

Stack Auth voor Stock2coat: Opzetten en implementatie

Het **Stack Auth Multi-Tenant Starter Template** biedt een uitstekende basis voor Stock2coat, met ingebouwde multi-tenancy, moderne UI-componenten en een open-source architectuur die perfect past bij onze eerder besproken tech stack.

Waarom Stack Auth kiezen voor Stock2coat?

Stack Auth onderscheidt zich als ideale keuze voor Stock2coat door zijn combinatie van open-source flexibiliteit, ingebouwde multi-tenancy en competitieve prijsstelling^[1] ^[2]. In tegenstelling tot Clerk biedt Stack Auth volledige self-hosting mogelijkheden en geen vendor lock-in^[3] ^[2].

Vergelijking met onze eerder voorgestelde Clerk-stack

Feature	Stack Auth	Clerk
Kosten	Gratis + Team \$49/m	10K MAU gratis + \$25/m
Multi-tenancy	Native teams/organisations	Ingebouwde organisations
Open-source	100% MIT/AGPL	Closed source
Self-hosting	Volledig mogelijk	Niet mogelijk
Next.js integratie	Optimaal (App Router)	Uitstekend
Vendor lock-in	Geen	Hoog

Stack Auth voordelen voor poederlakbedrijven

- **Data-soevereiniteit:** volledige controle over klantgegevens, cruciaal voor compliance in de chemische industrie^[2] ^[4]
- **Kostenefficiëntie:** geen per-MAU kosten na de gratis tier, ideaal voor groeiende MKB-bedrijven^[1] ^[5]
- **Flexibiliteit:** aanpasbare UI-componenten die passen bij industriële workflows^[6] ^[2]

Implementatie van de Stack Auth template

Stap 1: Project initialisatie

Begin met het klonen van de Stack Auth multi-tenant starter template:

```
git clone https://github.com/stack-auth/multi-tenant-starter-template.git stock2coat
cd stock2coat
```

```
npm install
```

Stap 2: Stack Auth configuratie

Initialiseer Stack Auth met de setup wizard^[4]:

```
npx @stackframe/init-stack@latest
```

De wizard detecteert automatisch je project structuur en configureert Stack Auth. Na voltooiing word je doorverwezen naar de browser om een account aan te maken.

Stap 3: Multi-tenant setup voor poederlakbedrijven

In het Stack Auth dashboard^[4]:

1. **Project configuratie:** geef je project de naam "Stock2coat"
2. **Authentication methods:** schakel email/password en OAuth providers (Google, Microsoft) in
3. **Teams feature:** activeer "client team creation" voor multi-tenancy^[7] ^[8]

Stap 4: Environment variabelen

Configureer je `.env.local` bestand met de API keys van Stack Auth^[8]:

```
STACK_PROJECT_ID=your_project_id  
STACK_PUBLISHABLE_CLIENT_KEY=your_publishable_key  
STACK_SECRET_SERVER_KEY=your_secret_key
```

Stap 5: Aanpassingen voor Stock2coat

Database schema uitbreiden voor voorraad-specifieke entiteiten:

```
// lib/schema.ts - voeg toe aan bestaande schema  
export const inventory = pgTable('inventory', {  
  id: serial('id').primaryKey(),  
  sku: varchar('sku', { length: 100 }).notNull(),  
  ralCode: varchar('ral_code', { length: 20 }),  
  brand: varchar('brand', { length: 100 }),  
  quantity: integer('quantity').default(0),  
  location: varchar('location', { length: 100 }),  
  teamId: varchar('team_id').references(() => teams.id), // Multi-tenant isolation  
  updatedAt: timestamp('updated_at').defaultNow(),  
});
```

API routes aanpassen voor inventory management:

```
// app/api/inventory/route.ts
import { useUser } from '@stackframe/stack';

export async function GET() {
  const user = useUser();
  const currentTeam = user.selectedTeam;

  if (!currentTeam) {
    return new Response('No team selected', { status: 400 });
  }

  // Haal inventory op voor current team
  const inventory = await db.inventory.findMany({
    where: { teamId: currentTeam.id }
  });

  return Response.json(inventory);
}
```

Aangepaste onboarding voor poederlakbedrijven

Team creation flow

Implementeer een aangepaste team-aanmaak flow specifiek voor poederlakbedrijven^[9]:

```
// components/coating-company-onboarding.tsx
export function CoatingCompanyOnboarding() {
  return (
    <div className="space-y-6">
      <h2>Welkom bij Stock2coat</h2>
      <form>
        <Input placeholder="Bedrijfsnaam" />
        <Select>
          <option>Poedercoating</option>
          <option>Natlakken</option>
          <option>Industriële coating</option>
        </Select>
        <Input placeholder="Aantal productie-lijnen" type="number" />
        <Button onClick={createTeamWithIndustryData}>
          Team aanmaken
        </Button>
      </form>
    </div>
  );
}
```

