



Technische Documentatie - ASP GATAM   
ISW-IP  
Group 8

Applied Software Project

ondersteund door de

Mariën Sven

DePooterMarijn

AP Hogeschool

begeleid door het bedrijf

GATAM

Studenten: Van Vaerenbergh Nils, Makdesi Tony, Baqloul Haitam, Onya Lauren

Table of Contents

[Projectoverzicht 2](#_Toc180605108)

[Technische Stack 2](#_Toc180605109)

[Project Structure 2](#_Toc180605110)

[Setup en Installatie 3](#_Toc180605111)

[Belangrijkste Kenmerken 4](#_Toc180605112)

[Codeconventies en Architectuur 4](#_Toc180605113)

[Testen en Implementatie 4](#_Toc180605114)

[Toekomstige Verbeteringen 4](#_Toc180605115)

# Projectoverzicht

Projectnaam: ASP - GATAM ISW-IP  
Doel: dit project is gericht op het ontwikkelen van een kernapplicatie volgens de principes van Clean Architecture, met duidelijke lagen die applicatielogica, infrastructuurkwesties en gebruikersinterface beheren. Het doel is om een robuuste oplossing te bieden voor het effectief beheren van gegevens en het integreren van externe systemen.

# Technische Stack

- Programmeertaal: C#  
- Framework: .NET 8  
- Architectuur: Clean Architecture  
- Containerisatie: Docker  
- CI/CD: GitLab pipelines

# Project Structure

Het project is gestructureerd in duidelijke lagen volgens de principes van Clean Architecture. Elke laag heeft zijn eigen verantwoordelijkheden en maakt gebruik van specifieke technologieën en patronen.  
  
- **ApplicationCore/**

* **Beschrijving:** bevat de kernapplicatielogica en de business rules die de primaire functionaliteiten van de applicatie definiëren.
* **Submappen:**
  + **Gatam.Application/**: bevat de use cases en services die de bedrijfslogica implementeren en ook de interfaces die de communicatie met de infrastructuurlaag definiëren, zoals repositories en services.
  + **Gatam.Domain/**: bevat domeinmodellen en entiteiten,

- **Infrastructure/**

* **Beschrijving:** beheert infrastructuurkwesties, zoals toegang tot databases, externe API’s en andere services.
* **Submappen:**
  + **Gatam.Infrastructure.Cotexts/**: bevat de databasecontext en migraties voor SQL Server.
  + **Gatam.Infrastructure.Repositories/**: bevat implementaties van repositories die toegang bieden tot de gegevenslaag.

- **Presentation/**

* **Beschrijving:** beheert de gebruikersinterface en zorgt voor de interactie met de eindgebruikers.
* **Submappen:**
  + **Gatam. WebAppBegeleider/**: bevat de website toegankelijk voor de beheerders en de begeleiders. Deze is gebouwd op Blazor server framework.
  + **Gatam. WebAppVolger/**: bevat de website toegankelijk voor de gebruiker. Deze is gebouwd op Blazor server framework.

- **Solution Items/**

* **Beschrijving:** bevat gemeenschappelijke configuratiebestanden en hulpmiddelen voor de gehele oplossing.
* **Bestanden:**
  + **.gitignore**: definieert welke bestanden en mappen moeten worden genegeerd door Git.
  + **docker-compose.yml**: bevat configuraties voor het opzetten van de Docker-containers voor de verschillende lagen van de applicatie.
  + **README.md**: bevat algemene informatie en instructies voor het project.

# Setup en Installatie

# Voor de server op te zetten hebben we een setup script gemaakt. Dit script neemt de taken over om de nodige dependencies te installeren. Dit script zal je de nodige feedback geven tijdens het installeren. Zo kan je makkelijk aftoetsen of de installatie gelukt is. Hieronder volgt een stappen plan om de volledige deployment omgeving op te zetten.

## Overzetten van de nodige scripts.

# Hier heb je 2 opties. Je kan kiezen om git al te installeren en dan de repo te clonen naar de server of de nodige scripts overzetten via SFTP. Ik kies hier voor SFTP omdat dit ons toelaat specifieke bestanden over te zetten. Zo besparen we server opslag.

# Maak een SFTP verbinding naar de server.

# Kopieer de nodige bestanden van “asp-gatam-isw-ip/Server Setup” naar een locatie naar keuze. Ik kies hier voor /etc/gatam/setup/

## Uitvoeren van de scripts

# Om ons leven iets makkelijker te maken staat de hele setup in een setup.sh script. Deze zal de nodige dependencies installeren alsook de nodige files aanmaken.

# Voer setup.sh uit en volg deze op.

## Toevoegen van de nodige credentials

# Als de vorige stap correct werd uitgevoerd dan zal je zien dat er in de locatie /etc/gatam/ nu 2 .env bestanden aanwezig zijn. Deze zijn .env.staging en .env.production. Aan de hand van .env.example die te vinden is in het project, voeg je de nodige keys terug met hun waarden.

## Opzetten van de reverse proxy

# Nu alle nodige setup is gedaan kunnen we ervoor zorgen dat onze server domeinnamen accepteert en inkomende verbindingen. Om deze verbindingen naar de juiste service te sturen maken we gebruik van “traefik”. Deze zal achterliggende de nodige connecties maken met de domeinnamen.

# Voeg een .env toe in de setup folder. Deze neemt de keys over van de .env.example van de server setup.

# Voer DeployProxySetup.sh uit. Als alles goed verlopen is heb je nu 1 docker container met “traefik” als image.

## Aanmaken van runners.

# Belangrijkste Kenmerken

- **Gebruikersauthenticatie:** ondersteunt individuele authenticatiemechanismen voor inloggen en registreren  
- R**olgebaseerde Toegangscontrole:** beheert toegang tot specifieke delen van de app met rollen zoals admin, editor en gebruiker.  
- **Database-integratie:** SQL Server wordt gebruikt voor gegevensopslag.

## Codeconventies en Architectuur

Het project houdt zich aan de principes van Clean Architecture, waarbij verantwoordelijkheden in verschillende lagen worden gescheiden:  
1. **Application Core:** bevat de bedrijfslogica en applicatieregels.  
2. **Infrastructure:** beheert externe bronnen zoals databases en bestandsystemen.  
3. **Presentatielaag:** beheert de gebruikersinterface, waarschijnlijk met Blazor.  
  
Ontwerppatronen zoals CQRS en het Repository-patroon worden toegepast om de scheiding van verantwoordelijkheden en efficiënt gegevensbeheer te waarborgen.

# Testen en Implementatie

- **Unit- en Integratietests:** tests zijn te vinden in de tests/ map.  
- **CI/CD:** geautomatiseerd met GitLab-pijplijnen, die het project bouwen en implementeren in containerized omgevingen.

# Toekomstige Verbeteringen

-