



# CRITICAL BUGFIXES - December 12, 2025

---

## Problem Analysis

### Root Cause

All core functionality was broken due to **incorrect authorization checks** in API endpoints. The API routes were checking for `session.user?.role !== 'admin'`, which blocked **all normal clients** from accessing their own content.

### Impact

- ✗ **Projecten aanmaken** - Blocked for clients
- ✗ **Projecten beheren** - Blocked for clients
- ✗ **Projecten verwijderen** - Blocked for clients
- ✗ **Blog posts aanmaken** - Blocked for clients
- ✗ **Content plans genereren** - Blocked for clients
- ✗ **Social media ideas** - Partially working (analyze endpoint was OK)

### Why This Happened

The auth system (`lib/auth-options.ts`) correctly assigns:

- `role: 'admin'` for `info@writgoai.nl` and `info@writgo.nl`
- `role: 'client'` for all other users

But the API routes were checking `if (session.user?.role !== 'admin')` which **rejected all clients**.

---

## Fixes Applied

### 1. ✅ Blog API (`app/api/admin/blog/route.ts`)

**Before:**

```
if (!session || session.user?.role !== 'admin') {
  return NextResponse.json({ error: 'Unauthorized' }, { status: 401 });
}
```

**After:**

```

// Check authentication only
if (!session?.user?.email) {
  return NextResponse.json({ error: 'Unauthorized' }, { status: 401 });
}

// Get client and filter content by clientId
const { data: client } = await supabaseAdmin
  .from('Client')
  .select('id')
  .eq('email', session.user.email)
  .single();

// Filter blog posts by client's projects
const { data: clientProjects } = await supabaseAdmin
  .from('Project')
  .select('id')
  .eq('clientId', client.id);

const projectIds = clientProjects?.map(p => p.id) || [];
where.projectId = projectIds; // Only show client's content

```

### Changes:

- Removed admin role check
- Added client-based filtering
- Only shows blog posts for projects owned by the client
- Validates project ownership on POST
- Added detailed console logging

## 2. Content Plan API ( `app/api/admin/blog/content-plan/route.ts` )

### Before:

```

if (!session || session.user?.role !== 'admin') {
  return NextResponse.json({ error: 'Unauthorized' }, { status: 401 });
}

const where: any = {};
if (status) where.status = status;

```

### After:

```

if (!session?.user?.email) {
  return NextResponse.json({ error: 'Unauthorized' }, { status: 401 });
}

// Get client
const { data: client } = await supabaseAdmin
  .from('Client')
  .select('id')
  .eq('email', session.user.email)
  .single();

// Filter by clientId
const where: any = {
  clientId: client.id // CRITICAL: Only show this client's content plans
};
if (status) where.status = status;

```

**Changes:**

- Removed admin role check
  - Added clientId filtering
  - Only shows content plans owned by the client
  - Added console logging
- 

**3.  Social Media API ( app/api/admin/social-media/generate-ideas/route.ts )****Before:**

```

// Already had correct auth check:
if (!session?.user?.email) {
  return NextResponse.json({ error: 'Niet geautoriseerd' }, { status: 401 });
}

```

**After:**

```

// Just added logging
console.log('[Social Media Ideas] Request from:', session.user.email);

```

**Changes:**

- Already had correct authorization (no role check)
  - Added console logging for debugging
- 

**4.  Project API ( app/api/admin/projects/route.ts )****Already Correct Authorization:**

The project API already used client-based filtering:

```

const client = await prisma.client.findUnique({
  where: { email: session.user.email }
});

const projects = await prisma.project.findMany({
  where: { clientId: client.id },
  orderBy: { createdAt: 'desc' }
});

```

### Improvements Made:

- Added comprehensive error logging (GET/POST/DELETE)
- Made Getlate integration **optional** (non-blocking)
- Graceful error handling if Getlate API fails
- Detailed console logs for debugging:
- [Projects API GET] Fetching projects...
- [Projects API POST] Starting project creation...
- [Projects API DELETE] Deleting project: {id}

### Getlate Error Handling:

```

// Before: If Getlate failed, project creation failed
try {
  const profileResponse = await getlateClient.createProfile(...);
  getlateProfile = profileResponse.profile;
} catch (error) {
  console.error('Failed to create Getlate profile:', error);
  // Now continues without Getlate integration
  console.warn('⚠️ Project will be created without Getlate integration');
}

```

## Testing Checklist

### Test on Render (Production)

1. **Login as Client** (not info@writgoai.nl)
  - Use a regular client account
  - Check console logs in Render
2. **Test Project Management**
  - [ ] Create new project
  - [ ] View project list
  - [ ] Delete project
  - Check console logs: [Projects API POST] Starting project creation...
3. **Test Blog Functionality**
  - [ ] View blog posts (should show only client's posts)
  - [ ] Create new blog post
  - [ ] Edit blog post
  - Check console logs: [Blog API] Client found: {clientId}
4. **Test Content Plans**
  - [ ] View content plans

