

Fase 2: Nieuwe Geconsolideerde API Structuur

Ontwerp Principes

1. **RESTful Design:** Gebruik standaard HTTP methods (GET, POST, PUT, DELETE)
2. **Resource-Based:** URLs representeren resources, niet acties
3. **Consistency:** Consistente naming en structuur
4. **Single Source of Truth:** Eén endpoint per functionaliteit
5. **Backwards Compatibility:** Waar mogelijk via redirects

Nieuwe API Structuur (14 routes)

 /api/client/social/

1. Posts Management (RESTful CRUD)

```
GET   /api/client/social/posts
POST  /api/client/social/posts
```

Functionaliteit:

- GET: List all posts (met filters: status, platform, date range)
- POST: Create new post (direct or draft)

Parameters:

- status : draft, scheduled, published
- platform : linkedin, twitter, facebook, instagram
- limit , offset voor pagination
- projectId

Consolidates:

- social-media-posts/route.ts
- social-media/posts/route.ts
- social/route.ts
- social-media/all-posts/route.ts

```
GET   /api/client/social/posts/[id]
PUT   /api/client/social/posts/[id]
DELETE /api/client/social/posts/[id]
```

Functionaliteit:

- GET: Get single post details
- PUT: Update post
- DELETE: Delete single post

Consolidates:

- social-media-posts/[postId]/route.ts
 - social/[id]/route.ts
-

```
POST /api/client/social/posts/bulk-delete
```

Functionaliteit:

- Bulk delete posts

Behouden: Specifieke bulk operation blijft apart

2. Content Generation (AI-powered)

```
POST /api/client/social/generate
```

Functionaliteit:

- Generate social media posts met AI
- Support voor verschillende types: post, carousel, thread

Parameters:

- type : post, carousel, thread, story
- platform : linkedin, twitter, instagram, facebook
- topic , tone , length
- projectId
- direct : boolean (direct publish of draft)

Consolidates:

- generate-social-post/route.ts (421 lines)
- social-media-posts/generate/route.ts (368 lines)
- social-media-posts/generate-direct/route.ts (307 lines)
- social-media/generate-post/route.ts (169 lines)
- social/generate/route.ts (196 lines)

Code reuse: Beste implementatie als base, merge features

3. Ideas Management

```
GET /api/client/social/ideas
POST /api/client/social/ideas
DELETE /api/client/social/ideas/[id]
PATCH /api/client/social/ideas/[id]
```

Functionaliteit:

- GET: List all ideas
- POST: Create or generate new idea

- DELETE: Delete idea
- PATCH: Update idea status

Parameters:

- generate : boolean (AI generation if true)
- topic , keywords
- projectId

Consolidates:

- social-media-ideas/route.ts
 - social-media-ideas/generate/route.ts
 - social/ideas/route.ts
 - social/generate-ideas/route.ts
-

4. Topics Management

```
GET   /api/client/social/topics
POST  /api/client/social/topics
DELETE /api/client/social/topics/[id]
```

Functionaliteit:

- GET: List topics
- POST: Create or generate topics
- DELETE: Delete topic

Parameters:

- generate : boolean (AI generation)
- projectId

Consolidates:

- social-media-topics/route.ts
 - social-media-topics/generate/route.ts
-

5. Scheduling

```
GET   /api/client/social/schedules
POST  /api/client/social/schedules
```

Functionaliteit:

- GET: List all scheduled posts
- POST: Schedule a post

Parameters:

- postId , scheduledAt
- timezone , platform

Consolidates:

- social-media-posts/schedule/route.ts

- social-media/schedules/route.ts
 - social/schedule/route.ts
-

```
GET /api/client/social/schedules/[id]
PUT /api/client/social/schedules/[id]
DELETE /api/client/social/schedules/[id]
```

Functionaliteit:

- GET: Get schedule details
- PUT: Update schedule
- DELETE: Cancel schedule

Consolidates:

- social/schedule/[id]/route.ts
-

6. Publishing

```
POST /api/client/social/publish
```

Functionaliteit:

- Publish post to social media platforms
- Support voor meerdere platforms tegelijk

Parameters:

- postId or content (direct publish)
- platforms : array van platforms
- scheduleAt : optional timestamp

Consolidates:

- social-media-posts/publish/route.ts
 - social-media/publish/route.ts
-

7. Analytics

```
GET /api/client/social/analytics
```

Functionaliteit:

- Get social media analytics
- Per platform of aggregated

Parameters:

- platform : optional filter
- dateFrom , dateTo
- projectId
- metrics : impressions, engagements, clicks

Behouden: social/analytics/route.ts (141 lines)

8. Planning

```
POST /api/client/social/planning
```

Functionaliteit:

