

2024

NoSQL Anbindung

Modul 165

Mahir Gönen



Table of contents

01

Informieren

Anforderungen

03

Entscheiden

Referenzen

05

Kontrollieren

Testprojekte

Planen

Planung, Anwendungsdesign,
Datenmodell

Realisieren

Backend, Skripts

Auswerten

Fazit

02

04

06

2024

01

Informieren



01 Anforderungen

Datenmigration durchführen

- Daten vom relationalen SQL in NoSQL migrieren
- Skripte für Schema, Index und automatisches Backup erstellen
- API CRUD Migration
- Datenmodell grafisch darstellen

Projektdokumentation nach IPERKA durchführen

Postman Testprojekt erstellen

Optionale Anforderungen

- Automatisiertes Backup-Konzept durchgeführt und implementiert
- Komplexe Schema Validierungen umgesetzt (Referenzen, enum, min, max. usw)



2024

02

Planen



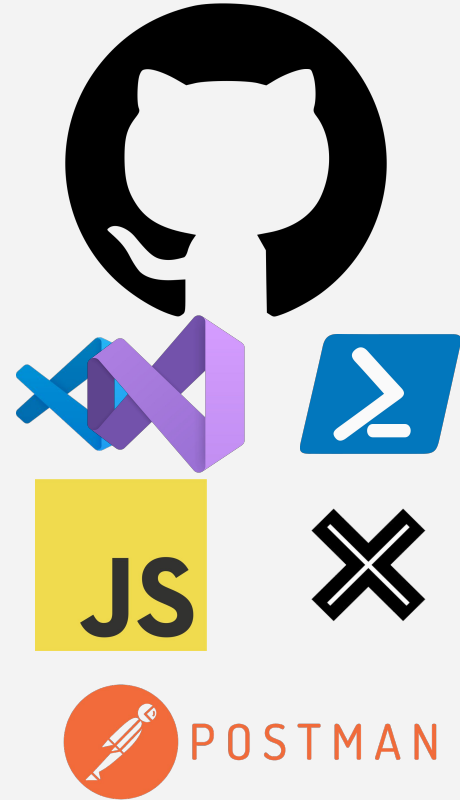
02 Planung

Skripte

- Schemas: Datentypen, aktuellen Datenanforderungen (relationale SQL)
- Index: Schlaue Indexe erstellen um Leistung zu fordern
- Seed-Daten: C# Initialisierung
- Migration von relationale SQL: C# Initialisierung

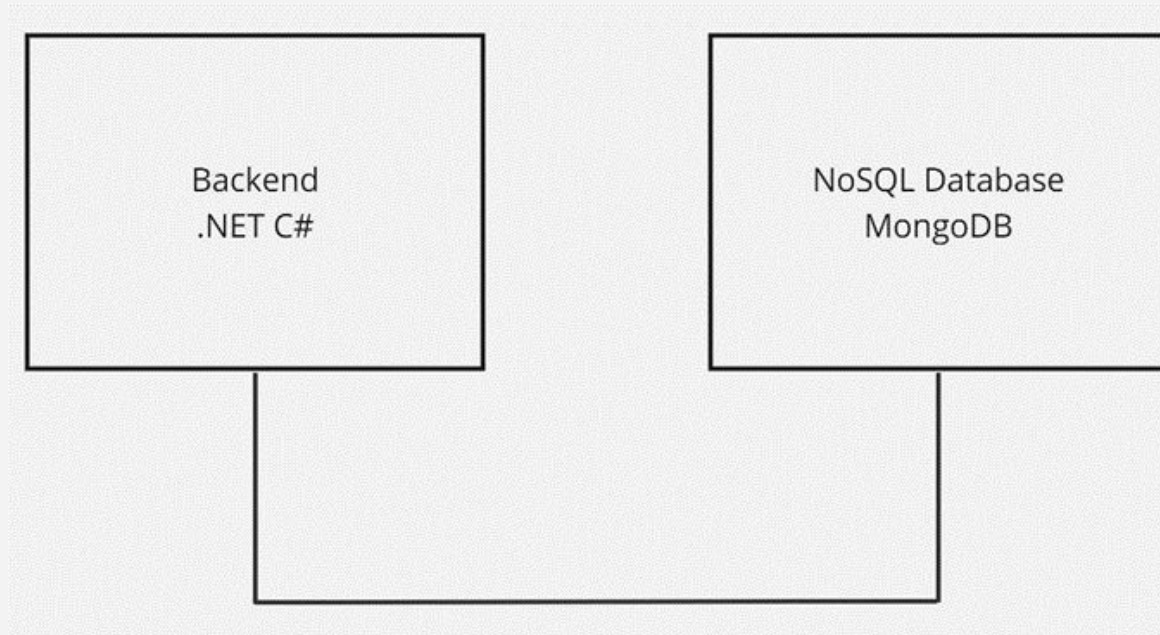
Technologiestack

- Entwicklung: VSC, VS, C# .NET, MongoDB, PowerShell, JavaScript
- Verteil- und Versionsverlaufsystem: GitHub
- Testing: xUnit .NET, Postman

