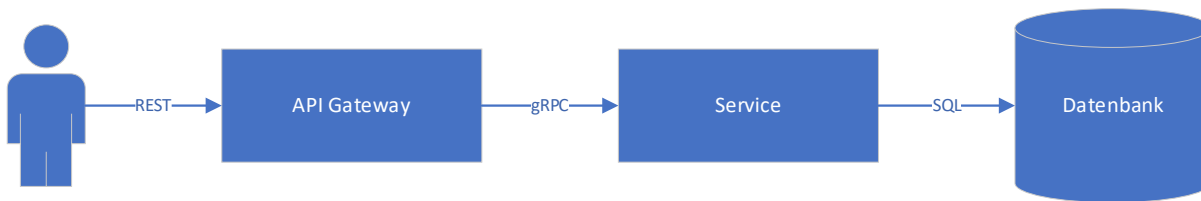


Es soll eine Anwendung erstellt werden, mit der man Kundendaten in einer Datenbank speichern und auslesen kann.

Die Anwendung soll aus den Komponenten API Gateway, Service und Datenbank bestehen. Der Benutzer soll über das API Gateway Kundendaten schreiben und auslesen können.

Das API Gateway sendet nach Kundeneingabe die Daten an einen Service, der sämtliche Methoden für die Datenbankkommunikation beinhaltet.



Service + Datenbank erstellen

- Docker installieren
 - o <https://docs.docker.com/desktop/install/windows-install/>
- Neues Projekt für einen Service mit der Projektvorlage „ASP.NET Core leer“ erstellen.
- Erstellung eines Datenbankmodells inkl. Kontext mit folgenden Tabellenspalten:
 - o Customer
 - Firstname (string)
 - Lastname (string)
 - Age (int)
 - Address (string)
- Anwenden des Datenbankmodells per Code First auf eine lokale Datenbank in einem Dockercontainer.
- Erstellen von DataWorker zum Erstellen und Auslesen von Customer-Daten.
 - o Beim Schreiben von Customer-Daten sollen diese über einen Validator validiert werden.
- Registrieren der DataWorker und Datenbankkontext per Dependency Injection.
- Erstellen von gRPC Methoden zum Erstellen und Auslesen von Customer-Daten.

API Gateway erstellen

- Neues Projekt für das API Gateway mit der Projektvorlage „ASP.NET Core leer“ erstellen.
- Swagger einrichten
- Controller implementieren

Bonus

- Zusätzlich kann ein Webfrontend erstellt werden, das die Benutzereingaben annimmt und die entsprechenden Endpunkt des API Gateways anspricht.

Quellen:

- \CloudAutomationvNext\api\src\api_rest_intern\api_rest_intern.csproj
- \CloudAutomationvNext\services\src\ticket\Ticket.csproj
- \CloudAutomationvNext\docker-compose.yml => sql
- <https://learn.microsoft.com/en-us/dotnet/core/extensions/dependency-injection>
- <https://learn.microsoft.com/en-us/ef/ef6/modeling/code-first/workflows/new-database>
- <https://learn.microsoft.com/en-us/aspnet/core/grpc/basics?view=aspnetcore-7.0>