

WritGo API Documentation

Version: 2.0

Last Updated: 18 december 2024

Base URL: <https://writgo.nl/api>

Overview

WritGo API provides endpoints for content generation, management, and publishing. All API routes follow RESTful conventions with consistent error handling and authentication.

Authentication

All API endpoints require authentication via NextAuth session.

```
// Session format
{
  user: {
    email: string;
    name?: string;
    role?: 'admin' | 'client';
  }
}
```

Error Responses

All errors follow this format:

```
{
  "error": "ERROR_CODE",
  "message": "User-friendly error message",
  "details": "Technical details (optional)"
}
```

Status Codes:

- 200 - Success
- 400 - Bad Request (validation errors)
- 401 - Unauthorized (not logged in)
- 403 - Forbidden (insufficient permissions)
- 404 - Not Found
- 500 - Internal Server Error

Simplified API Routes

Account Management

GET /api/simplified/account

Get current user account information.

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "name": "User Name",
    "email": "user@example.com",
    "companyName": "Company Name",
    "website": "https://example.com",
    "automationActive": false,
    "subscriptionCredits": 1000,
    "topUpCredits": 500,
    "totalCredits": 1500,
    "createdAt": "2024-01-01T00:00:00Z"
  }
}
```

PATCH /api/simplified/account

Update account information.

Request Body:

```
{
  "name": "New Name",
  "companyName": "New Company",
  "website": "https://newsite.com",
  "targetAudience": "Target audience description",
  "brandVoice": "Brand voice description"
}
```

Response:

```
{
  "success": true,
  "message": "Account succesvol bijgewerkt",
  "data": { /* updated account */ }
}
```

Content Management

GET /api/simplified/content-overview

Get all content for the authenticated user.

Query Parameters:

- `status` (optional) - Filter by status: `all`, `draft`, `published`
- `projectId` (optional) - Filter by project ID
- `limit` (optional) - Limit number of results

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "title": "Article Title",
      "status": "published",
      "source": "wordpress",
      "projectName": "Project Name",
      "publishedDate": "2024-01-01T00:00:00Z",
      "createdAt": "2024-01-01T00:00:00Z"
    }
  ],
  "stats": {
    "total": 100,
    "generated": 50,
    "wordpress": 50,
    "published": 80,
    "draft": 20
  },
  "meta": {
    "total": 100,
    "filters": { "status": "all", "projectId": null, "limit": null }
  }
}
```

Platform Management

GET /api/simplified/platforms

Get all connected social media platforms.

Response:

```
[
  {
    "platform": "linkedin",
    "display_name": "LinkedIn",
    "username": "jouw-bedrijf",
    "connected": true,
    "is_enabled": true,
    "last_post_at": "2024-01-01T00:00:00Z",
    "posts_this_month": 12
  }
]
```

PUT /api/simplified/platforms

Update platform connection status.

Request Body:

```
{
  "platform": "linkedin",
  "enabled": true
}
```

Response:

```
{
  "success": true,
  "message": "Platform linkedin updated successfully"
}
```

GET /api/simplified/platforms/[id]

Get specific platform information.

Response:

```
{
  "success": true,
  "data": {
    "id": "linkedin",
    "platform": "linkedin",
    "display_name": "LinkedIn",
    "username": "jouw-bedrijf",
    "connected": true,
    "is_enabled": true
  }
}
```

PATCH /api/simplified/platforms/[id]

Update specific platform settings.

Request Body:

```
{
  "enabled": true,
  "username": "new-username"
}
```

DELETE /api/simplified/platforms/[id]

Disconnect a platform.

Response:

```
{
  "success": true,
  "message": "Platform linkedin disconnected successfully"
}
```

Dashboard & Statistics**GET /api/simplified/stats**

Get dashboard statistics for authenticated user.

Response:

```
{
  "totalProjects": 3,
  "contentThisMonth": 15,
  "publishedArticles": 42,
  "recentContent": [
    {
      "id": "uuid",
      "title": "Article Title",
      "status": "published",
      "publishedAt": "2024-01-01T00:00:00Z",
      "project": {
        "name": "Project Name"
      }
    }
  ]
}
```

GET /api/simplified/dashboard/projects

Get all projects for dashboard.

Response:

```
[
  {
    "id": "uuid",
    "name": "Project Name",
    "websiteUrl": "https://example.com",
    "isActive": true,
    "articleCount": 25,
    "lastPublished": "2024-01-01T00:00:00Z"
  }
]
```

Content Generation

POST /api/simplified/generate

Generate a new article.

