

Sitemap Ophalen Fix - Samenvatting

Probleem

Gebruiker meldde: "sitemap opgehaald 0 gevonden"

- Backend vond 0 URLs bij het ophalen van WordPress sitemaps
- Sitemap parsing werkte niet correct

Root Cause Analyse

1. **Frontend/Backend mismatch:** Backend returneerde `site_data.links` maar frontend verwachtte `pages_found` en `pages`
2. **Beperkte sitemap URL detectie:** Alleen `/sitemap.xml` werd geprobeerd
3. **Sitemap index handling:** Alleen eerste sub-sitemap werd opgehaald in `backend_utils.py`
4. **Geen error handling:** Bij 0 URLs werd geen duidelijke foutmelding gegeven

Geïmplementeerde Fixes

1. Frontend/Backend Response Fix (app.py)

Bestand: `app.py` (regel 1978-2001)

Probleem: Backend returneerde verkeerde data structuur

```
# VOOR (verkeerd):
return jsonify({
    'success': True,
    'site_data': {
        'links': urls,
        'links_count': len(urls)
    }
})
```

Oplossing: Toegevoegd `pages_found` en `pages` voor frontend

```
# NA (correct):
return jsonify({
    'success': True,
    'pages_found': len(urls), # ✅ Frontend verwacht dit
    'pages': pages,          # ✅ Frontend verwacht dit
    'site_data': {           # Behouden voor backwards compatibility
        'url': url,
        'domain': domain,
        'name': site_name,
        'links': urls,
        'links_count': len(urls)
    }
})
```

2. Multiple Sitemap URL Detection (app.py)

Bestand: app.py (regel 1927-1960)

Probleem: Alleen /sitemap.xml werd geprobeerd

Oplossing: Probeert nu meerdere WordPress sitemap formaten:

```
sitemap_urls = [
    f"{base_url}/sitemap.xml",           # Yoast SEO
    f"{base_url}/sitemap_index.xml",     # Yoast SEO index
    f"{base_url}/wp-sitemap.xml",        # WordPress core
    f"{base_url}/sitemap-index.xml",     # Alternatief
    f"{base_url}/post-sitemap.xml"      # Direct post sitemap
]

# Probeert elke URL tot er één werkt
for test_url in sitemap_urls:
    try:
        test_response = requests.get(test_url, timeout=15)
        if test_response.status_code == 200:
            response = test_response
            sitemap_url = test_url
            print(f"✅ Found sitemap at: {test_url}")
            break
    except Exception as e:
        continue
```

3. Zero URLs Error Handling (app.py)

Bestand: app.py (regel 2004-2010)

Probleem: Geen duidelijke foutmelding bij 0 URLs

Oplossing: Expliciete check en error message


```
if not urls:
    print(f"⚠️ Sitemap found at {sitemap_url} but contains 0 URLs")
    return jsonify({
        'success': False,
        'error': f'Sitemap gevonden maar bevat geen URLs. Controleer of de sitemap correct is: {sitemap_url}'
    }), 400
```

4. Improved Sitemap Index Handling (backend_utils.py)

Bestand: backend_utils.py (regel 262-342)

Probleem: Alleen eerste sub-sitemap werd opgehaald

Oplossing: Haalt nu alle sub-sitemaps op (max 5)

```
# Check if it's a sitemap index
sitemaps = soup.find_all('sitemap')
if sitemaps:
    print(f"

```

5. Better XML Parsing

Beide bestanden: Gebruik van BeautifulSoup met 'xml' parser

Voordeel:

- Betere namespace handling
- Robuuster tegen malformed XML
- Consistente parsing tussen app.py en backend_utils.py

Ondersteunde Sitemap Formaten

WordPress Core Sitemap

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>https://example.com/post-1/</loc>
    <lastmod>2025-10-09</lastmod>
  </url>
</urlset>
```

URL: /wp-sitemap.xml

Yoast SEO Sitemap Index

