# **Cyrus Blog Backend API 文档**

## **概述**

Cyrus Blog Backend 是一个基于 Rust Axum 框架构建的现代化博客系统后端服务。提供完整的博客管理、AI聊天助手、文件上传等功能。

### **技术栈**

• **框架**: Axum (异步Web框架)

• **数据库**: SQLite + SQLx

• **认证**: JWT Token + 管理员Token

• **AI集成**: DeepSeek API

• **文件存储**: 本地文件系统

### **服务器信息**

• **默认端口**: 3001

• **基础URL**: http://127.0.0.1:3001

• **API前缀**: /api

## **认证说明**

### **管理员认证**

管理员接口需要在请求头中包含认证Token：

Authorization: Bearer admin123456

### **公开接口**

博客查看、AI聊天等接口无需认证。

## **API 接口详情**

### **1. 健康检查接口**

#### **1.1 基础健康检查**

• **接口**: GET /api/health

• **描述**: 基础健康状态检查

• **认证**: 无需认证

**响应示例**:

**json**

{

  "status": "healthy",

  "timestamp": "2024-01-01T12:00:00Z",

  "service": "cyrus-blog-backend"

}

#### **1.2 详细健康检查**

• **接口**: GET /api/health/detailed

• **描述**: 详细的系统健康状态检查

• **认证**: 无需认证

**响应示例**:

**json**

{

  "status": "healthy",

  "timestamp": "2024-01-01T12:00:00Z",

  "version": "0.1.0",

  "uptime\_seconds": 3600,

  "checks": {

    "database": {

      "status": "pass",

      "response\_time\_ms": 5,

      "message": "Database connection healthy",

      "details": {

        "pool\_size": 10,

        "idle\_connections": 8,

        "utilization\_percent": 20.0

      }

    },

    "memory": {

      "status": "pass",

      "response\_time\_ms": 1,

      "message": "Memory usage: 100.00 MB",

      "details": {

        "memory\_usage\_mb": 100.0,

        "threshold\_warn\_mb": 1000.0,

        "threshold\_fail\_mb": 2000.0

      }

    },

    "disk": {

      "status": "pass",

      "response\_time\_ms": 2,

      "message": "Disk usage: 10.0%",

      "details": {

        "usage\_percent": 10.0,

        "used\_bytes": 1073741824,

        "total\_bytes": 10737418240

      }

    },

    "external\_services": {

      "status": "pass",

      "response\_time\_ms": 1,

      "message": "External services check skipped"

    }

  },

  "metrics": {

    "memory\_usage\_mb": 100.0,

    "cpu\_usage\_percent": 0.0,

    "active\_connections": 10,

    "request\_count": 0,

    "cache\_hit\_rate": 0.0,

    "avg\_response\_time\_ms": 0

  }

}

#### **1.3 就绪检查**

• **接口**: GET /api/health/ready

• **描述**: Kubernetes就绪检查

• **认证**: 无需认证

#### **1.4 存活检查**

• **接口**: GET /api/health/live

• **描述**: Kubernetes存活检查

• **认证**: 无需认证

### **2. 博客文章接口 (公开)**

#### **2.1 获取文章列表**

• **接口**: GET /api/posts

• **描述**: 获取已发布的文章列表（分页）

• **认证**: 无需认证

**查询参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 类型 | 必需 | 默认值 | 描述 |
| page | integer | 否 | 1 | 页码 |
| limit | integer | 否 | 10 | 每页数量 |
| q | string | 否 | - | 搜索关键词 |

**响应示例**:

**json**

{

  "success": true,

  "data": {

    "posts": [

      {

        "id": 1,

        "title": "我的第一篇博客",

        "excerpt": "这是文章摘要...",

        "slug": "my-first-blog",

        "date": "2024-01-01",

        "categories": ["技术", "Rust"],

        "created\_at": "2024-01-01T12:00:00Z",

        "updated\_at": "2024-01-01T12:00:00Z"

      }

    ],

    "totalPosts": 1,

    "totalPages": 1

  }

}

#### **2.2 根据slug获取文章详情**

• **接口**: GET /api/posts/{slug}

• **描述**: 根据文章slug获取完整文章内容

• **认证**: 无需认证

**路径参数**:

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必需 | 描述 |
| slug | string | 是 | 文章的唯一标识符 |

**响应示例**:

**json**

{

  "success": true,

  "data": {

    "id": 1,

    "title": "我的第一篇博客",

    "excerpt": "这是文章摘要...",

    "content": "# 标题\n\n这是完整的文章内容...",

    "slug": "my-first-blog",

    "date": "2024-01-01",

    "categories": ["技术", "Rust"],

    "created\_at": "2024-01-01T12:00:00Z",

    "updated\_at": "2024-01-01T12:00:00Z"

