

Valg av Tjenester for Dasigpt

Denne dokumentasjonen beskriver hvorfor vi har valgt spesifikke tjenester for Dasigpt, hvordan de er satt opp, og hvordan de bidrar til å sikre en trygg og effektiv brukeropplevelse, og tilgjengelighet.

Innholdsfortegnelse

1. [Firebase](#)
 - [Oppsett](#)
 - [Bruk](#)
 - [Eksempel på bruk](#)
 - [Forklaring](#)
 - [Begrunnelse](#)
 2. [Stripe](#)
 - [Oppsett](#)
 - [Bruk](#)
 - [Eksempel på bruk](#)
 - [Forklaring](#)
 - [Begrunnelse](#)
 3. [Google Sign-In](#)
 - [Oppsett](#)
 - [Bruk](#)
 - [Eksempel på bruk](#)
 - [Forklaring](#)
 - [Begrunnelse](#)
 4. [Cloudflare](#)
 - [Oppsett](#)
 - [Bruk](#)
 - [Begrunnelse](#)
 5. [Konklusjon](#)
-

Firebase

Oppsett

Vi har satt opp en real-time database ved hjelp av Firebase. Hvor vi kan lagre informasjon vi måtte trenge om brukere, chater og annen eventuel data vi måtte trenge.

Bruk

Firebase brukes til å:

- Lagre meldingsdata i sanntid.
- Sikre rask og pålitelig distribusjon av applikasjonen.

Eksempel på bruk

I backend bruker vi Firebase til å skrive for eksempel brukerdata til databasen. Her er et eksempel på hvordan dette er implementert:

```
import { initializeApp, getApps } from "firebase/app";
import { getDatabase, ref, set } from "firebase/database";
import { env } from "$env/dynamic/private"; // Importerer miljøvariabler
fra en privat fil

// Initialiserer Firebase-konfigurasjonen
const firebaseConfig = {
  apiKey: env.FB_API_KEY,
  authDomain: env.FB_AUTH_DOMAIN,
  databaseURL:
    "https://dasi-gpt-b2beb-default-rtdb.europe-
west1.firebaseio.com",
  projectId: env.FB_PROJECT_ID,
  storageBucket: env.FB_STORAGE_BUCKET,
  messagingSenderId: env.FB_MESSAGING_SENDER_ID,
  appId: env.FB_APP_ID,
  measurementId: env.FB_MEASUREMENT_ID,
};

// Initialiserer Firebase-appen
const app = initializeApp(firebaseConfig);

// Funksjon som skriver brukerdata til Firebase-databasen
function writeUserData(userID, name, email, imageUrl) {
  const db = getDatabase();
  return set(ref(db, "users/" + userID), {
    username: name,
    email: email,
    profile_picture: imageUrl,
  });
}

// Funksjon som håndterer POST-forespørsler
export async function POST({ request }) {
  try {
    // Henter JSON data fra forespørselen
    const { userID, name, email, imageUrl } = await request.json();

    // Skriver brukerdata til Firebase-databasen
    await writeUserData(userID, name, email, imageUrl);

    // Returnerer en suksessrespons
    return new Response(
      JSON.stringify({ message: "User data written successfully" }),
      {
        status: 200, // HTTP-statuskode for suksess
        headers: {
```

```
        "Content-Type": "application/json", // Angir at vi
sender JSON-data
    },
    }
    );
} catch (error) {
    // Logger feilen for debugging
    console.error("Error:", error);
    // Returnerer en feilmelding som en JSON-respons
    return new Response(
        JSON.stringify({ error: "Internal Server Error" }),
        {
            status: 500, // HTTP-statuskode for intern serverfeil
            headers: {
                "Content-Type": "application/json", // Angir at vi
sender JSON-data
            },
        }
    );
}
}
```

Forklaring

- **Firebase-konfigurasjon:** Firebase-appen initialiseres med konfigurasjonsdata fra miljøvariabler for å sikre sikkerhet og fleksibilitet.
- **Lagring av brukerdata:** Funksjonen `writeUserData` lagrer brukerdata som navn, e-post og profilbilde i Firebase Realtime Database.
- **POST-forespørsel:** Backend håndterer en POST-forespørsel som inneholder brukerdata i JSON-format og lagrer disse i databasen.
- **Feilhåndtering:** Eventuelle feil logges, og en passende feilmelding returneres til klienten.