```
<?xml version="1.0" encoding="UTF-8"?>
<sitemapindex xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <sitemap>
    <loc>https://example.com/post-sitemap.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://example.com/page-sitemap.xml</loc>
  </sitemap>
</sitemapindex>
```

URLs: /sitemap.xml , /sitemap_index.xml

Yoast SEO Post Sitemap

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>https://example.com/blog-post/</loc>
  </url>
</urlset>
```

URL: /post-sitemap.xml

Test Resultaten

Test 1: Yoast SEO Sitemap Index

URL: https://yoast.com/sitemap_index.xml
 ✓ SUCCESS: Found 10 URLs
 - Detecteert sitemap index met 21 sub-sitemaps
 - Haalt URLs op uit eerste 5 sub-sitemaps
 - Extraheert titels uit URL paths

Test 2: TechCrunch Sitemap

URL: https://techcrunch.com/sitemap.xml
 ✓ SUCCESS: Found 5 URLs
 - Detecteert sitemap index met 2023 sub-sitemaps
 - Haalt URLs op uit eerste 5 sub-sitemaps
 - Werkt met grote sitemaps

Verbeteringen

Error Handling

- ✓ Duidelijke foutmeldingen bij geen sitemap gevonden
- ✓ Specifieke melding bij 0 URLs in sitemap
- ✓ Timeout handling (15 seconden per request)
- ✓ HTTP error handling met status codes

Logging

- ✓ Gedetailleerde console logs voor debugging
- ✓ Progress indicators (🔍, 📄, 📄, ✓, ⚠️, ✖️)
- ✓ URL counts per sub-sitemap

Performance

- ✓ Timeout van 15 seconden per request
- ✓ Maximum 5 sub-sitemaps bij sitemap index
- ✓ Configurable max_urls parameter (default 100)

Data Structuur

- ✓ Consistente return format: `[{'url': '...', 'title': '...'}]`
- ✓ Automatische titel extractie uit URL path
- ✓ Backwards compatible met oude code

Gewijzigde Bestanden

1. **app.py**

- `/api/scrape-website` endpoint (regel 1900-2050)
- Multiple sitemap URL detection
- Frontend response format fix
- Zero URLs error handling

2. **backend_utils.py**

- `fetch_sitemap_urls()` functie (regel 262-342)
- Improved sitemap index handling
- Better error handling
- BeautifulSoup XML parsing

3. **test_sitemap.py** (nieuw)

- Test script voor sitemap functionaliteit
- Valideert beide sitemap formaten

Gebruik

Via Frontend (Website Beheer)

```
// Gebruiker voert WordPress URL in
const siteUrl = "https://example.com";

// Frontend roept API aan
fetch('/api/scrape-website', {
  method: 'POST',
  body: JSON.stringify({ site_url: siteUrl })
});

// Response:
{
  "success": true,
  "pages_found": 150,
  "pages": [
    {"url": "https://example.com/post-1/", "title": "Post 1"},
    ...
  ]
}
```

Via Backend (Artikel Generatie)

```
from backend_utils import fetch_sitemap_urls

# Direct sitemap URL
urls = fetch_sitemap_urls("https://example.com/sitemap.xml", max_urls=50)

# Returns: [{'url': '...', 'title': '...'}, ...]
```

Backwards Compatibility

✅ Alle oude functionaliteit blijft werken:

- `fetch_wordpress_sitemap()` in `app.py` (regel 676)
- `parse_sitemap_urls()` in `app.py` (regel 742)
- Interne link toevoeging aan artikelen

Conclusie

De sitemap ophalen functionaliteit is nu volledig gefixed en ondersteunt:

- ✅ Meerdere WordPress sitemap formaten
- ✅ Sitemap index met sub-sitemaps
- ✅ Correcte frontend/backend communicatie
- ✅ Duidelijke error messages
- ✅ Robuuste XML parsing
- ✅ Goede logging en debugging

Resultaat: Gebruiker zal nu URLs zien in plaats van “0 gevonden”!