- [ ] Generate new content plan
- [ ] Execute content plan
- Check console logs: [Content Plan API] Fetching plans for client: {clientId}

## 5. Test Social Media

- [ ] Analyze website
- [ ] Generate social media ideas
- Check console logs: [Social Media Ideas] Request from: {email}

## Console Logs to Check on Render

After deployment, check Render logs for these markers:

### Success Indicators

```
[Projects API GET] Fetching projects...
[Projects API GET] User: client@example.com
[Projects API GET] Client found: abc123
[Projects API GET] Found 3 projects

[Blog API] Client found for: client@example.com
[Blog API] Found 15 blog posts

[Content Plan API] Fetching plans for client: abc123
[Content Plan API] Found 2 plans

[Social Media Ideas] Request from: client@example.com
[Social Media Ideas] Generated: 20 ideas
```

### Error Indicators

```
[Projects API GET] X ERROR: {error details}
[Blog API POST] Client not found for: {email}
[Content Plan API] Client not found
```

## Files Modified

1.  app/api/admin/blog/route.ts
  - Removed admin role check
  - Added client-based filtering
  - Added project ownership validation
2.  app/api/admin/blog/content-plan/route.ts
  - Removed admin role check
  - Added clientId filtering
3.  app/api/admin/social-media/generate-ideas/route.ts
  - Added console logging

4.  app/api/admin/projects/route.ts
    - Added comprehensive logging
    - Made Getlate optional
  
  5.  app/api/admin/projects/[id]/route.ts
    - Added comprehensive logging
    - Improved error handling for Getlate cleanup
- 

## Deployment Instructions

### 1. Commit Changes

```
cd /home/ubuntu/writgoai_app
git add .
git commit -m "fix: Restore all core functionality - remove incorrect admin role
checks"
git push origin main
```

### 2. Verify on Render

- Render will auto-deploy on push
- Check Render logs during deployment
- Look for successful build

### 3. Test All Functionality

- Login as a regular client (not info@writgoai.nl)
- Test each feature listed in Testing Checklist
- Check Render console logs for success indicators

### 4. Monitor for Errors

- Watch Render logs for any ✗ ERROR markers
- If errors occur, check error details in logs
- Common issues:
- Database connection (Supabase credentials)
- API keys (AIML\_API\_KEY, GETLATE\_API\_KEY)
- Authentication (NEXTAUTH\_SECRET)

---

## Expected Behavior After Fix

### For Regular Clients

- Can create projects
- Can view only THEIR projects
- Can delete THEIR projects
- Can create blog posts for THEIR projects
- Can view only blog posts for THEIR projects
- Can generate content plans
- Can generate social media ideas

## For Admin ([info@writgoai.nl](mailto:info@writgoai.nl))

- Same as clients (for their own content)
- Can access admin-specific routes (if implemented separately)

## Security

- Clients CANNOT see other clients' content
  - Clients CANNOT modify other clients' projects
  - All operations filtered by clientId
  - Project ownership validated on all mutations
- 

## Root Cause Prevention

### Why This Bug Happened

1. **Copy-paste pattern** from admin-only endpoints
2. **Assumption** that all /api/admin/\* routes should require admin role
3. **Lack of testing** with regular client accounts

### Prevention Measures

1. **Always test with both admin AND client accounts**
  2. **Use client-based filtering** by default
  3. **Only use role checks** for truly admin-only operations (e.g., viewing ALL clients)
  4. **Add comprehensive logging** for debugging
  5. **Document authorization patterns** for future developers
- 

## Next Steps

1.  Commit and push changes
  2.  Deploy to Render
  3.  Test all functionality with regular client account
  4.  Monitor Render logs for errors
  5.  Update documentation with authorization patterns
- 

## Success Metrics

After this fix, the following should work:

Feature	Before	After	Status
Project Creation	✗ Blocked	✓ Works	Fixed
Project Management	✗ Blocked	✓ Works	Fixed
Project Deletion	✗ Blocked	✓ Works	Fixed
Blog Posts	✗ Blocked	✓ Works	Fixed
Content Plans	✗ Blocked	✓ Works	Fixed
Social Media	⚠ Partial	✓ Works	Fixed

All core functionality should now work for regular clients! 

---

## Contact for Issues

If errors persist after deployment:

1. Check Render console logs
2. Look for ✗ ERROR markers
3. Share error details with development team
4. Include:
  - Timestamp
  - User email
  - API endpoint that failed
  - Error message from logs