

IT-personell Veiledning for Postly

Dokument Betegnelse

Innledning

- **Formål:** Denne veiledningen er laget for å hjelpe IT-personell med tekniske aspekter ved Postly-plattformen.
- **Intro:** En oversikt over hva dokumentet dekker og hvordan det skal brukes.
- **Målgruppe og forutsetninger:** Dokumentet er rettet mot IT-ansatte med grunnleggende kjennskap til webutvikling og serveradministrasjon.

Teknisk

- **Oversikt:** En detaljert oversikt over de teknologiene og verktøyene som brukes i Postly.
 - **Webserver:** Beskrivelse av servermiljøet som brukes for å kjøre Postly.
 - **C# (backend):** Informasjon om backend-utviklingen ved bruk av C#.
 - **Typescript og frontend miljø:** Hvordan Typescript brukes for frontend-utviklingen.
 - **MongoDB:** Databaseadministrasjon ved bruk av MongoDB.
 - **Azure:** Bruk av Azure for skytjenester og deployering.
- **Kommunikasjon og Samhandling:** Hvordan de ulike komponentene kommuniserer og samhandler.
- **Ressurser:** Eksterne ressurser og dokumentasjon som kan være nyttige.

Backend

- **Arkitektur og Beskrivelse:** Detaljert beskrivelse av backend-arkitekturen.
- **Hvordan clone fra git og bygge i dev miljø:** Steg-for-steg veiledning for å klonе prosjektet fra Git og sette opp utviklingsmiljøet.
- **Eventuelle packages som må lastes ned:** Liste over nødvendige pakker og avhengigheter som må installeres.

Frontend

- **Arkitektur og Beskrivelse:** Detaljert beskrivelse av frontend-arkitekturen.

- **Hvordan clone fra git og kjøre i dev miljø:** Steg-for-steg veiledning for å klonе frontend-prosjektet fra Git og kjøre det i utviklingsmiljøet.
- **Eventuelle vanskeligheter:** Kjente utfordringer og hvordan man løser dem.

Serverkonfigurasjon

- **Hvordan koble til serveren gjennom SSH:** Steg-for-steg veiledning for å koble til serveren via SSH.
- **Ytterligere konfigurasjon:**
 - **Oppsett av miljøvariabler**
 - **Konfigurasjon av serveren for produksjonsmiljø**
 - **Sikkerhetsinnstillinger**
 - **Loggadministrasjon**

Databaseadministrasjon

- **Database ressurser, se ressurser høyere i dokumentet:** Referanse til relevante ressurser.
- **Installasjon av MongoDB Compass:** Hvordan installere MongoDB Compass for enklere databaseadministrasjon.
- **Hvordan koble til:** Instruksjoner for å koble til MongoDB-databasen.
- **Databaseadministrasjonsverktøy:** Beskrivelse av verktøy og metoder for effektiv databaseadministrasjon.

Deploying og Oppdatering

- **Intro:** Generell informasjon om deployering og oppdatering av Postly.
- **Hvordan deploye siden, konfigurere frontend og backend:** Steg-for-steg veiledning for deployering av Postly, inkludert konfigurasjon av frontend og backend.
- **Hvordan oppdatere koden gjennom git:** Instruksjoner for å oppdatere prosjektkoden via Git, inkludert håndtering av merges og konflikter.

Detaljert Veiledning

Innledning

Formål

Denne veiledningen er ment for IT-personell som jobber med Postly, og dekker tekniske detaljer, konfigurasjon, og administrasjon av plattformen.

Intro

Dokumentet gir en oversikt over systemets arkitektur, teknologier, og beste praksis for vedlikehold og videreutvikling.

Målgruppe og Forutsetninger

Målgruppen er IT-ansatte med kunnskap om webutvikling, serveradministrasjon, og bruk av versjonskontrollsystemer som Git.

Teknisk

Oversikt

Webserver

Postly kjører på en webserver konfigurert for høy ytelse og sikkerhet. Typiske innstillinger inkluderer SSL-sertifikater, lastbalansering, og sikkerhetsforsterkninger.

C# (backend)

Backend er utviklet i C#, og benytter ASP.NET Core for å bygge og vedlikeholde API-er og serverfunksjonalitet.

Typescript og frontend miljø

Frontend er utviklet med Typescript og moderne rammeverk som React. Dette gir sterke og skalerbare brukergrensesnitt.

MongoDB

MongoDB brukes som database for å lagre brukerdata, innlegg, og annen viktig informasjon. Det er konfigurert for å sikre høy tilgjengelighet og ytelse.