- Generate content planning/calendar
- AI-powered content strategy

Parameters:

- duration : week, month, quarter
- frequency : daily, weekly
- topics , keywords

Behouden: social-media/generate-planning/route.ts (174 lines)

9. Queue Management

```
GET /api/client/social/queue
PUT /api/client/social/queue
```

Functionaliteit:

- GET: View post queue
- PUT: Reorder queue

Behouden: social/queue/route.ts (161 lines)

10. Account Management

```
GET /api/client/social/accounts
POST /api/client/social/accounts
```

Functionaliteit:

- GET: List connected accounts (load accounts)
- POST: Save/update accounts configuration

Consolidates:

- social-media/load-accounts/route.ts
 - social-media/save-accounts/route.ts
-

```
POST /api/client/social/accounts/connect
DELETE /api/client/social/accounts/[id]
```

Functionaliteit:

- POST: Connect new account (with connection test)
- DELETE: Disconnect account

Parameters:

- platform , credentials
- testConnection : boolean

Consolidates:

- social-media/connect/route.ts
- social-media/link-account/route.ts
- social-media/test-connection/route.ts

11. Settings

```
GET   /api/client/social/settings
POST  /api/client/social/settings
```

Functionaliteit:

- GET: Get social media configuration
- POST: Update configuration (config + profile)

Consolidates:

- social-media/config/route.ts
- social-media/profile/route.ts

12. Automation

```
POST  /api/client/social/autopilot/setup
```

Functionaliteit:

- Auto-setup autopilot configuration

Behouden: social-media/auto-setup/route.ts (157 lines)

```
POST  /api/client/social/autopilot/run
```

Functionaliteit:

- Trigger autopilot run manually

Behouden: social-media/autopilot-run/route.ts (239 lines)

13. Team Management

```
GET /api/client/social/invites
POST /api/client/social/invites
DELETE /api/client/social/invites/[id]
```

Functionaliteit:

- GET: List invites
- POST: Create invite
- DELETE: Revoke invite

Consolidates:

- social-media/create-invite/route.ts
- social-media/invites/route.ts

Structuur Overzicht

```
/api/client/social/
  posts/
    route.ts          (GET, POST)
    [id]/route.ts    (GET, PUT, DELETE)
    bulk-delete/route.ts (POST)
  generate/route.ts (POST) - Unified generation
  ideas/
    route.ts          (GET, POST)
    [id]/route.ts    (DELETE, PATCH)
  topics/
    route.ts          (GET, POST)
    [id]/route.ts    (DELETE)
  schedules/
    route.ts          (GET, POST)
    [id]/route.ts    (GET, PUT, DELETE)
  publish/route.ts (POST)
  analytics/route.ts (GET)
  planning/route.ts (POST)
  queue/route.ts   (GET, PUT)
  accounts/
    route.ts          (GET, POST)
    connect/route.ts (POST)
    [id]/route.ts    (DELETE)
    settings/route.ts (GET, POST)
  autopilot/
    setup/route.ts   (POST)
    run/route.ts     (POST)
  invites/
    route.ts          (GET, POST)
    [id]/route.ts    (DELETE)
```

Totaal: 24 route files voor 14 hoofdfuncties

Migration Strategy

Fase 1: Creëer nieuwe geconsolideerde routes

1. Implementeer nieuwe routes met beste code van oude routes
2. Merge functionaliteit waar nodig
3. Test nieuwe routes

Fase 2: Backwards compatibility

1. Oude routes blijven bestaan maar redirecten naar nieuwe
2. Voeg deprecation warnings toe in responses
3. Update documentatie

Fase 3: Frontend migration

1. Scan alle frontend calls naar oude routes
2. Update naar nieuwe routes
3. Test alle functionaliteiten

Fase 4: Cleanup

1. Verwijder oude routes na frontend migration
2. Final tests
3. Deploy

Verwachte Resultaten

Metric	Voor	Na	Reductie
Route files	38	24	37%
Duplicate logic	~2000 lines	0	100%
Endpoints	38	14 hoofdfuncties	63%
Maintenance burden	Hoog	Laag	~60%

API Response Format (Gestandaardiseerd)

Alle responses volgen dit format:

```
// Success
{
  success: true,
  data: { ... },
  meta?: {
    total: number,
    page: number,
    limit: number
  }
}

// Error
{
  success: false,
  error: {
    code: string,
    message: string,
    details?: any
  }
}
```

Next Steps

1. Ontwerp voltooid
2. → Implementeer geconsolideerde routes (Fase 2.4)
3. → Update frontend references (Fase 2.5)
4. → Test & verify (Fase 2.6)
5. → Commit & document (Fase 2.7)