

AI leerlijn

Supabase Auth & RLS

Les 10

PLANNING VANDAAG

AGENDA

09:00 Welkom + terugblik

09:10 Eindexamenopdracht

09:25 Theorie: Authenticatie

09:45 Auth in Next.js + RLS

10:00 PAUZE

10:15 Hands-on: Auth opzetten

10:30 Login & Registratie bouwen

10:55 Sessie & beschermde routes

11:10 Basis RLS

11:45 Afsluiting

LEERDOELEN

1. Auth vs Autorisatie begrijpen
2. Supabase Auth 3 methodes kennen
3. @supabase/ssr configureren
4. Login/registratie pagina bouwen
5. RLS policies schrijven

TERUGBLIK LES 8-9

We hebben al een werkende Poll App met Supabase.

WAT WE HEBBEN

- ✓ Supabase project aangemaakt
- ✓ Database: polls + options tabellen
- ✓ Next.js app gekoppeld
- ✓ CRUD werkend

→ **Nu: Authenticatie toevoegen!**

DATABASE STRUCTUUR

polls

id, question, created_at



options

id, text, votes, poll_id

EINDEXAMENOPDRACHT

Je gaat een eigen app bouwen met alles wat je hebt geleerd.

VRIJE KEUZE APP

Kies je eigen app-idee.

Denk aan: todo-app, blog, quiz, receipt-app, e-commerce, etc.

Je past alles toe wat je hebt geleerd in les 1-14.

VEREISTEN

- ✓ Next.js + TypeScript
- ✓ Supabase (database + auth)
- ✓ CRUD operaties
- ✓ RLS policies
- ✓ Deployment
- ✓ Clean code + Git

Les 10-14: bouwen | Les 15-16: afronden | Les 17: presentatie

WAT IS AUTHENTICATIE?

AUTHENTICATIE

"Wie ben je?"

- Email + wachtwoord
- Biometrics (vingerafdruk)
- Magic link
- Google account

AUTORISATIE

"Wat mag je?"

- Alleen eigen data zien
- Admin vs gebruiker
- RLS policies
- Rollen en permissions

Vandaag: eerst authenticatie opzetten, daarna autorisatie met RLS

SUPABASE AUTH — 3 METHODES

Supabase biedt meerdere auth providers. Wij gebruiken deze drie:

EMAIL / PASSWORD

"Klassiek" — user maakt account

Wachtwoord gehasht opgeslagen

Bevestigingsmail (optioneel)

MAGIC LINK

"Passwordless" — geen wachtwoord

Email met login link

Link verloopt na 1 uur

GOOGLE OAUTH

"Third-party" — via Google

Gebruiker logt in met Google

Makkelijkst voor gebruikers

Tim laat elke methode LIVE zien in het Supabase Dashboard

HOE WERKT EEN SESSIE?

- 1 LOGIN** User stuurt email + wachtwoord
- 2 VERIFICATIE** Supabase checkt credentials
- 3 JWT TOKEN** Supabase maakt een token aan
- 4 COOKIE** Token opgeslagen in browser cookie
- 5 ELKE REQUEST** Cookie automatisch meegestuurd
- 6 MIDDLEWARE** Refresht de sessie token

WAT IS EEN JWT?

- Header (algoritme)
- Payload (user_id, email, exp)
- Signature (bewijs van Supabase)

AUTH IN NEXT.JS

@supabase/ssr — de brug tussen Supabase en Next.js

BROWSER CLIENT

`createBrowserClient()`

Voor client components ("use client")

`src/lib/supabase/client.ts`

SERVER CLIENT

`createServerClient()`

Voor server components + API routes

`src/lib/supabase/server.ts`

MIDDLEWARE

Draait voor elke request

Refresht de sessie

Redirect naar /login

`src/middleware.ts`

AUTH CALLBACK

Vangt magic link / OAuth op

Wisselt code voor sessie

`src/app/auth/callback/route.ts`

ROW LEVEL SECURITY

ZONDER RLS

- Iedereen kan ALLES lezen
- Iedereen kan ALLES schrijven
- Iedereen kan ALLES verwijderen

ONVEILIG!

MET RLS

- Database checkt ELKE query
- Policies bepalen toegang
- `auth.uid()` = de ingelogde user

VEILIG

```
CREATE POLICY "Poʌs lezen"  
ON poʌs FOR SELECT  
USING (true); -- iedereen mag lezen
```

Pauze

15 minuten — tot 10:15

HANDS-ON: AUTH OPZETTEN

1. Pak poll-app-starter.zip uit

2. `npm install`

3. Maak `.env.local`:

`NEXT_PUBLIC_SUPABASE_URL= ...`

`NEXT_PUBLIC_SUPABASE_ANON_KEY= ...`

4. `npm install @supabase/ssr`

`@supabase/supabase-js`

5. Maak `src/lib/supabase/client.ts`

6. Maak `src/lib/supabase/server.ts`

7. Maak `src/middleware.ts`

8. Maak `src/app/auth/callback/route.ts`

9. Supabase Dashboard:

Auth → Providers → Emailaan

HANDS-ON: LOGIN & REGISTRATIE

1. Maak `src/app/login/page.tsx`

2. "use client" + `import` ports

3. `State: email, password, message, loading`

4. `handleSignUp:`
`supabase.auth.signUp({`
`email, password`
`})`

5. `handleSignIn:`
`supabase.auth.signInWith`
`Password({ email, password })`

6. `handleMagicLink:`
`supabase.auth.signInWith`
`({ email })`

7. `Formulier: email + password`
`inputs, 3 buttons`

8. `Test: registreer → check`
`Supabase dashboard → login`

HANDS-ON: SESSIE & ROUTES

1. Update Navbar component

2. Haal user op:

```
const { data: { user } } =  
  await supabase.auth.getUser()
```

3. Toon email + logout button
als ingelogd

4. handleSignOut:

```
await supabase.auth.signOut()  
router.push('/login')
```

5. Test: naar /zonder login
→ redirect naar /login

6. Test: na login → email
zichtbaar in navbar

HANDS-ON: ROW LEVEL SECURITY

-- 1. Enable RLS

```
ALTER TABLE polls
    ENABLE ROW LEVEL SECURITY;
ALTER TABLE options
    ENABLE ROW LEVEL SECURITY;
```

-- 2. Polls lezen (iedereen)

```
CREATE POLICY "Polls lezen"
ON polls FOR SELECT
USING (true);
```

-- 3. Polls maken (ingelogd)

```
CREATE POLICY "Polls maken"
ON polls FOR INSERT
TO authenticated
WITH CHECK (true);
```

-- 4. Options lezen (iedereen)

```
CREATE POLICY "Options lezen"
ON options FOR SELECT
USING (true);
```

-- 5. Stemmen (ingelogd)

```
CREATE POLICY "Stemmen"
ON options FOR UPDATE
TO authenticated
USING (true);
```

Huiswerk

Zie: Les10-Huiswerk.m d

- Google OAuth toevoegen
- user_id kolom + RLS uitbreiden
- Eindexamenopdracht brainstorm

Vragen? Stel ze nu of via Teams.