Azure

Azure brukes for skytjenester, inkludert hosting og kontinuerlig integrasjon/deployering (CI/CD).

Kommunikasjon og Samhandling

Postly benytter API-er for kommunikasjon mellom frontend, backend og databasen. Datautveksling skjer over HTTP.

Ressurser

Nyttige ressurser inkluderer dokumentasjon for ASP.NET Core, React, MongoDB, og Azure-tjenester. Alle passord og praktiske ressurser finner man i Excel arket som er vedlagt.

Backend

Arkitektur og Beskrivelse

Backend-arkitekturen er modulær, med klare skiller mellom forretningslogikk, datatilgangslag, og API-kontrollere. Dette sikrer lett vedlikehold og utvid barhet. Postly bruker swagger i dev miljøet for å teste endepunkter og funksjonaliteten til API'en.

Hvordan clone fra git og bygge i dev miljø

1. **Clone repository:** git clone <https://github.com/fla1ly/PostlyAPI.git>
2. **Naviger til prosjektmappe**
3. **Installer nødvendige pakker:** dotnet restore
4. **Bygg prosjektet:** dotnet build eller trykke på grønne run knappen i visual studio.

Eventuelle packages som må lastes ned

- ASP.NET Core
- Entity Framework Core
- MongoDB.driver

- Swashbuckle.AspNetCore

Frontend

Arkitektur og Beskrivelse

Frontend benytter komponentbasert arkitektur via React Vite. Dette gir en modulær tilnærming hvor hver komponent håndterer spesifikke funksjoner. Postly bruker Material UI som sin styling library. Grunnen er enkel bruk og effektiv utvikling i front end.

Hvordan clone fra git og kjøre i dev miljø

1. **Clone repository:** `git clone https://github.com/fla1ly/Postly.git`
2. **Naviger til prosjektmappe:** `cd Postly`
3. **Installer nødvendige pakker:** `npm install`
4. **Start utviklingsserver:** `npm run dev`

Eventuelle vanskeligheter

Vanlige utfordringer inkluderer:

- Versjonskonflikter med npm-pakker
- Problemer med CORS når du kjører frontend og backend på forskjellige domener
- Spør meg direkte hvis det oppstår noen problemer eller vanskeligheter

Serverkonfigurasjon

Hvordan koble til serveren gjennom SSH

1. **Åpne terminal.**
2. **Koble til serveren:** `ssh flaily@4.235.113.46`
3. **Skriv inn passord.** (passord ligger i excel dokumentet)

Ytterligere konfigurasjon

- **Konfigurasjon av serveren for produksjonsmiljø:** Optimaliser serverinnstillinger for produksjonsbruk.
- **Sikkerhetsinnstillinger:** Implementer brannmurregler, sikre SSH-tilgang, og sett opp automatiserte sikkerhetsoppdateringer.
- **Loggadministrasjon:** Konfigurer loggrotasjon og overvåking.

Databaseadministrasjon

Database ressurser

Se tidligere nevnte ressurser for informasjon om MongoDB.

Installasjon av MongoDB Compass

1. Last ned MongoDB Compass fra [MongoDBs offisielle nettsted](#).
2. Følg installasjonsveiledningen for ditt operativsystem.

Hvordan koble til

1. Åpne MongoDB Compass.
2. Skriv inn database-URL (ligger i excel arket).
3. Klikk på "Connect".
4. Hvis du ikke får tilgang, send meg en beskjed med din ip, sånn at jeg kan whiteliste ip'en din.

Databaseadministrasjonsverktøy

- **MongoDB Compass:** For administrasjon av MongoDB i skyen.
- **Robo 3T:** Et annet populært verktøy for MongoDB-administrasjon.

Deployering og Oppdatering

Intro

Denne delen dekker hvordan man deployerer Postly-plattformen og holder den oppdatert.

Hvordan deploye siden, konfigurere frontend og backend

1. **Backend Deploy:**
 - Bygg prosjektet: `dotnet publish -c Release`
 - Overfør bygget til serveren: `scp -r ./publish flaily@4.235.113.46:/path/to/deploy`
 - Start backend-tjenesten (f.eks. via systemd).
2. **Frontend Deploy:**
 - Bygg frontend: `npm run build`

- Overfør bygget til serveren: `scp -r ./build flaily@4.235.113.46:/path/to/deploy`
- Konfigurer webserveren (f.eks. Nginx) til å servere frontend-bygg.

Veiledning for hosting av Vite og C#

Denne veiledningen tar deg gjennom trinnene for å hoste en Vite frontend og en C# backend.