Begrunnelse

- **Skalerbarhet:** Firebase er en skalerbar løsning som passer for både små og store applikasjoner.
- **Sikkerhet:** Det tilbyr innebygd sikkerhet og GDPR-kompatibilitet.
- **Effektivitet:** Firebase reduserer utviklingstid med ferdige løsninger for vanlige backend-behov.

Stripe

Oppsett

Stripe er integrert som vår betalingsløsning for å håndtere transaksjoner på en sikker måte.

Bruk

Stripe brukes til å:

- Opprette abonnementer for brukere.
- Sikre at alle transaksjoner er i samsvar med GDPR og andre relevante lover.

- Håndtere betalinger med støtte for flere valutaer og betalingsmetoder.

Eksempel på bruk

I backend bruker vi Stripe til å opprette en utsjekkingsøkt basert på handlekurvdata fra brukeren. Her er et eksempel på hvordan dette er implementert:

```
import { stripe } from "../stripe"; // Importerer Stripe-biblioteket
import type { RequestHandler } from "@sveltejs/kit"; // Importerer
RequestHandler-typen fra SvelteKit
import type { CartItem } from "../api/cart"; // Importerer CartItem-typen
fra cart API
import { env } from "$env/dynamic/private"; // Importerer miljøvariabler

export const POST: RequestHandler = async ({ request }) => {
  const data = await request.json(); // Henter JSON-data fra forespørselen
  const cartItems: CartItem[] = data.items; // Henter handlekurvdata

  // Oppretter linjeelementer for Stripe-utsjekkingsøkt
  const lineItems = cartItems.map((item) => {
    return {
      price_data: {
        currency: "USD",
        product_data: {
          name: item.name,
          images: [],
        },
      },
      unit_amount: item.price * 100, // Pris i cent
      recurring: {
        interval: "month", // Abonnement månedlig
      },
    },
    quantity: item.amount,
  });

  // Oppretter en Stripe-utsjekkingsøkt
  const session = await stripe.checkout.sessions.create({
    line_items: lineItems,
    shipping_address_collection: {
      allowed_countries: ["NO"], // Tillater frakt til Norge
    },
    mode: "subscription",
    success_url: [`${env.BASE}/success`]
  });
  (command: _github.copilot.openSymbolFromReferences?
  %5B%22%24%7Benv.BASE%7D%2Fsuccess%22%2C%5B%7B%22uri%22%3A%7B%22%24mid%22%3
  A1%2C%22fsPath%22%3A%22%2FUsers%2Fsimenevenrudblien%2FRepositories%2FDASI-
  gpt%2FDASI%2FDasi%2Fsrc%2Froutes%2Fcheckout%2F%2Bserver.ts%22%2C%22externa
  l%22%3A%22file%3A%2F%2F%2FUsers%2Fsimenevenrudblien%2FRepositories%2FDASI-
  gpt%2FDASI%2FDasi%2Fsrc%2Froutes%2Fcheckout%2F%252Bserver.ts%22%2C%22path%
  22%3A%22%2FUsers%2Fsimenevenrudblien%2FRepositories%2FDASI-
  gpt%2FDASI%2FDasi%2Fsrc%2Froutes%2Fcheckout%2F%2Bserver.ts%22%2C%22scheme%
```

```

22%3A%22file%22%7D%2C%22pos%22%3A%7B%22line%22%3A35%2C%22character%22%3A18
%7D%7D%5D%5D "Go to definition"), // URL for vellykket betaling
    cancel_url: [`${env.BASE}/cancel`]
(command:_github.copilot.openSymbolFromReferences?
%5B%22%24%7Benv.BASE%7D%2Fcancel%22%2C%5B%7B%22uri%22%3A%7B%22%24mid%22%3A
1%2C%22fsPath%22%3A%22%2FUsers%2Fsimenevenrudblien%2FRepositories%2FDASI-
gpt%2FDASI%2FDasi%2Fsrc%2Froutes%2Fcheckout%2F%2Bserver.ts%22%2C%22externa
l%22%3A%22file%3A%2F%2F%2FUsers%2Fsimenevenrudblien%2FRepositories%2FDASI-
gpt%2FDASI%2FDasi%2Fsrc%2Froutes%2Fcheckout%2F%252Bserver.ts%22%2C%22path%
22%3A%22%2FUsers%2Fsimenevenrudblien%2FRepositories%2FDASI-
gpt%2FDASI%2FDasi%2Fsrc%2Froutes%2Fcheckout%2F%2Bserver.ts%22%2C%22scheme%
22%3A%22file%22%7D%2C%22pos%22%3A%7B%22line%22%3A35%2C%22character%22%3A18
%7D%7D%5D%5D "Go to definition"), // URL for avbrutt betaling
    phone_number_collection: {
        enabled: true, // Aktiverer innsamling av telefonnummer
    },
});

return new Response(JSON.stringify({ url: session.url }), {
    status: 200,
    headers: { "Content-Type": "application/json" },
});
};

