ChipSoft Assessment documentatie

Hier is even kort wat uitleg over de software. Ik heb aan deze documentatie niet veel tijd besteed, sinds aannam dat ook nog uitgebreid gaan bespreken.

Ik had aangenomen dat er een source of truth is, dus dat alle communicatie met een zorgplatform verloopt en de API die ik heb ontwikkeld voor deze opdracht is van het zorgplatform, die dan alle data heeft en dan kunnen de zorginstanties in dit voorbeeld dan (Ziekenhuis A en Verpleeghuis B) via deze API bij de data die ze willen hebben.

Daarom als je in het huidige systeem toegang wil verlenen tot de allergieën aan een andere organisatie, is het alleen maar nodig om die toevoegen als bemachtigde aan het patiëntendossier. Nu krijgt de organisatie toegang tot alle bestanden van de patiënt, maar in een meer realistisch scenario, zou je waarschijnlijk, maar beperkte rechten verlenen. Sinds authenticatie niet geïmplementeerd is kunnen organisaties overal bij en is er geen check of ze bevoegd zijn op het moment.

Solution Structuur

Het project is opgesplitst in drie verschillende projecten. Deze projecten zijn lagen met een 1 op 1 verhouding. De API is de hoogste laag. Die communiceert met de BusinessLogic en de BusinessLogic communiceert met de repository laag. Deze opsplitsing is vooral gedaan zodat UnitTesting makkelijker zou zijn in de toekomst omdat de logica in de BusinessLogic project gebeurt.

De API is waar de requests worden geregeld in de controllers. Er zijn ook modellen zodat de Outside world anders kan zijn dan de inside world.

BusinessLogic, hier wordt de echte logica uitgevoerd, dit is in deze simpele business case erg beperkt.

Repository, is waar de entiteiten worden vastgelegd en de repository die ideaal, met de database zou interacteren, maar dat is niet geïmplementeerd dus is gewoon in-memory.

Requests

Er zijn 4 requests geïmplementeerd.

POST /patient/{patientId}referral, waarin je een pdf-verwijsbrief moet gegeven en een string van huidig gemachtigde organisatie in dit geval “Ziekenhuis A”, dan kan rechten geven aan “Verpleeghuis B”, er is een patiëntenId verwijst van de patient die je doorverwijst. Die kun je niet aanmaken op het moment, maar het patientId dat werkt is “**9b2d6c0a409946ffb03b8aadc883d5bf**”. Als reactie krijg je de referral en URI, naar waar je het kan ophalen.

GET /patient/{patientId}/referral/{referralId}, kun je referrals ophalen als je het Id hebt.

GET /patient/{patientId}/medicine, haalt alle medicijnen op voor gegeven patiënt.

POST /patient/{patientId}/allergies, kun je een allergie toevoegen voor een patiënt.

Logging:

In het aspect van logging, wordt alles dat door de client fout gedaan wordt aangegeven met een warning, en verder is alles informatie logs.

Validatie

Er wordt altijd gecheckt dat de patiënt bestaat. Er wordt gekeken naar de filetype waar dat relevant is, zelfde geld voor refferalId. Ik heb geen UnitTests geschreven, sinds ik niet dacht dat het nodig was, want ik zag het niet in te requirements staan, als dat wel het geval is, kan ik die altijd nog toevoegen.

Entiteiten

Voor het makkelijker te kunnen uitbreiden staan Allergieën en medicijnen in records. Ik heb paar example fields gemaakt is niet representatief, normaal zou dat soort data types veel meer informatie bevatten en complexer zijn.