellung eines gut strukturierten Plans behalte ich den Überblick und steigere die Effizienz meiner Arbeit, was letztendlich dazu beiträgt, meine Ziele erfolgreich zu erreichen.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 2 Zeitplan

Beim anschauen dieses Zeitplanes denken sie sich warum 40 stunden für das ganze Programm, die ist einfach zu erklären weil ich in den ersten paar stunden nur am recherchieren war wie ich das ganze anfangen soll. Sobald ich angefangen habe musste ich oft lang recherchieren wie und warum Probleme entstanden sind und deshalb habe ich insgesamt eine Woche gebraucht.

## Trello

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

Automatisch generierte Beschreibung

Hier habe ich eine kleine Planung gemacht, hier ziehe ich den schritt den ich mache muss in die Spalte in welcher Phase der schritt ist.

# Entscheiden

Im Schritt "Entscheiden" von IPERKA stand für mich die Wahl zwischen Neo4j und MongoDB an. Nach gründlicher Überlegung habe ich mich letztendlich für MongoDB entschieden. Diese Entscheidung basierte auf meiner Einschätzung, dass mir die Arbeit mit MongoDB leichter von der Hand gehen wird. Dabei habe ich berücksichtigt, dass MongoDB besser zu meinen Kenntnissen und meinem Arbeitsstil passt. Die klare Struktur und die Benutzerfreundlichkeit dieser Datenbanklösung spielten eine entscheidende Rolle bei meiner Wahl. Durch diese bewusste Entscheidung setze ich auf Effizienz und Vertrautheit, was die Umsetzung meiner Aufgaben erleichtert und einen reibungslosen Arbeitsprozess fördert.

# Realisieren

Im Schritt "Realisieren" von IPERKA habe ich mich darauf konzentriert, den Code für mein Projekt zu schreiben und diesen anschließend zu dokumentieren. Während dieses Prozesses wurde mir deutlich, dass ein beträchtlicher Teil meiner Zeit in die Recherche investiert wurde, um sicherzustellen, dass der Code den Anforderungen entspricht und qualitativ hochwertig ist. Die intensive Recherche war notwendig, um mögliche Probleme frühzeitig zu erkennen und effektive Lösungen zu finden. Obwohl der Zeitaufwand für die Recherche höher war als für die eigentliche Programmierung, trug dies dazu bei, eine solide Grundlage zu schaffen und potenzielle Fehlerquellen zu minimieren