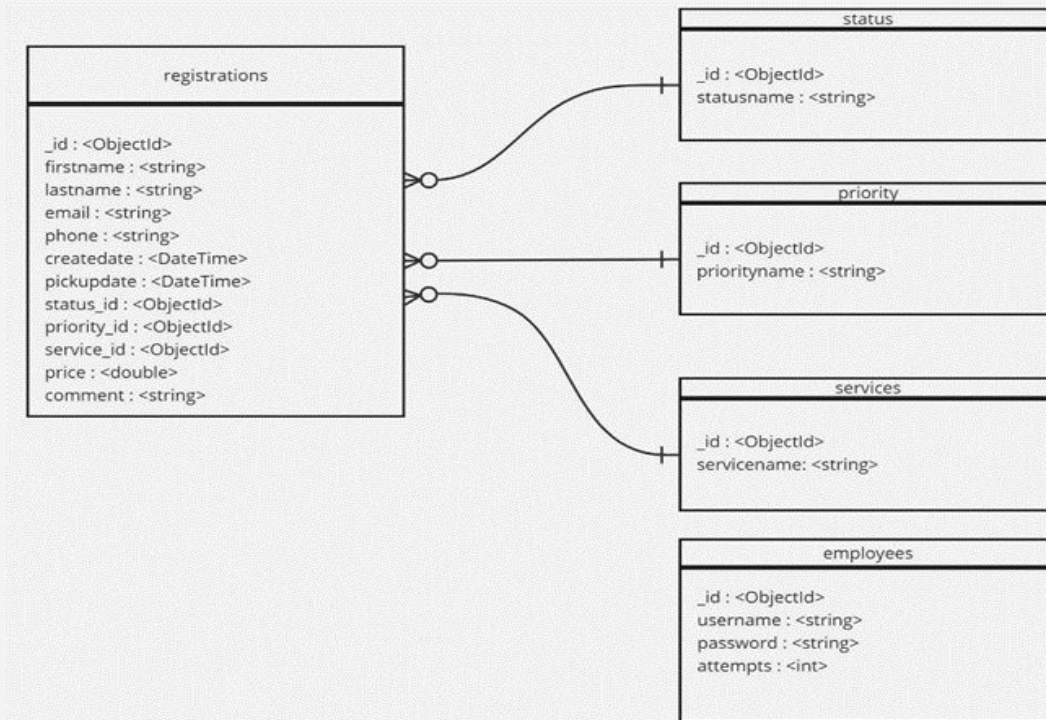


02 Anwendungsdesign

Einfacher Visualisierung der Anwendungen



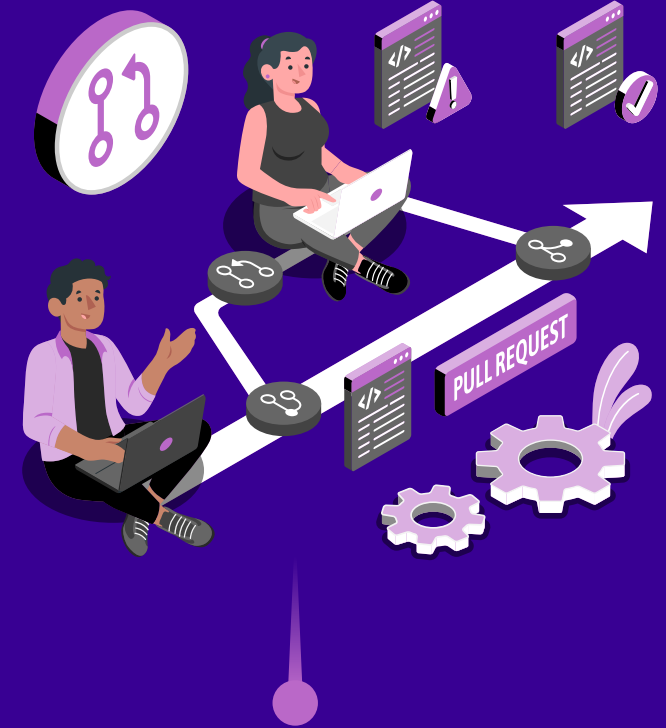
02 Datenmodell



2024

Entscheiden

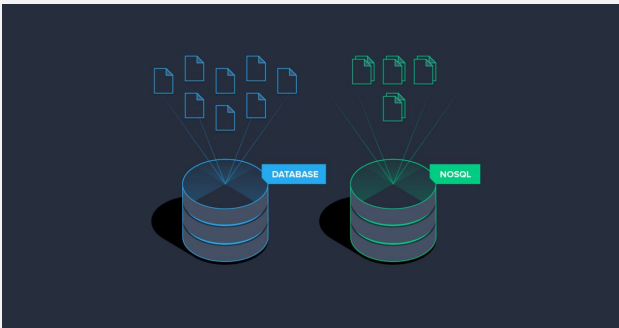
03



03 Referenzen

Entscheidung auf Referenzen statt Embedded Documents

- Aufgrund Leistungssteigerung
- Optimale Nutzung von NoSQL Vorteile
- Vereinfachung der Datenpflege
- Skalierbarkeit und Erweiterbarkeit



2024

04

Realisieren



04 Backend

Aktualisierung auf MongoDB-Treiber

- Implementation und Aktualisierung von EFCore auf MongoDB

Anpassung der Geschäftslogik

- Auf neuen Datenstruktur angepasst

Test und Qualitätssicherung










- Nach Aktualisierung wird alles getestet



04 Skripts

Skripts nach Planung implementiert und eingesetzt

- Schemaskript
- Indexskript
- Datenbankinitialisierung
- Backup- und Restoreskript
- Automatisiertes Backupskript
- Dokumentation: README.txt im Skriptordner