```

Forklaring

- **Handlekurvdata:** Backend mottar handlekurvdata fra frontend som JSON.
- **Linjeelementer:** Dataene mapper til Stripe's `line_items`-format, inkludert produktnavn, pris og mengde.
- **Utsjekkingsøkt:** En Stripe-utsjekkingsøkt opprettes med abonnement som betalingsmodus.
- **URL-er:** Brukeren omdirigeres til en suksess- eller avbrutt-side basert på resultatet av betalingen.

Begrunnelse for bruk av Stripe

Vi valgte Stripe som betalingsløsning fordi:

- **Sikkerhet:** Stripe tilbyr avanserte sikkerhetsfunksjoner som kryptering, tokenisering og PCI DSS-sertifisering, noe som sikrer at sensitive betalingsdata håndteres trygt.
- **GDPR-samsvar:** Stripe hjelper oss med å oppfylle kravene i GDPR ved å sikre at brukerdata behandles i samsvar med europeiske personvernlover.
- **Fleksibilitet:** Stripe støtter flere betalingsmetoder og valutaer, noe som gjør det enkelt å tilpasse løsningen til ulike markeder.
- **Pålitelighet:** Stripe er en velprøvd løsning som brukes av mange store selskaper, noe som gir oss trygghet i at systemet er robust og skalerbart.
- **Enkel integrasjon:** Stripe tilbyr et brukervennlig API og omfattende dokumentasjon, noe som gjør det raskt og enkelt å integrere i vår applikasjon.
- **Abonnementsstøtte:** Stripe gjør det enkelt å håndtere abonnementer, noe som er en viktig del av vår forretningsmodell.

Google Sign-In

Oppsett

Google Sign-In er integrert for å gi brukerne en enkel og sikker måte å logge inn på applikasjonen ved hjelp av deres Google-konto.

Bruk

Google Sign-In brukes til å:

- Autentisere brukere ved hjelp av deres Google-konto.
- Hente brukerdata som navn, e-post og profilbilde for å personalisere opplevelsen.
- Sikre en rask og brukervennlig innloggingsprosess.

Eksempel på bruk

Her er et eksempel på hvordan Google Sign-In er implementert i applikasjonen:

```
<script lang="ts">
  import { onMount } from 'svelte';
  import { user } from "../../store/userStore";

  declare const google: any;

  onMount(() => {
    // Last inn Google Identity Services (GIS)-biblioteket
    const script = document.createElement('script');
    script.src = 'https://accounts.google.com/gsi/client';
    script.async = true;
    script.defer = true;
    document.body.appendChild(script);

    script.onload = () => {
      // Initialiser Google Identity Services
      google.accounts.id.initialize({
        client_id: 'DIN_KLIENT_ID_HER', // Erstatt med din klient-ID
        callback: onSignIn,
      });
      renderButton();
    };
  });

  function renderButton() {
    // Opprett en Sign-In-knapp
    google.accounts.id.renderButton(
      document.getElementById('signIn'), // Elementet hvor knappen skal vises
      { theme: 'outline', size: 'large' } // Tilpasningsalternativer
    );
  }
}
```

```
function onSignIn(response) {
  const id_token = response.credential;
  // Dekoder JWT-tokenet for å hente brukerdata
  const base64UrlPayload = id_token.split('.')[1];
  const base64Payload = base64UrlPayload.replace(/-/g,
'+').replace(/_/g, '/');
  const payload =
JSON.parse(decodeURIComponent(atob(base64Payload).split('').map(function(c
) {
    return '%' + ('00' + c.charCodeAt(0).toString(16)).slice(-2);
}).join(''))));

  // Oppdaterer Svelte store med brukerinformasjon
  user.set({
    id: payload.sub,
    name: payload.name,
    email: payload.email,
    picture: payload.picture,
    method: 'google'
  });
  console.log('Logged in as: ' + payload.name);
}

function signOut(event: Event) {
  event.preventDefault();
  google.accounts.id.disableAutoSelect();
  user.set({
    id: null,
    name: null,
    email: null,
    picture: null,
    method: null
  });
  location.reload();
  console.log('User signed out.');
```

}

</script>

<div id="signIn"></div>

Forklaring

- **Google Identity Services (GIS):** GIS-biblioteket lastes inn dynamisk og initialiseres med en klient-ID.
- **Sign-In-knapp:** En Google Sign-In-knapp opprettes og rendres i brukergrensesnittet.
- **LocalStorage:** Brukerens innloggingsstatus lagres i LocalStorage for å unngå at brukeren må logge inn på nytt ved for eksempel en sideoppdatering.
- **Svelte Store:** Brukerdata lagres i en Svelte store for å gjøre det enkelt å bruke dataene i hele applikasjonen.
- **Logg ut:** En funksjon for å logge ut brukeren er implementert, som også tilbakestiller brukerdata i Svelte store.

