Fase 4: Projects Route Ontwerp

🎯 RESTful API Structuur (21 → 11 Routes)

Nieuwe Route Structuur

```
/api/client/projects/
├── route.ts                                    [GET, POST]
├── [id]/
│   ├── route.ts                                [GET, PUT, PATCH, DELETE]
│   ├── collaborators/
│   │   └── route.ts                            [GET, POST, DELETE]
│   ├── knowledge/
│   │   ├── route.ts                            [GET, POST]
│   │   └── [knowledgeId]/
│   │       └── route.ts                        [PUT, DELETE]
│   ├── affiliate-links/
│   │   ├── route.ts                            [GET, POST]
│   │   └── [linkId]/
│   │       └── route.ts                        [GET, PATCH, DELETE]
│   ├── research/
│   │   └── [type]/
│   │       └── route.ts                        [GET, POST, PATCH]
│   ├── sitemap/
│   │   └── route.ts                            [GET, POST]
│   └── integrations/
│       ├── wordpress/
│       │   └── route.ts                        [GET, PUT, POST, DELETE]
│       └── woocommerce/
│           └── route.ts                        [GET, PUT]
```

📝 Detailed Route Specifications

1. `/api/client/projects/[id]/research/[type]` (NEW - Consolidates 3 routes)

Replaces:
- `content-analysis/route.ts` (194 lines)
- `content-strategy/route.ts` (194 lines)
- `keyword-research/route.ts` (194 lines)

Parameters:
- `type`: `analysis` | `strategy` | `keywords`

Methods:

GET - Haal research data op

```
// Query params: ?type=analysis|strategy|keywords (optional, returns all if omitted)
Response: {
  id: string;
  name: string;
  contentAnalysis?: any;
  contentAnalysisStatus?: string;
  contentAnalysisDate?: Date;
  contentStrategy?: any;
  contentStrategyStatus?: string;
  contentStrategyDate?: Date;
  keywordResearch?: any;
  keywordResearchStatus?: string;
  keywordResearchDate?: Date;
}
```

POST - Sla research data op

```
Body: {
  data: any;
  status?: 'not_started' | 'in_progress' | 'completed' | 'needs_review';
}
Response: {
  success: boolean;
  message: string;
  project: Project;
}
```

PATCH - Update alleen status

```
Body: {
  status: 'not_started' | 'in_progress' | 'completed' | 'needs_review';
}
Response: {
  success: boolean;
  message: string;
}
```

Field Mapping:

Type	Content Field	Status Field	Date Field
analysis	contentAnalysis	contentAnalysisStatus	contentAnalysisDate
strategy	contentStrategy	contentStrategyStatus	contentStrategyDate
keywords	keywordResearch	keywordResearchStatus	keywordResearchDate

Estimated Lines: ~220 lines (vs 582 lines) = **62% reduction**

---

2. `/api/client/projects/[id]/affiliate-links` (ENHANCED)

Replaces:

- `affiliate-feed/route.ts` (107 lines)
- `affiliate-links/route.ts` (300 lines)
- `affiliate-links/bulk/route.ts` (~150 lines)

- `bolcom/test/route.ts` (72 lines)
- `tradetracker-feed/route.ts` (212 lines)

Methods:

GET - Haal affiliate links op

```
// Query params: ?page=1&limit=50&search=...
Response: {
  links: AffiliateLink[];
  total: number;
  page: number;
  limit: number;
}
```

POST - Create/Import affiliate links

```
// Query params: ?source=manual|bolcom|tradetracker|bulk
// For source=bolcom|tradetracker: ?source=bolcom&test=true (test mode)

Body (manual): {
  url: string;
  title?: string;
  description?: string;
  price?: number;
  currency?: string;
}

Body (feed): {
  feedUrl?: string;
  productIds?: string[];
  campaignId?: string;
  credentials?: { apiKey: string; ... };
}

Body (bulk): {
  links: Array<{
    url: string;
    title?: string;
    description?: string;
  }>;
  generateTitles?: boolean;
}

Response: {
  success: boolean;
  message: string;
  links?: AffiliateLink[];
  imported?: number;
  errors?: string[];
}
```

Estimated Lines: ~400 lines (vs 691 lines) = **42% reduction**

---

3. `/api/client/projects/[id]/affiliate-links/[linkId]` (NEW)

Methods:

GET - Haal specifieke link op

```
Response: AffiliateLink
```

PATCH - Update affiliate link

```
Body: Partial<AffiliateLink>
Response: {
  success: boolean;
  link: AffiliateLink;
}
```

DELETE - Verwijder affiliate link

```
Response: {
  success: boolean;
  message: string;
}
```

Estimated Lines: ~80 lines (extracted from main route)

---

4. `/api/client/projects/[id]/sitemap` (ENHANCED)

Replaces:
- `sitemap/route.ts` (~100 lines)
- `[projectId]/load-sitemap/route.ts` (~150 lines)
- `rescan/route.ts` (~80 lines)