Role-based permissions voor coating workflows

```
// lib/permissions.ts
export const coatingRoles = {
  owner: ['manage_team', 'view_all_inventory', 'manage_recipes'],
  manager: ['view_inventory', 'update_quantities', 'manage_orders'],
  operator: ['view_inventory', 'update_status'],
  readonly: ['view_inventory']
};
```

Integratie met bestaande Stock2coat architectuur

Supabase + Stack Auth combinatie

Stack Auth werkt naadloos samen met Supabase voor database operations^[10]:

```
// lib/supabase-client.ts
import { createClientComponentClient } from '@supabase/auth-helpers-nextjs';
import { useUser } from '@stackframe/stack';

export function useSupabaseWithTeam() {
  const user = useUser();
  const supabase = createClientComponentClient();

  // Voeg team context toe aan alle queries
  const getInventoryForTeam = () => {
    return supabase
      .from('inventory')
      .select('*')
      .eq('team_id', user.selectedTeam?.id);
  };

  return { supabase, getInventoryForTeam };
}
```

Realtime updates per team

```
// hooks/use-team-inventory.ts
export function useTeamInventory() {
  const { user } = useUser();
  const [inventory, setInventory] = useState([]);

  useEffect(() => {
    if (!user.selectedTeam) return;

    const channel = supabase
      .channel(`inventory:${user.selectedTeam.id}`)
      .on('postgres_changes', {
        event: '*',
        schema: 'public',
        table: 'inventory',
```

```

    filter: `team_id=eq.${user.selectedTeam.id}`
  }, (payload) => {
    // Update local state
  })
  .subscribe();

  return () => supabase.removeChannel(channel);
}, [user.selectedTeam]);
}

```

Production deployment overwegingen

Self-hosting Stack Auth

Voor maximale controle en kostenefficiëntie kun je Stack Auth zelf hosten^[11] ^[4]:

```

# docker-compose.yml voor Stack Auth
version: '3.8'
services:
  stack-auth:
    image: stackframe/stack
    environment:
      - DATABASE_URL=${DATABASE_URL}
      - NEXTAUTH_SECRET=${NEXTAUTH_SECRET}
    ports:
      - "3001:3000"

```

Migratie van Clerk naar Stack Auth

Mocht je al begonnen zijn met Clerk, Stack Auth biedt migratietools^[12] ^[11]:

```

# Exporteer gebruikers uit Clerk
# Importeer in Stack Auth via API
curl -X POST https://api.stack-auth.com/v1/import \
  -H "Authorization: Bearer ${STACK_SECRET_KEY}" \
  -d @users-export.json

```

Conclusie en aanbeveling

Stack Auth biedt Stock2coat een kosteneffectieve, schaalbare oplossing die perfect aansluit bij onze open-source tech stack (Next.js + Supabase). De ingebouwde multi-tenancy, competitive pricing (\$49/maand vs Clerk's escalerende kosten) en volledige data-controle maken het ideaal voor een B2B SaaS in de poederlak-industrie.

Concrete volgende stappen:

1. Clone het Stack Auth starter template
2. Configureer multi-tenancy voor poederlakbedrijven
3. Integreer met je bestaande Supabase database

4. Implementeer aangepaste onboarding workflows

5. Test de complete flow van team-creatie tot voorraad-beheer

Met Stack Auth behoud je de flexibiliteit en kostencontrole die cruciaal zijn voor een succesvolle solo-developer lancering, terwijl je toekomstbestendige enterprise-features krijgt voor schaalbaarheid.



1. https://dev.to/tanishk_kumarshrivastava/authentication-system-in-nextjs-using-authjs-20g3
2. <https://github.com/stack-auth/stack-auth>
3. https://www.reddit.com/r/nextjs/comments/1hc21ad/multitenant_app_stack_betterauth_nextjs_nile/
4. <https://stack-auth.com/pricing>
5. https://www.youtube.com/watch?v=0eu4_ILFkGk
6. <https://betterstack.com/community/guides/scaling-nodejs/better-auth-vs-nextauth-authjs-vs-auth0/>
7. <https://zenstack.dev/blog/stackauth-multitenancy>
8. <https://news.ycombinator.com/item?id=41092942>
9. <https://stackoverflow.com/questions/75366149/multi-tenancy-whit-jwt-autentication-and-validation-of-user-for-tenant>
10. <https://bucket.co/blog/clerk>
11. <https://dev.to/martygo/top-3-best-authentication-frameworks-for-2025-51ej>
12. <https://learn.microsoft.com/en-us/entra/identity/multi-tenant-organizations/overview>