## Ergebnis

### Swagger

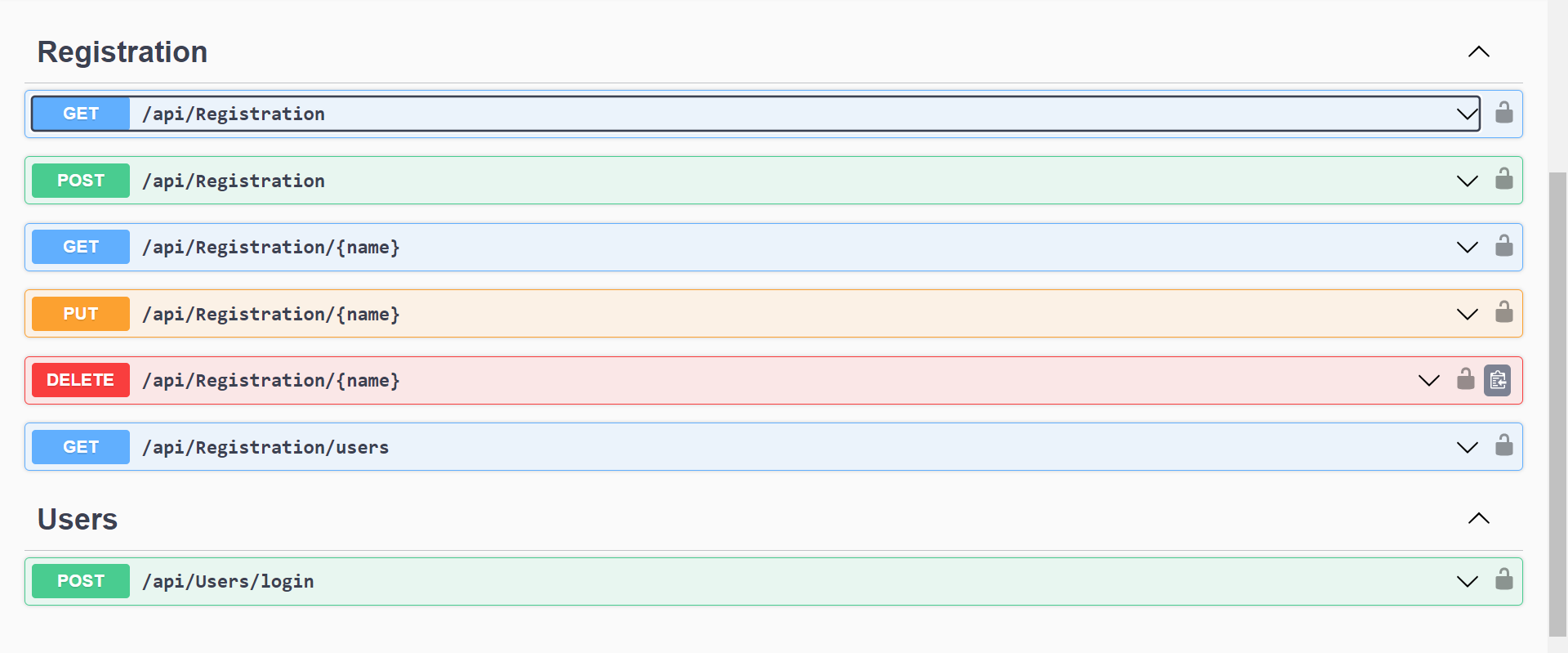


Abbildung 3 Swagger

Hier sieht man den Swagger teil. Alles unter dem Registration Titel hat mit den Bestellungen zu tun.

#### Alle CRUD Befehle einzeln

##### Get-All

Ein Bild, das Text, Screenshot, Software, Webseite enthält.

Automatisch generierte Beschreibung

Abbildung 4 Swagger Get\_All

Hier sieht man wen man das «Get» ausführt beim ersten bekommt man alle Resultate welche in der Collection Registration sind.

##### Post

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

Automatisch generierte Beschreibung

Abbildung 5 Post

Hier posten eine Bestellung zur Datenbank hinzu, genauer gesagt noch in die dazugehörige Collection («Registration»). Hier sehen wir das die Bestellung mit dem Namen «du» hinzugefügt wird. Wir müssen unbedingt den Namen, die "priorityId" und die "serviceId" setzten weil es Felder sind welche Required sind, das heisst die Datenbank braucht diese Felder um diese Bestellung zu erstellen.

##### Get-By-Name

Ein Bild, das Text, Screenshot, Software, Webseite enthält.

Automatisch generierte BeschreibungHier sehen wir das wir den User welchen wir im vorherigen schritt erstellt haben und abfragen ob wir ihn in der Datenbank finden können.

Abbildung 6 Get-By-Name

##### Put

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

Automatisch generierte BeschreibungHier beim Put ändere ich den Namen bei der Bestellung wo der Name = «du» ist auf «update». Das heisst die Bestellung wo der Name «du» war existiert nicht mehr man findet diese Bestellung nur noch unter Namen «update».

Abbildung 7 Put

##### Delete

Für diesen Schritt muss ich erstmals den Token holen (im Schritt «Post User» genauer erklärt)

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

Automatisch generierte BeschreibungSobald wir den Token oben beim Authorize eingegeben haben kann ich in Namen eingeben beim Feld und wir sehen wenn wir das ausführen das dieser Datensatz gelöst wird.