  }

}

#### **2.3 获取分类列表**

• **接口**: GET /api/categories

• **描述**: 获取所有文章分类

• **认证**: 无需认证

**响应示例**:

**json**

{

  "success": true,

  "data": ["技术", "生活", "Rust", "前端"]

}

#### **2.4 根据分类获取文章**

• **接口**: GET /api/categories/{category}

• **描述**: 获取指定分类下的所有文章

• **认证**: 无需认证

**路径参数**:

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必需 | 描述 |
| category | string | 是 | 分类名称 |

### **3. AI聊天接口**

#### **3.1 AI聊天**

• **接口**: POST /api/chat

• **描述**: 与AI助手进行对话

• **认证**: 无需认证

**请求体**:

**json**

{

  "message": "你好，请介绍一下这个博客",

  "conversationHistory": [

    {

      "id": "msg-1",

      "content": "之前的消息",

      "isUser": true,

      "timestamp": "2024-01-01T12:00:00Z"

    }

  ]

}

**响应示例**:

**json**

{

  "success": true,

  "data": {

    "response": "你好！这是Cyrus的个人博客，主要分享技术文章和生活感悟..."

  }

}

### **4. 管理员接口 (需要认证)**

#### **4.1 Token验证**

• **接口**: GET /api/admin/verify

• **描述**: 验证管理员Token是否有效

• **认证**: 需要管理员Token

**响应示例**:

**json**

{

  "success": true

}

#### **4.2 获取仪表板数据**

• **接口**: GET /api/admin/dashboard

• **描述**: 获取管理员仪表板统计数据

• **认证**: 需要管理员Token

**响应示例**:

**json**

{

  "totalPosts": 10,

  "totalCategories": 5,

  "recentPosts": [...],

  "popularCategories": [...]

}

#### **4.3 获取所有文章 (管理员)**

• **接口**: GET /api/admin/posts

• **描述**: 获取所有文章（包括草稿）

• **认证**: 需要管理员Token

**查询参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 类型 | 必需 | 默认值 | 描述 |
| page | integer | 否 | 1 | 页码 |
| limit | integer | 否 | 10 | 每页数量 |

#### **4.4 创建文章**

• **接口**: POST /api/admin/posts

• **描述**: 创建新文章

• **认证**: 需要管理员Token

**请求体**:

**json**

{

  "title": "新文章标题",

  "excerpt": "文章摘要",

  "content": "# 文章内容\n\n这是文章的完整内容...",

  "slug": "new-article-slug",

  "date": "2024-01-01",

  "categories": ["技术", "Rust"]

}

**字段说明**:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 必需 | 描述 |
| title | string | 是 | 文章标题 |
| excerpt | string | 是 | 文章摘要 |
| content | string | 是 | 文章内容（Markdown格式） |
| slug | string | 否 | URL标识符（自动生成） |
| date | string | 否 | 发布日期（ISO格式，默认当前日期） |
| categories | array | 是 | 文章分类数组 |

#### **4.5 获取文章详情 (管理员)**

• **接口**: GET /api/admin/posts/{slug}

• **描述**: 获取指定文章的完整信息

• **认证**: 需要管理员Token

#### **4.6 更新文章**

• **接口**: PUT /api/admin/posts/{slug}

• **描述**: 更新指定文章

• **认证**: 需要管理员Token

**请求体** (所有字段都是可选的):

**json**

{

  "title": "更新后的标题",

  "excerpt": "更新后的摘要",

  "content": "更新后的内容",

  "slug": "new-slug",

  "date": "2024-01-02",

  "categories": ["技术", "更新"]

}

#### **4.7 删除文章**

• **接口**: DELETE /api/admin/posts/{slug}

• **描述**: 删除指定文章

• **认证**: 需要管理员Token

#### **4.8 获取文章Markdown内容**

• **接口**: GET /api/admin/posts/{slug}/markdown

• **描述**: 获取文章的原始Markdown内容用于编辑

• **认证**: 需要管理员Token

**响应示例**:

**json**

{

  "success": true,

  "content": "# 文章标题\n\n这是文章的Markdown内容..."

