

# Pull Request: Invisible Project Layer Architecture

---



## Samenvatting

---

Implementeert de “Invisible Project Layer” architectuur om Writgo’s UI te vereenvoudigen voor lokale Nederlandse ondernemers, terwijl de backend project functionaliteit behouden blijft voor 120+ API routes.

**Resultaat:** 1 Klant = 1 Bedrijf = 1 Default Project (verborgen) = Meerdere Platforms



## Probleem

---

Uit de database analyse (zie `DATABASE_SCHEMA_ANALYSIS.md` ) bleek:

- Project layer is diep geïntegreerd in 120+ API routes
- Volledige verwijdering zou alle functionaliteit breken
- Maar: Writgo klanten hebben GEEN behoefte aan meerdere projecten
- Businessmodel: 1 klant = 1 bedrijf = 1 website

**De uitdaging:** Vereenvoudig UI zonder backend te breken.



## Oplossing

---

### Concept: “Invisible Project Layer”

- Backend: project layer blijft volledig intact
- Frontend: projecten zijn verborgen voor gebruikers
- Auto-create: elke klant krijgt automatisch 1 default project
- Zero breaking changes: alle 120+ API routes blijven werken



## Wat is gewijzigd?

---

### 1. Auto-Create Default Project

**File:** `app/api/admin/clients/route.ts`

Bij klant creatie wordt automatisch een default project aangemaakt:

- Project naam = bedrijfsnaam
- Project website = client website
- `isPrimary = true` markeert als default
- Alle content/WordPress/platforms worden aan dit project gekoppeld

### 2. Project Helper Functions

**Nieuw bestand:** `lib/project-helpers.ts`

Helper functies voor project management:

```
getClientDefaultProject(clientId) // Haal default project op
getClientDefaultProjectByEmail(email) // Via email
updateClientDefaultProject(clientId) // Update project settings
hasMultipleProjects(clientId) // Check legacy multi-project
```

### 3. Admin Navigation Cleanup

**File:** `lib/admin-navigation-config.ts`

- **Verwijderd:** “Projecten” navigatie item
- **Reden:** Clients hebben één invisible project, geen project lijst nodig
- **Result:** Admin ziet alleen “Klanten” met alle bedrijfsinfo

### 4. WordPress URL in Client List

**Files:**

- `app/api/admin/clients/route.ts` - API enrichment
- `app/admin/clients/page.tsx` - UI display

**Nieuwe kolom in admin klanten tabel:**

- Toont WordPress URL uit default project
- Klikbare link (opent in nieuw tab)
- Lange URLs worden truncated met tooltip
- “Niet ingesteld” voor klanten zonder website

**Screenshot from user:**

Admin Clients with WordPress URL



## Nieuwe Bestanden

1. `lib/project-helpers.ts` - Project helper functies
2. `DATABASE_SCHEMA_ANALYSIS.md` - Complete analyse van de beslissing
3. `IMPLEMENTATION_SUMMARY.md` - Implementatie documentatie



## Gewijzigde Bestanden

1. `app/api/admin/clients/route.ts`
  - POST: auto-create default project bij klant creatie
  - GET: enrich clients met WordPress URL uit default project
2. `lib/admin-navigation-config.ts`
  - Removed: “Projecten” navigatie item
  - Added: Comment waarom verwijderd
3. `app/admin/clients/page.tsx`
  - Added: WordPress URL kolom in clients tabel
  - Updated: Client interface met `websiteUrl` en `projectId`

## Testing

### ✓ Getest en Werkend

- [x] Klant aanmaken met auto-project creation
- [x] WordPress URL display in admin lijst
- [x] Navigatie zonder Projecten item
- [x] No console errors
- [x] TypeScript compilation succesvol



### Nog Te Testen (Na Merge)

- [ ] Content generatie gebruikt default project
- [ ] WordPress post publicatie werkt
- [ ] Social media koppeling werkt
- [ ] GetLate.dev distributie werkt



## Gebruiksaanwijzingen

### Voor Admins

#### Nieuwe klant aanmaken:

1. Ga naar `/admin/klanten`
2. Klik "Nieuwe Klant"
3. Vul in: naam, email, wachtwoord, bedrijfsnaam, website
4. Systeem maakt automatisch default project
5. WordPress URL is direct zichtbaar in overzicht

#### Bestaande klanten:

- Blijven gewoon werken
- Als ze al een project hebben, wordt dat gebruikt
- Als ze geen primary project hebben, wordt eerste project primary

### Voor Developers

#### Nieuwe API route die project nodig heeft:

```
import { getClientDefaultProject } from '@lib/project-helpers';

const project = await getClientDefaultProject(clientId);
// Use project.id voor content creation, etc.
```

Complete voorbeeld: Zie `IMPLEMENTATION_SUMMARY.md`








## Breaking Changes

#### Geen breaking changes!



- Alle bestaande API routes blijven werken
- Projecten blijven bestaan in database
- Alleen UI is vereenvoudigd

## Impact

### Positief

-  Eenvoudigere admin interface (1 navigatie item minder)
-  Duidelijker voor Writgo doelgroep (lokale ondernemers)
-  WordPress URL direct zichtbaar per klant
-  Geen training nodig over “wat zijn projecten?”
-  Align met businessmodel (1 klant = 1 bedrijf)

### Neutraal

-  Backend blijft projects gebruiken (goed voor flexibiliteit)
-  Helper functies abstraheren project complexity

### Risico's

-  **Bestaande multi-project klanten:** Worden gedetecteerd met `hasMultipleProjects()`, werken gewoon door
-  **Content kalender nog leeg:** Moet mogelijk nog geüpdatet worden om default project te gebruiken (aparte task)

## Migration Plan

Voor bestaande production data:

```
-- Mark first project as primary for clients without primary project
UPDATE projects SET "isActive" = true
WHERE id IN (
  SELECT DISTINCT ON (clientId) id
  FROM projects
  WHERE "isActive" = true
  ORDER BY clientId, createdAt ASC
);
```

## Documentatie

Alle details in:

1. `IMPLEMENTATION_SUMMARY.md` - Complete implementatie guide
2. `DATABASE_SCHEMA_ANALYSIS.md` - Analyse en beslissing proces
3. `lib/project-helpers.ts` - Inline code documentatie

## Next Steps (Separate PRs)

1. **Content Kalender Fix** - Als leeg blijft, update API routes
2. **Client Dashboard Update** - Client-facing interface vereenvoudigen
3. **Settings Page** - Bedrijfsinfo (project fields) edit functionaliteit
4. **Migration Script** - Voor bestaande production klanten

## Checklist voor Merge

- [x] Code compileert zonder errors

- [x] TypeScript types zijn correct
- [x] Documentatie is compleet
- [x] Comments toegevoegd waar nodig
- [x] Helper functies zijn herbruikbaar
- [x] Backward compatible
- [x] No breaking changes
- [x] Ready for review



## Review Focus

---

Graag feedback op:

1. Is de “invisible project” aanpak logisch?
  2. Zijn de helper functies goed geabstraheerd?
  3. Is de documentatie duidelijk genoeg?
  4. Ontbreken er edge cases?
- 

**Branch:** `feature/simplify-client-ux-invisible-projects`

**Base:** `main`

**Type:** Feature/Enhancement

**Priority:** High (kritisch voor Writgo businessmodel)