Methods:

GET - Load sitemap URLs

```
// Query params: ?limit=100&offset=0
Response: {
  urls: string[];
  total: number;
  lastScanned?: Date;
}
```

POST - Scan/Rescan sitemap

```
// Query params: ?action=scan|load
Body: {
  sitemapUrl?: string;
  forceRescan?: boolean;
}
Response: {
  success: boolean;
  message: string;
  urls?: string[];
  scannedCount: number;
}
```

Estimated Lines: ~250 lines (vs 330 lines) = **24% reduction**

---

5. `/api/client/projects/[id]/integrations/wordpress` (NEW)

Replaces:
- `wordpress/route.ts` (126 lines)
- `wordpress/test/route.ts` (95 lines)
- `auto-create-content-hub/route.ts` (120 lines)

Methods:

GET - Haal WordPress instellingen op

```
Response: {
  siteUrl?: string;
  username?: string;
  hasPassword: boolean;
  isConnected: boolean;
  contentHubId?: string;
}
```

PUT - Update WordPress instellingen

```
Body: {
  siteUrl: string;
  username: string;
  password: string;
}
Response: {
  success: boolean;
  message: string;
}
```

POST - WordPress actions

```
// Query params: ?action=test|create-hub
Body (test): {
  siteUrl: string;
  username: string;
  password: string;
}

Body (create-hub): {
  // Uses existing credentials
}

Response: {
  success: boolean;
  message: string;
  contentHubId?: string;
}
```

DELETE - Verwijder WordPress verbinding

```
Response: {
  success: boolean;
  message: string;
}
```

Estimated Lines: ~300 lines (vs 341 lines) = **12% reduction**

---

6. `/api/client/projects/[id]/integrations/woocommerce` (MOVED & ENHANCED)

Replaces:

- `woocommerce-settings/route.ts` (~150 lines)

Methods:

GET - Haal WooCommerce instellingen op

```
Response: {
  enabled: boolean;
  settings: any;
  isConfigured: boolean;
}
```

PUT - Update WooCommerce instellingen

```
Body: {
  enabled: boolean;
  settings: any;
}
Response: {
  success: boolean;
  message: string;
}
```

Estimated Lines: ~150 lines (similar, but better location)

7-11. Unchanged Routes (Keep As-Is)

7. `/api/client/projects` - GET, POST

- List all projects
- Create new project

8. `/api/client/projects/[id]` - GET, PUT, PATCH, DELETE

- Get project details
- Update project
- Delete project

9. `/api/client/projects/[id]/collaborators` - GET, POST, DELETE

- Manage project collaborators

10-11. `/api/client/projects/[id]/knowledge` - GET, POST

- Manage knowledge base items
- `/api/client/projects/[id]/knowledge/[knowledgeId]` - PUT, DELETE

📊 Code Reduction Summary

Category	Before (Routes/ Lines)	After (Routes/ Lines)	Reduction
Research Routes	3 routes / 582 lines	1 route / 220 lines	**62% lines**
Affiliate Routes	5 routes / 691 lines	2 routes / 480 lines	**31% lines**
Sitemap Routes	3 routes / 330 lines	1 route / 250 lines	**24% lines**
WordPress Routes	3 routes / 341 lines	1 route / 300 lines	**12% lines**
WooCommerce Routes	1 route / 150 lines	1 route / 150 lines	**0% lines (moved)**
Unchanged	6 routes / ~800 lines	6 routes / ~800 lines	**0%**
TOTAL	**21 routes / ~2894 lines**	**11 routes / ~2200 lines**	**~48% routes, ~24% lines**

🔧 Implementation Strategy

Phase 1: Create New Consolidated Routes

Step 1.1: Research Route

```
mkdir -p app/api/client/projects/[id]/research/[type]
# Create consolidated route with dynamic type handling
```

Step 1.2: Affiliate Links Enhancement

```
# Enhance existing affiliate-links route
# Create new [linkId] sub-route
mkdir -p app/api/client/projects/[id]/affiliate-links/[linkId]
```

Step 1.3: Sitemap Enhancement

```
# Enhance existing sitemap route with action parameter
```

Step 1.4: Integrations Structure

```
mkdir -p app/api/client/projects/[id]/integrations/{wordpress,woocommerce}
# Create consolidated integration routes
```