Request Body:

```
{
  "topic": "Article topic",
  "projectId": "uuid",
  "searchIntent": "informational"
}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "title": "Generated Title",
    "content": "Generated content...",
    "metaDescription": "Meta description"
  }
}
```

POST /api/simplified/generate/quick

Quick generate content without full processing.

Request Body:

```
{
  "prompt": "Quick generation prompt",
  "type": "blog" || "social" || "email"
}
```

Publishing

POST /api/simplified/publish

Publish content to platform.

Request Body:

```
{
  "contentId": "uuid",
  "platform": "wordpress",
  "schedule": "2024-01-01T00:00:00Z" // optional
}
```

POST /api/simplified/publish/wordpress

Publish specifically to WordPress.

Request Body:

```
{
  "projectId": "uuid",
  "title": "Article Title",
  "content": "Article content",
  "status": "publish" || "draft",
  "categories": ["category1", "category2"]
}
```

Blog Management

GET /api/simplified/blog

Get all blog posts.

Query Parameters:

- `limit` (optional) - Number of posts to return
- `offset` (optional) - Pagination offset

Response:

```
{
  "posts": [
    {
      "id": "uuid",
      "title": "Blog Title",
      "slug": "blog-title",
      "excerpt": "Post excerpt",
      "publishedAt": "2024-01-01T00:00:00Z",
      "author": "Author Name"
    }
  ],
  "total": 50,
  "hasMore": true
}
```

GET /api/simplified/blog/[slug]

Get specific blog post by slug.

Response:

```
{
  "id": "uuid",
  "title": "Blog Title",
  "slug": "blog-title",
  "content": "Full blog content",
  "publishedAt": "2024-01-01T00:00:00Z",
  "author": {
    "name": "Author Name",
    "avatar": "https://..."
  }
}
```

Social Media

POST /api/simplified/social-media/generate

Generate social media post.

Request Body:

```
{
  "platform": "linkedin" || "facebook" || "twitter",
  "topic": "Post topic",
  "tone": "professional" || "casual" || "funny"
}
```

GET /api/simplified/social-media/history

Get social media post history.

Response:

```
{
  "posts": [
    {
      "id": "uuid",
      "platform": "linkedin",
      "content": "Post content",
      "publishedAt": "2024-01-01T00:00:00Z",
      "engagement": {
        "likes": 10,
        "comments": 5,
        "shares": 2
      }
    }
  ]
}
```

POST /api/simplified/social-media/post

Post to social media.

Request Body:

```
{
  "platform": "linkedin",
  "content": "Post content",
  "media": ["https://..."], // optional
  "schedule": "2024-01-01T00:00:00Z" // optional
}
```

GET /api/simplified/social-media/settings

Get social media settings.

PUT /api/simplified/social-media/settings

Update social media settings.

Projects

GET /api/simplified/projects

Get all projects.

Response:

```
[
  {
    "id": "uuid",
    "name": "Project Name",
    "websiteUrl": "https://example.com",
    "isActive": true,
    "createdAt": "2024-01-01T00:00:00Z"
  }
]
```

GET /api/simplified/projects/[id]

Get specific project.

Response:

```
{
  "id": "uuid",
  "name": "Project Name",
  "websiteUrl": "https://example.com",
  "isActive": true,
  "settings": {
    "wordpress": { /* wordpress config */ },
    "seo": { /* seo settings */ }
  }
}
```

Content Planning

GET /api/simplified/content-plan

Get content plan.

Response:

```
{
  "plan": [
    {
      "id": "uuid",
      "topic": "Topic",
      "scheduledFor": "2024-01-01T00:00:00Z",
      "status": "scheduled" || "published" || "draft"
    }
  ]
}
```

POST /api/simplified/content-plan/analyze-wordpress

Analyze WordPress site for content planning.

Request Body:

```
{
  "projectId": "uuid"
}
```

Rewriting

POST /api/simplified/rewrite

Rewrite existing content.

Request Body:

```
{  
  "content": "Content to rewrite",  
  "tone": "professional" || "casual",  
  "instructions": "Optional rewrite instructions"  
}
```

Rate Limiting

API endpoints are rate limited based on user plan:

- Instapper: 100 requests/hour
- Starter: 200 requests/hour
- Groei: 500 requests/hour
- Dominant: 1000 requests/hour

Webhooks

Coming soon.

Support

Voor vragen over de API, neem contact op via support@writgo.nl