Abbildung 8 Delete

##### Get-Users

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

Automatisch generierte BeschreibungIch habe extra noch eine Funktion eingefügt die Get-user Funktion und zwar mit dem Gedanken das diesen Befehl nur der Admin sieht und falls man die Einlogge Daten vergisst das man diese wieder sieht. Wen man das ausführt bekommt man einfach alle Users mit welchem man ein Token erstellen kann (Schritt «Post User»)

Abbildung 9 Get-Users

##### Post-User

Ein Bild, das Screenshot, Text, Software, Computersymbol enthält.

Automatisch generierte BeschreibungHier in diesem schritt erstellt man keinen User sondern eigentlich ist es dazu das man den Token bekommen kann. Man muss sich zuerst im «Request Body» anmelden mit den Einlogge Daten, nachdem bekommt man im «Server Response» unter der Eigenschaft «Token» bekommt man den Token diesen fügt man oben bei Authorize ein (Bild 11). Wen man das gemacht hat kann man ein Delete machen welches ohne Einloggen nicht gehen würde

Abbildung 10 Post-User

Ein Bild, das Text, Software, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

Hier muss man den Token einfügen

Abbildung 11 Authorize

### Postman

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Das einzige was man beim Postman beachten muss ist das man immer die richtige URL nimmt. Man nimmt die URL von den Befehl fügt sie im Postman ein

Beispiel;

Ein Bild, das Text, Screenshot, Software, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 12 Request-URL Postman

Hier wen ich abfragen das man alle Bestellungen aus der Datenbank bekommt geht man in den Swagger und nimmt die Request URL und fügt sie Im Postman hier ein;

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

Automatisch generierte Beschreibung

Abbildung 13 Postman

## Datenmodel

Ein Bild, das Text, Diagramm, Screenshot, Reihe enthält.

Automatisch generierte Beschreibung

Abbildung 14 Datenmodel

Wie man das verstehen kann ist so: Die Collection nimmt den PK der Tabelle Priority und fügt ihn zu Bestellung hinzu so das man weiss welche Priorität eine Bestellung hat, als zweitens nimmt die Collection den PK aus Service so man weiss welchen service eine Bestellung hat und fügt ihn zur Tabelle hinzu, als letztens die Verbindung zwischen User uns Bestellung die ist dazu da so das ein User die Bestellung nimmt und diese dann erarbeitet

# Kontrollieren

Im "Kontrollieren"-Schritt von IPERKA habe ich mein Programm mit Postman getestet, um sicherzustellen, dass es den festgelegten Anforderungen entspricht. Dabei wurden verschiedene Szenarien und Eingaben überprüft, um Fehler zu identifizieren und zu beheben. Die Nutzung von Postman ermöglichte eine effiziente Kontrolle der Software, um eine zuverlässige und fehlerfreie Funktionalität sicherzustellen.

# Auswerten

## Fazit

Insgesamt bin ich mit dem Verlauf von IPERKA zufrieden und habe sogar Spaß an der Arbeit, insbesondere mit MongoDB. Anfangs hatte ich einige Startprobleme und war unsicher, wo ich beim Programmieren beginnen sollte. Trotz dieser Herausforderungen konnte ich jedoch einen erfolgreichen Startpunkt finden. Die Auseinandersetzung mit MongoDB bereitet mir Freude, und ich bin beeindruckt von den Möglichkeiten dieser Datenbanklösung. Durch die Überwindung der anfänglichen Unsicherheiten fühle ich mich motiviert, mein Wissen weiter auszubauen und MongoDB intensiver zu verfolgen. Diese Erfahrung hat gezeigt, dass auch bei anfänglichen Schwierigkeiten ein erfolgreicher Weg gefunden werden kann, und ich bin zuversichtlich, dass dies zu weiterem Lernfortschritt und erfolgreichen Projekten führen wird.