}

#### **4.9 获取分类列表 (管理员)**

• **接口**: GET /api/admin/categories

• **描述**: 获取所有分类

• **认证**: 需要管理员Token

#### **4.10 AI助手**

• **接口**: POST /api/admin/ai-assist

• **描述**: 管理员AI助手功能

• **认证**: 需要管理员Token

**请求体**:

**json**

{

  "prompt": "帮我写一篇关于Rust的文章",

  "type": "content\_generation",

  "deepseekApiKey": "可选的API密钥",

  "deepseekModel": "可选的模型名称"

}

#### **4.11 图片上传**

• **接口**: POST /api/admin/upload/image

• **描述**: 上传图片文件

• **认证**: 需要管理员Token

• **Content-Type**: multipart/form-data

**请求参数**:

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必需 | 描述 |
| image | file | 是 | 图片文件 (支持 jpg, jpeg, png, webp, gif) |

**文件限制**:

• 最大文件大小: 10MB

• 支持格式: JPG, JPEG, PNG, WebP, GIF

**响应示例**:

**json**

{

  "success": true,

  "url": "/uploads/images/1704110400-123456789.jpg",

  "filename": "1704110400-123456789.jpg",

  "size": 1024000

}

#### **4.12 系统状态**

• **接口**: GET /api/admin/system-status

• **描述**: 获取系统状态信息

• **认证**: 需要管理员Token

**响应示例**:

**json**

{

  "success": true,

  "data": {

    "serverStatus": "online",

    "databaseStatus": "connected",

    "storageUsage": 10,

    "lastUpdated": "2024-01-01T12:00:00Z",

    "uptime": 3600,

    "version": "0.1.0",

    "metrics": {...}

  }

}

#### **4.13 统计趋势**

• **接口**: GET /api/admin/stats-trends

• **描述**: 获取统计趋势数据

• **认证**: 需要管理员Token

## **错误响应格式**

所有API在发生错误时都会返回统一的错误格式：

**json**

{

  "success": false,

  "error": {

    "code": "VALIDATION\_ERROR",

    "message": "详细的错误描述",

    "details": "可选的额外错误信息"

  }

}

### **常见错误码**

• VALIDATION\_ERROR: 请求参数验证失败

• NOT\_FOUND: 资源不存在

• UNAUTHORIZED: 未授权访问

• INTERNAL\_ERROR: 服务器内部错误

• DATABASE\_ERROR: 数据库操作失败

## **数据验证规则**

### **文章相关**

• **标题**: 1-200字符，不能为空

• **摘要**: 1-500字符，不能为空

• **内容**: 1-50000字符，不能为空

• **Slug**: 字母、数字、连字符，3-100字符

• **分类**: 每个分类1-50字符，最多10个分类

### **搜索和分页**

• **页码**: 最小值1，最大值1000

• **每页数量**: 最小值1，最大值100

• **搜索关键词**: 最大100字符

### **AI聊天**

• **消息内容**: 1-2000字符，不能为空

• **AI提示**: 1-5000字符，不能为空

## **静态文件服务**

上传的文件可通过以下URL访问：

• **图片**: GET /uploads/images/{filename}

## **配置说明**

服务器配置文件 config.toml 包含以下主要配置：

**toml**

[server]

host = "127.0.0.1"

port = 3001

[database]

url = "sqlite:./data/blog.db"

[auth]

jwt\_secret = "your-jwt-secret-key-here"

admin\_token = "admin123456"

[ai]

deepseek\_api\_key = "your-deepseek-api-key-here"

deepseek\_api\_url = "https://api.deepseek.com/v1/chat/completions"

[cors]

allowed\_origins = ["http://localhost:3000", "http://127.0.0.1:3000"]

allowed\_methods = ["GET", "POST", "PUT", "DELETE", "OPTIONS"]

allowed\_headers = ["Content-Type", "Authorization"]

[storage]

upload\_dir = "./uploads"

max\_file\_size = 10485760 # 10MB

## **数据库结构**

### **blog\_posts 表**

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| id | INTEGER | 主键，自增 |
| title | TEXT | 文章标题 |
| excerpt | TEXT | 文章摘要 |
| content | TEXT | 文章内容 |
| slug | TEXT | URL标识符（唯一） |
| date | TEXT | 发布日期（ISO格式） |
| categories | TEXT | 分类（JSON数组字符串） |
| created\_at | DATETIME | 创建时间 |
| updated\_at | DATETIME | 更新时间 |

### **其他表**

• categories: 分类表（预留）

• post\_categories: 文章分类关联表（预留）

• users: 用户表（预留）

• chat\_sessions: 聊天会话表（可选）

• chat\_messages: 聊天消息表（可选）

## **部署说明**

1. **环境要求**: Rust 1.70+

2. **数据库**: 自动创建SQLite数据库文件

3. **端口**: 默认3001，可通过环境变量 PORT 修改

4. **文件存储**: 确保 uploads 目录有写权限

## **开发和测试**

### **启动服务器**

**bash**

cd backend

cargo run

### **运行测试**

**bash**

cargo test

### **构建生产版本**

**bash**

cargo build --release

**文档版本**: v1.0  
**最后更新**: 2024-01-01  
**API版本**: v1