1. Oppdatere systemet

```
sudo apt update  
sudo apt upgrade -y
```

2. Installere nødvendige pakker

```
sudo apt install -y curl gnupg
```

3. Installere Node.js og NPM

Last ned og installer Node.js, som er kompatibel med Vite.

```
curl -fsSL https://deb.nodesource.com/setup_16.x | sudo -E bash -  
sudo apt install -y nodejs
```

4. Installere .NET SDK

Last ned og installer .NET SDK for å kjøre C# backend.

```
wget https://packages.microsoft.com/config/debian/10/packages-microsoft-prod.deb -O packages-microsoft-prod.deb  
sudo dpkg -i packages-microsoft-prod.deb  
sudo apt update  
sudo apt install -y apt-transport-https  
sudo apt update  
sudo apt install -y dotnet-sdk-7.0
```

5. Kodekloning fra GitHub

Klone frontend og backend fra GitHub til din VM. Erstatt `your-frontend-repo-url` og `your-backend-repo-url` med de faktiske URLene til dine repositorier.

```
cd ~
git clone https://github.com/Fla1ly/Postly
git clone https://github.com/Fla1ly/PostlyAPI
```

6. Bygge og distribuere Vite Frontend

Naviger til frontend-mappen, installer avhengigheter og bygg prosjektet.

```
cd ~/frontend
npm install
npm run build
```

Kopier byggefilene til en offentlig tilgjengelig mappe. Vi bruker `/var/www/html`

```
sudo mkdir -p /var/www/html
sudo cp -r dist/* /var/www/html
```

7. Installere og konfigurere Nginx

Nginx brukes som en webserver for å servere frontend-filer.

```
sudo apt install -y nginx
```

Konfigurer Nginx for å servere Vite frontend. Rediger Nginx konfigurasjonsfilen.

```
sudo nano /etc/nginx/sites-available/default
```

Erstatt innholdet med følgende (tilpass root-dir hvis nødvendig):

```
nginx
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    root /var/www/html;
    index index.html;

    server_name _;

    location / {
        try_files $uri $uri/ /index.html;
    }
}
```


Start Nginx på nytt for at endringene skal tre i kraft.

```
sudo systemctl restart nginx
```

8. Bygge og distribuere C# Backend

Naviger til backend-mappen og bygg prosjektet.

```
cd ~/backend  
dotnet publish -c Release -o out
```

9. Konfigurere systemd Service for C# Backend

Lag en systemd tjenestefil for å kjøre backend som en tjeneste.

```
sudo nano /etc/systemd/system/kestrel-postly.service
```

Legg til følgende innhold:

```
ini  
[Unit]  
Description=Postly .NET Web API  
[Service]  
WorkingDirectory=/home/ditt-navn /backend/out  
ExecStart=/usr/bin/dotnet /home/ditt-navn/backend/out/BackendApp.dll  
Restart=always  
# Restart service after 10 seconds if the dotnet service crashes:  
RestartSec=10  
KillSignal=SIGINT  
SyslogIdentifier=dotnet-postly  
User=www-data  
Environment=ASPNETCORE_ENVIRONMENT=Production  
[Install]  
WantedBy=multi-user.target
```

Erstatt ditt-navn og BackendApp.dll med de riktige verdiene.

Start og aktiver tjenesten.

```
sudo systemctl enable kestrel-postly.service  
sudo systemctl start kestrel-postly.service
```

10. Konfigurere Nginx som en reverse proxy for backend

Rediger Nginx konfigurasjonsfilen for å inkludere proxy-innstillinger.

```
bash
sudo nano /etc/nginx/sites-available/default
```

Legg til følgende før den siste }:

```
nginx
location /api {
    proxy_pass http://localhost:5000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection keep-alive;
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
}
```

Start Nginx på nytt for at endringene skal tre i kraft.

```
sudo systemctl restart nginx
```

11. Teste applikasjonen

Besøk VM's IP-adresse i nettleseren for å bekrefte at frontend er tilgjengelig. Test API-endepunktene for å sikre at backend fungerer som forventet.

Hvordan oppdatere koden gjennom git

1. Pull oppdateringer fra Git:

- Naviger til prosjektets mappe: `cd /path/to/project`
- Pull siste endringer: `git pull origin main`

2. Håndtering av merges og konflikter:

- Merge endringer lokalt: `git merge branch-name`
- Løs eventuelle konflikter manuelt.

3. Bygg og deploy på nytt:

- Følg de relevante trinnene for å bygge og deployere frontend og backend.

Trenger du mer info eller hjelp?

Kontakt meg direkte.