



Ventixe - Dokumentation för Azure, Frontend & Microservices

Inledning Detta dokument sammanfattar alla implementationer, kopplingar och distributioner mellan frontend (React), backend (.NET 8 microservices) och Azure, för projektet *Ventixe* - en eventplattform utvecklad som MVP under kursen Molntjänster och Distribuerade System. Projektet använder ett Azure for Students-konto med minimal kostnad och offentlig GitHub-organisation för kodhantering.

1. Frontendstruktur och routing - React-applikationen använder `BrowserRouter` med autentiseringskontroll via JWT (lokalt sparad token). - Offentliga sidor: Login, Register, ConfirmEmail - Skyddade sidor (inloggning krävs): Events, My Events, Attending, My Page - Varje sida är responsivt designad, med olika vyer för desktop/tablet/mobil.

2. Frontendkomponenter per vy: - **Login/Register:** Full validering, samt resend-confirmation via AuthService - **My Events:** Skapa, ladda upp bild (Blob Storage), editera, radera event - **Events:** Kalenderbaserad vyn med filtrering per månad och paginering - **Attending:** Lista på events du gått med i, med Unattend-funktion - **My Page:** Profilställningar (username, email, lösenord, profilbild)

3. Microservices & API Alla mikrotjänster ligger på Azure Web App for Students:

- **AuthService** (<https://authservice-api-e6aghrh5e2gpc4a0.westeurope-01.azurewebsites.net>)
 - POST /api/Auth/register (validering + emailbekräftelse)
 - POST /api/Auth/login (JWT-token retur + email-confirm check)
 - POST /api/Auth/resend-confirmation
 - GET /api/Auth/confirm?email=
- **UserService** (<https://user-service-api-fgbuhbe9dmgb3gp.swedencentral-01.azurewebsites.net>)
 - GET /api/user/me
 - PUT /api/user/me (update email, username, password)
 - POST /api/user/me/upload-profile-image (Blob Storage)
- **EventService** (<https://eventservice-api-cebndaaheydrfbcs.swedencentral-01.azurewebsites.net>)
 - GET /api/events/month?year=&month=
 - GET /api/events/attending
 - GET /api/events/my-attending
 - GET /api/events/mine
 - POST /api/events/{id}/attend (eller DELETE för unattend)
 - POST /api/events (skapa event)
 - PUT /api/events/{id} (redigera event)
 - DELETE /api/events/{id}
 - PUT /api/events/upload-image (Blob Storage)
- **EmailService** (<https://emailservice-api-e4c5b9cnfxehg6h8.swedencentral-01.azurewebsites.net>)

- POST via ServiceBus från Auth/UserService
 - GET /api/health/servicebus (pingtest från frontend)
-

4. Azureresurser och säkerhet - Resursgrupp: `inlamningUppgift-Moln-Distsys` - Azure SQL Database: Gemensam för alla microservices (separerade tabeller) - Azure Key Vault: Används för API-nycklar, JWT-secret, SendGrid - Azure Blob Storage: Används för profilbilder + eventbilder - Azure Service Bus: Används för e-postkommunikation från Auth/UserService till EmailService

5. Publicering och GitHub - Reactfrontend deployas via Azure Static Web Apps från GitHub (main-branch) - Varje backendservice ligger i ett separat publikt repo i organisationen - CI/CD pipelines via GitHub Actions för push-to-deploy till Azure Web App

6. GDPR-stöd - GDPRModal på login/register visas första gången om cookies saknas - Inställningar (Essential, Marketing, Analytics) sparas lokalt - Marketing-banner visas enbart vid godkänt samtycke - Går att ändra via Topbar / Sidebar efter inloggning

Slutsats Projektet Ventixe är ett komplett eventhanteringssystem i modern molnarkitektur, byggt med säkerhet, skalbarhet och separation av ansvar i åtanke. Arkitekturen är SOA-baserad med tydliga API:er, centraliserad identitetshantering och optimerad frontend UX.