

Admin Access & Role-Based Routing

Overzicht

Dit document beschrijft hoe admin access en role-based routing werkt in WritGo.nl.

Admin Users

Admin gebruikers hebben toegang tot beide portals:

- **Admin Portal:** `/admin/*` routes voor beheer en administratie
- **Client Portal:** `/client/*` routes voor het testen van de client ervaring

Admin Email Adressen

De volgende email adressen krijgen automatisch admin rechten:

- `info@writgoai.nl`
- `info@writgo.nl`

Deze lijst is gedefinieerd in `lib/auth-options.ts` (regel 8).

Authentication Flow

1. Login Process

Wanneer een gebruiker inlogt via `/inloggen`:

1. NextAuth verifieert de credentials tegen de database (User of Client tabel)
2. Als het email adres in de `adminEmails` lijst staat, krijgt de gebruiker de `admin` role
3. Na succesvolle login wordt de gebruiker geredirect op basis van hun role:

- **Admin/Superadmin:** → `/admin/dashboard`
- **Client:** → `/client/overzicht`

2. Role-Based Access Control (Middleware)

De middleware (`middleware.ts`) controleert bij elke request:

- **Admin Routes** (`/admin/*`): Alleen toegankelijk voor users met role `admin` of `superadmin`
- **Client Routes** (`/client/**`): Toegankelijk voor iedereen (clients én admins)
- **Legacy Routes** (`/dashboard`, `/client-portal`): Worden automatisch geredirect naar de nieuwe structuur

3. Automatic Redirects

- Admins die naar `/dashboard` navigeren → automatisch naar `/admin/dashboard`
- Clients die naar `/dashboard` navigeren → automatisch naar `/client/overzicht`
- Niet-admins die naar `/admin/*` proberen te gaan → redirect naar `/client/overzicht`

Portal Switcher

Admin gebruikers zien een Portal Switcher in de header waarmee ze kunnen schakelen tussen:

- **Admin Portal:** Voor beheer en administratie
- **Client Portal:** Voor het testen van de client ervaring

Implementatie Details

- Component: `components/PortalSwitcher.tsx`
- Zichtbaar in: Admin layout en Client layout (alleen voor admins)
- Werking: Redirect naar `/admin/dashboard` of `/client/overzicht`

Bestanden Overzicht

Authentication

- `lib/auth-options.ts` : NextAuth configuratie met admin email lijst en role assignment
- `app/api/auth/[...nextauth]/route.ts` : NextAuth API route handler
- `types/next-auth.d.ts` : TypeScript type definitions voor NextAuth session

Routing

- `middleware.ts` : Central routing middleware met role-based access control
- `lib/routing-config.ts` : Route configuratie en helper functies

Layouts

- `app/admin/layout.tsx` : Admin layout met role check
- `app/client/layout.tsx` : Client layout
- `components/admin/AdminLayoutClient.tsx` : Admin layout client component (met PortalSwitcher)
- `components/dashboard/unified-layout.tsx` : Unified client layout
- `components/dashboard/header.tsx` : Dashboard header (met PortalSwitcher)

Login

- `app/inloggen/page.tsx` : Login pagina met role-based redirect logic

Troubleshooting

Admin User Heeft Geen Toegang Tot Admin Portal

Probleem: Admin gebruiker (info@writgo.nl) zit in Client Portal in plaats van Admin Portal.

Oplossing:

1. Verifieer dat het email adres exact overeenkomt met de lijst in `lib/auth-options.ts`
2. Log uit en log opnieuw in om een nieuwe session te krijgen
3. Check de browser console voor errors
4. Verifieer dat de user in de database de juiste role heeft

Redirect Loop

Probleem: Oneindige redirect tussen portals.

Oplossing:

1. Check de middleware configuratie in `middleware.ts`
2. Verifieer dat de NextAuth session correct is
3. Clear browser cookies en cache
4. Check dat de login redirect logic correct is in `app/inloggen/page.tsx`

Portal Switcher Niet Zichtbaar

Probleem: Admin gebruiker ziet de Portal Switcher niet.

Oplossing:

1. Verifieer dat de user role `admin` of `superadmin` is
2. Check de NextAuth session in browser dev tools
3. Verifieer dat `PortalSwitcher` component correct is geïmporteerd in beide layouts

Database Schema

User Table (Admin Users)

```
Table: User
- id: String (Primary Key)
- email: String (Unique)
- password: String (Hashed)
- name: String
- role: String ('admin', 'superadmin', 'client')
```

Client Table (Client Users)

```
Table: Client
- id: String (Primary Key)
- email: String (Unique)
- password: String (Hashed)
- name: String
```

Note: Client users kunnen admin rechten krijgen als hun email in de `adminEmails` lijst staat.

How to Add New Admin Users

Methode 1: Via adminEmails Array (Aanbevolen)

1. Open `lib/auth-options.ts`
2. Voeg het nieuwe email adres toe aan de `adminEmails` array (regel 8)
3. Deploy de wijziging
4. De gebruiker logt uit en opnieuw in

```
const adminEmails = ['info@writgoai.nl', 'info@writgo.nl', 'new-admin@example.com'];
```

Methode 2: Via Database

1. Maak een user aan in de `User` tabel
2. Zet de `role` kolom op `'admin'` of `'superadmin'`
3. De gebruiker kan direct inloggen met admin rechten

Testing Admin Access

Test Scenario's

1. **Login als Admin:**
 - Login met `info@writgo.nl`
 - Verifieer redirect naar `/admin/dashboard`
 - Verifieer toegang tot `/admin/projects/new`

2. Portal Switcher:

- Login als admin
- Klik op Portal Switcher in header
- Verifieer switch tussen Admin en Client portal

3. Access Control:

- Login als normale client
- Probeer `/admin/dashboard` direct te bezoeken
- Verifieer redirect naar `/client/overzicht`

4. Legacy Routes:

- Login als admin
- Navigeer naar `/dashboard`
- Verifieer redirect naar `/admin/dashboard`

Security Considerations

- Admin emails zijn hardcoded in `auth-options.ts` - deze lijst moet worden beheerd via environment variables voor productie
- Wachtwoorden worden gehashed met bcryptjs
- Middleware controleert elke request op correct role
- NextAuth JWT strategy wordt gebruikt voor sessies

Toekomstige Verbeteringen

- [] Admin emails via environment variables
- [] Admin user management UI
- [] Audit logging voor admin actions
- [] Two-factor authentication voor admin users
- [] Role-based permissions (granular access control)