Phase 2: Route Migration Details

Research Route Implementation

```
// app/api/client/projects/[id]/research/[type]/route.ts

const FIELD_MAPPING = {
  analysis: {
    content: 'contentAnalysis',
    status: 'contentAnalysisStatus',
    date: 'contentAnalysisDate',
  },
  strategy: {
    content: 'contentStrategy',
    status: 'contentStrategyStatus',
    date: 'contentStrategyDate',
  },
  keywords: {
    content: 'keywordResearch',
    status: 'keywordResearchStatus',
    date: 'keywordResearchDate',
  },
};

export async function GET(req, { params }) {
  const { type } = params;
  const mapping = FIELD_MAPPING[type];

  if (!mapping) {
    return NextResponse.json({ error: 'Invalid type' }, { status: 400 });
  }

  // Fetch project with dynamic field selection
  // ... validation logic ...
}

export async function POST(req, { params }) {
  const { type } = params;
  const mapping = FIELD_MAPPING[type];
  const { data, status } = await req.json();

  // Update project with dynamic fields
  const updateData = {
    [mapping.content]: data,
    [mapping.status]: status || 'completed',
    [mapping.date]: new Date(),
  };

  // ... update logic ...
}
```

Affiliate Links Enhancement

```
// app/api/client/projects/[id]/affiliate-links/route.ts

export async function POST(req, { params }) {
  const searchParams = req.nextUrl.searchParams;
  const source = searchParams.get('source') || 'manual';
  const test = searchParams.get('test') === 'true';

  switch (source) {
    case 'manual':
      return handleManualLink(req, params);
    case 'bulk':
      return handleBulkImport(req, params);
    case 'bolcom':
      return handleBolcomFeed(req, params, test);
    case 'tradetracker':
      return handleTradeTrackerFeed(req, params, test);
    default:
      return NextResponse.json({ error: 'Invalid source' }, { status: 400 });
  }
}
```

Phase 3: Frontend Updates

Search for Frontend References

```
# Find all components using old routes
grep -r "api/client/projects.*content-analysis" app/
grep -r "api/client/projects.*content-strategy" app/
grep -r "api/client/projects.*keyword-research" app/
grep -r "api/client/projects.*affiliate-" app/
grep -r "api/client/projects.*wordpress" app/
grep -r "api/client/projects.*sitemap" app/
grep -r "api/client/projects.*woocommerce-settings" app/
```

Update Pattern

```
// OLD:
fetch(`/api/client/projects/${id}/content-analysis`)

// NEW:
fetch(`/api/client/projects/${id}/research/analysis`)

// OLD:
fetch(`/api/client/projects/${id}/affiliate-feed`, { method: 'POST', ... })

// NEW:
fetch(`/api/client/projects/${id}/affiliate-links?source=bolcom`, { method: 'POST', ..
. })
```

Phase 4: Backwards Compatibility (Optional)

Create redirect/proxy routes for critical old endpoints:

```
// app/api/client/projects/[id]/content-analysis/route.ts
export async function GET(req, { params }) {
  // Redirect to new endpoint
  return NextResponse.redirect(
    new URL(`/api/client/projects/${params.id}/research/analysis`, req.url)
  );
}
```

Phase 5: Testing Checklist

- [ ] Research routes (all 3 types)
- [ ] GET analysis
- [ ] POST strategy
- [ ] PATCH keywords status
- [ ] Affiliate links
- [ ] GET list
- [ ] POST manual link
- [ ] POST bulk import
- [ ] POST Bol.com feed (test mode)
- [ ] PATCH link update
- [ ] DELETE link
- [ ] Sitemap
- [ ] GET sitemap URLs
- [ ] POST rescan
- [ ] Integrations
- [ ] WordPress: GET, PUT, POST (test), POST (create-hub), DELETE
- [ ] WooCommerce: GET, PUT
- [ ] Build verification
- [ ] `npm run build` succeeds
- [ ] No TypeScript errors
- [ ] No broken imports

🚀 Deployment Checklist

- [ ] All new routes created
- [ ] All old route logic migrated
- [ ] Frontend references updated
- [ ] Tests passing
- [ ] Build successful
- [ ] Old routes removed (or redirected)
- [ ] Documentation updated
- [ ] Git commit with clear message
- [ ] Push to GitHub
- [ ] Deployment report created

📈 Expected Benefits

1. **Maintainability**: 48% fewer routes to maintain
2. **Consistency**: RESTful design with predictable patterns
3. **Discoverability**: Logical grouping under `/integrations/` and `/research/`
4. **Flexibility**: Query parameters allow for variations without new routes
5. **Code Reuse**: Shared validation and error handling logic
6. **Testing**: Fewer endpoints = easier testing
7. **Documentation**: Simpler API documentation
8. **Performance**: Reduced route registration overhead

⚠️ Migration Risks & Mitigations

Risk 1: Breaking Frontend Code

Mitigation:
- Search and update all frontend references before removing old routes
- Use TypeScript to catch errors at compile time
- Test all affected components

Risk 2: Query Parameter Confusion

Mitigation:
- Clear documentation of all query parameters
- Validate query parameters with helpful error messages
- Use TypeScript types for query parameter validation

Risk 3: Complex Route Logic

Mitigation:
- Break down complex routes into helper functions
- Add comprehensive comments
- Use switch statements for clarity

Risk 4: Build Failures

Mitigation:
- Test build after each major change
- Keep old routes until new routes are verified
- Use feature flags if needed