 SQLMigrations	05.02.2024 11:10	Dateiordner
 AccessControl.js	05.02.2024 15:23	JavaScript-Quelldatei
 Index.js	06.02.2024 17:22	JavaScript-Quelldatei
 InitializeDatabase.ps1	06.02.2024 21:30	Windows PowerShell...
 MongoBackup.ps1	06.02.2024 15:28	Windows PowerShell...
 MongoRestore.ps1	05.02.2024 16:37	Windows PowerShell...
 README.txt	06.02.2024 22:29	Textdokument
 SchemaValidation.js	06.02.2024 21:24	JavaScript-Quelldatei
 WindowsTaskplanner.ps1	06.02.2024 15:27	Windows PowerShell...

2024

05

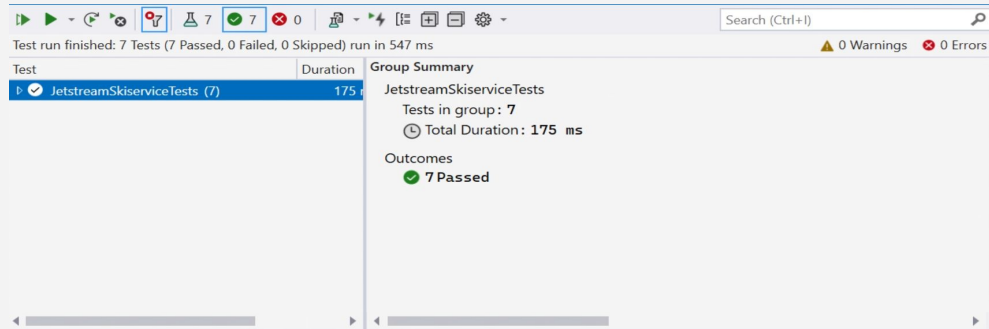
Kontrollieren



05 Testprojekte

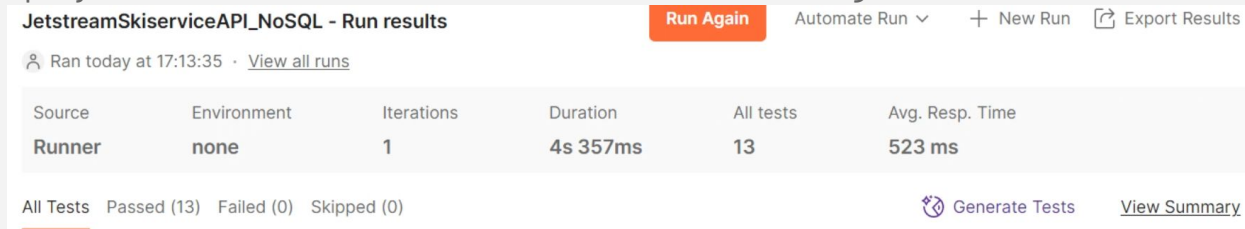
Testprojekt xUnit

- Testprojekt wurde nach dokumentierter Testplan erstellt



Testprojekt Postman

- Testprojekt über Postman Collection wurde erstellt und getestet



2024

06

Auswerten



06 Fazit

Erfolgreiches Projekt - Migration komplett gemacht

Durch Tests Qualität gesichert

Herausforderndes Projekt

Skripts erfolgreich implementiert