Begrunnelse for bruk av Google Sign-In

- **Brukervennlighet:** Google Sign-In gir brukerne en rask og enkel måte å logge inn på uten å måtte opprette en ny konto.
 - **Sikkerhet:** Google tilbyr en sikker autentiseringsprosess med støtte for OAuth 2.0 og kryptering av brukerdata.
 - **Tillitsverdighet:** Mange brukere har allerede en Google-konto, noe som gjør det til en pålitelig og kjent løsning for autentisering.
 - **GDPR-samsvar:** Google Sign-In hjelper oss med å oppfylle kravene i GDPR ved å sikre at brukerdata behandles i samsvar med europeiske personvernlover.
 - **Effektiv datainnsamling:** Google Sign-In gir oss tilgang til nødvendige brukerdata som navn, e-post og profilbilde, noe som gjør det enkelt å personalisere brukeropplevelsen.
 - **Redusert friksjon:** Ved å bruke Google Sign-In slipper brukerne å huske enda et passord, noe som reduserer friksjon og forbedrer brukeropplevelsen.
-

Cloudflare

Oppsett

Vi bruker Cloudflare for å hoste domenet vårt (dasigpt.com). Dette gjør Cloudflare ved å lage en redirect-url som automatisk redirigerer deg til våres sider. Dette gjøres mulig ved kjøp av domene hos [Cloudflare](#).

Bruk

Cloudflare brukes til å:

- Hoste domenet vårt.
- Filtrere ut skadelig trafikk automatisk.
- Beskytte applikasjonen mot DDoS-angrep.
- Forbedre ytelsen ved å cache statiske ressurser.

Begrunnelse

Vi valgte Cloudflare fordi:

- **Risikoanalyse:** Risikoanalysen vår viste at DDoS-angrep og skadelig trafikk er en betydelig trussel.
 - **Under Attack Mode:** Cloudflare er konfigurert med "Under Attack Mode" for å beskytte sidene våre mot skadelig trafikk.
 - **Sikkerhetsbarriere:** Det gir en ekstra sikkerhetsbarriere uten å påvirke brukeropplevelsen.
 - **Kostnadseffektivitet:** Det er en kostnadseffektiv løsning for å beskytte applikasjonen.
 - **Trafikkkontroll:** Det gir god oversikt og kontroll over trafikk, og alt man måtte trenge å holde oversikt over på domenet.
-

Konklusjon

Ved å kombinere Firebase, Stripe, Google Sign-In og Cloudflare har vi bygget en robust, sikker og brukervennlig plattform for Dasigpt. Hver tjeneste er nøye valgt basert på spesifikke behov og risikoanalyser:

- **Firebase** gir oss en skalerbar og effektiv backend-løsning for sanntidsdatabehandling og sikker lagring av brukerdata.
- **Stripe** sikrer en trygg og pålitelig betalingsprosess som er i samsvar med GDPR og andre relevante lover, samtidig som det gir fleksibilitet for ulike betalingsmetoder og abonnementshåndtering.
- **Google Sign-In** forenkler autentisering for brukerne, reduserer friksjon og gir en sikker og tillitsvekkende innloggingsopplevelse.
- **Cloudflare** beskytter applikasjonen mot skadelig trafikk og DDoS-angrep, samtidig som det forbedrer ytelsen og sikrer høy tilgjengelighet.

Ved å bruke disse tjenestene har vi sikret at Dasigpt er:

- **Sikker:** Brukerdata og betalinger håndteres med høyeste standard for sikkerhet.
- **Effektiv:** Plattformen er optimalisert for rask ytelse og enkel bruk.
- **Skalerbar:** Løsningene kan vokse med applikasjonen og tilpasse seg økende krav.
- **Brukervennlig:** Brukerne får en sømløs opplevelse, fra innlogging til betaling.

Disse valgene reflekterer vår forpliktelse til å levere en trygg, pålitelig og brukervennlig tjeneste for våre brukere.
