

## Verwendete Technologie:

- Rest
- .Net Core
- MVC
- SQL Server
- Windows Service
- Entity Framework

## Projektbestandteile:

- **Framework.Core** : Diese Class Library Projekt, das ich vor 2 Jahren geschrieben und für verschiedene Projekte verwendet, ist DataLayer Projekt. Dies ermöglicht eine einfachere Nutzung des Entity-Frameworks, ohne eine Menge Code neu zu schreiben.
- **CoffeeService**: Ein WebAPI-Dienst, der alles für Koffee bereitstellt (Einfügen, Aktualisieren, Löschen, Suchen, Kafeebestand und ...).
- **StoreService**: Ein WebAPI-Dienst, die Alles für Store bereitstellt (Einfügen, Aktualisieren, Löschen, Suchen, Zeitplan und...).
- **Coffee.APIProvider** : Ich habe 2 Client-Projekte(Windows-Dienst für Coffee Factory und WebApplicaion für Kunden und Management), die von meinen APIs verwenden. Um keinen doppelten Code für die Verbindung zu APIs zu schreiben, habe ich diese Class Library entwickelt und sie in beiden Projekten verwendet.
- **Coffee.WebApplication**: Eine MVC Project, die für Kunden und System-meister entwickelt wurde. In der nächsten Schritten werde ich sie im Detail erzählen.

- **CoffeeFactory.WindowsService:** Ein Dienst, der für Coffee Factory geschrieben wurde. Wenn sie geläuft wird, wird jede 5 Sekunde ein Kaffee produziert.

## Business Logic:

Weil Teile des Business Logic für mich unverständlich und fragwürdig waren, habe ich es so umgesetzt, wie ich es für richtig hielt. Ich hoffe, das ist kein Problem.

Natürlich benötige dieses system ein Authentication-System. Aber jetzt wird es nicht entwickelt.

### Funktionsweise des Systems:

1. Der Kaffee wird in der Kaffee Fabrik jede 5 Sekunde hergestellt und dann in einem Paket verschickt. Die Anzahl Jeder Packung hängt von Coffee Type.

#### CoffeeFactory.WindowsService

2. In der Kaffee fabrik gibt es einen oder mehrere Benutzer, die entscheiden, welche Pakete zu welchem Zeitpunkt und an welches Geschäft versandt werden müssen. Hier wird der Arbeitsplan des Shops so gesteuert, dass er geöffnet ist. Das logistische System (Fahrer, Auto, Kapazität der Autos und ...) ist nicht entwickelt.

#### Menu:(Management/Products)

3. Wenn die Pakete empfangen wurden, ändert ein anderer Benutzer im Coffe Shop den Status der Pakete auf "Recieved". (Management/RecieveCoffees) Danach wird die Menge in das Inventar des Coffeeshops aufgenommen und die Kunden können im Menü (Home) die Menge sehen (die Bestellung wurde in diesem System nicht entwickelt).
4. Menü(Coffee Shops): In diesem Menü kann mann die Informationen von den Coffee Shops und deren Arbeitzeiten eingeben.

## wie kann man die Projekt laufen?

1. Zuerst CoffeService und StoreService ausführen (Microsesrvice für den Dienst am Clients)
2. Run Coffee.WebApplication as Client (Kunden und Management Möglichkeit)
3. Run CoffeeFactory.WindowsService as Coffee Factory (Coffee zu producieren)

## SQL Info:

Server: 45.83.104.117

User: nodapo

Pass: ra8G@kfhXmrr

Natürlich hat dieses System viele andere Details und andere Punkte zu entwickeln und zu kontrollieren, die in meinem Kopf existieren, aber ich denke, dass ich in diesem kleinen Projekt zeigen kann, wie ich Code schreibe und die Software entwickle.

Wenn Sie Frage haben, stehe ich Ihnen gerne zur Verfügung.