WallMagazine API Documentation

Base URL

```
Development: http://localhost:3000
Production: https://your-domain.com
```

Authentication

All protected endpoints require session cookie authentication.

http

```http
Cookie: session_token=<jwt_token>
```

Posts Management

1. Create Post Submission

POST `/api/posts`

Permission: `create_post` (student+)

Body (FormData):

typescript

```
{
  title: string;              // Required, 3-200 chars
  category: "article" | "poem" | "artwork" | "notice";
  submission_type: "upload" | "paste" | "image_upload";
  tags: string;               // Comma-separated

  // If submission_type === "paste"
  raw_content: string;        // Max 50KB

  // If submission_type === "upload"
  file: File;                 // .docx, .pdf, .txt (max 16MB)

  // If submission_type === "image_upload"
  images: File[];             // .jpg, .png (max 10MB each, up to 10)
}
```

Response:

json

```json
{
  "message": "Post submitted successfully",
  "post": {
    "id": "507f1f77bcf86cd799439011",
    "title": "My Article",
    "status": "PENDING_REVIEW",
    "category": "article"
  }
}
```

---

2. List Posts (Public)

GET `/api/posts`

Query Params:

- `status` - Filter by status (default: PUBLISHED)
- `category` - Filter by category
- `author` - Filter by author ID
- `page` - Page number (default: 1)
- `limit` - Items per page (default: 10)

Response:

json

```json
{
  "posts": [...],
  "pagination": {
    "page": 1,
    "limit": 10,
    "total": 45,
    "pages": 5
  }
}
```

---

3. Get User's Own Posts

GET `/api/posts/my-posts`

Permission: Authenticated user

Query Params: `status`, `category`, `page`, `limit`

Response:

json

```json
{
  "posts": [...],
  "pagination": {...},
  "stats": {
    "pending": 2,
    "accepted": 1,
    "rejected": 1,
    "published": 3,
    "total": 7
  }
}
```

---

4. Get Single Post

GET `/api/posts/[id]`

Access:

- Public: Only PUBLISHED posts
- Authenticated: Own posts + PUBLISHED

Response:

json

```json
{
  "post": {
    "_id": "...",
    "title": "...",
    "status": "PUBLISHED",
    "author": {...},
    "designed_files": [...],
    "views": 123,
    "likes": ["user_id_1", "user_id_2"]
  }
}
```

---

5. Update Post

PATCH `/api/posts/[id]`

Permission: Author only (status: PENDING_REVIEW or REJECTED)

Body:

json

```json
{
  "title": "Updated Title",
  "tags": ["tag1", "tag2"],
  "raw_content": "Updated content" // If paste type
}
```

---

6. Delete Post

DELETE `/api/posts/[id]`

Permission:

- Author: PENDING_REVIEW or REJECTED only
- Admin: Any status

Response:

json

```json
{
  "message": "Post deleted successfully"
}
```

Editorial Workflow

7. Download Original File

GET `/api/posts/[id]/download`

Permission: `download_original_files` (editor/admin)

Query Params:

- `type=image` - Download original image
- `imageIndex=0` - Image index (if multiple)

Response: File stream

---

8. Review Post (Accept/Reject)

POST `/api/posts/[id]/review`

Permission: `accept_reject_submissions` (editor/admin)

Body:

json

```
// Accept
{
  "action": "accept"
}

// Reject
{
  "action": "reject",
  "rejection_reason": "Does not meet quality standards"
}
```

---

9. Upload Designed Files

POST `/api/posts/[id]/design`

Permission: `upload_designed_version` (editor/admin)

Body (FormData):

typescript

```typescript
{
  designs: File[];  // Images (.jpg, .png) or Documents (.pdf, .docx, .odt)
                    // Max 20 files, 20MB each
}
```

Response:

json

```json
{
  "message": "Designed files uploaded successfully",
  "post": {
    "id": "...",
    "status": "AWAITING_ADMIN",
    "designed_files_count": 3
  }
}
```

---

10. Admin Approve/Reject Design

POST `/api/posts/[id]/approve`

Permission: `approve_designs` (admin)

Body:

json

```json
// Approve
{
  "action": "approve"
}

// Reject
{
  "action": "reject",
  "rejection_reason": "Design needs improvement"
}
```

---

11. Publish Post

POST `/api/posts/[id]/publish`

Permission: `publish_post` (admin/publisher)

Body:

json

```
// Publish
{
  "action": "publish",
  "featured_until": "2025-02-15T00:00:00Z" // Optional
}

// Unpublish
{
  "action": "unpublish"
}
```

PATCH `/api/posts/[id]/publish` - Update featured status

Dashboards

12. Editor Dashboard

GET `/api/posts/editor/pending`

Permission: `view_pending_submissions` (editor/admin)

Query Params:

- `status` - PENDING_REVIEW, ACCEPTED, DESIGNING, ADMIN_REJECTED
- `category`
- `page`, `limit`

Response:

json

```
{
  "posts": [...],
  "pagination": {...},
  "stats": {
    "pending_review": 5,
    "accepted": 3,
    "admin_rejected": 2
  },
  "category_stats": {
    "article": 3,
    "poem": 2
  }
}
```

---

13. Admin Dashboard

GET `/api/posts/admin/awaiting`

Permission: `approve_designs` (admin)

Response:

json

```
{
  "posts": [...],
  "stats": {
    "awaiting_admin": 8,
    "approved": 5,
    "published": 42
  },
  "recent_published": [...]
}
```

Files

14. View/Stream File

GET `/api/posts/files/[fileId]`

Access: Public for PUBLISHED posts

Response: File stream (image/PDF inline, others as download)

User Management

15. Assign Role

POST `/api/admin/users/assign-role`

Permission: `assign_editors` (admin)

Body:

json

```
{
  "id_number": "PHY2023001",
  "role": "editor" // student, Professor, editor, publisher, admin
}
```

---

16. Search Users

GET `/api/admin/users/assign-role`

Permission: `manage_users` (admin)

Query Params:

- `q` - Search name/email/ID
- `role` - Filter by role
- `page`, `limit`

Response:

json

```json
{
 "users": [...],
 "role_stats": {
  "student": 45,
  "editor": 5,
  "admin": 1
 }
}
```

Analytics

17. Platform Analytics

GET `/api/admin/analytics`

Permission: `view_analytics` (admin)

Query Params:

- `period` - Days to analyze (default: 30)

Response:

json

```
{
  "overview": {
    "total_posts": 150,
    "published_posts": 42,
    "total_users": 53
  },
  "engagement": {
    "total_views": 5432,
    "total_likes": 876,
    "avg_views_per_post": 129
  },
  "top_posts": {
    "most_viewed": [...],
    "most_liked": [...]
  },
  "top_contributors": [...],
  "editor_performance": [...],
  "posts_over_time": [...]
}
```

Security Features

Rate Limiting

- **Signup:** 60 requests/hour per IP
- **Login:** 20 requests/hour per IP
- **Post creation:** Authenticated users only
- **File uploads:** Size limits enforced

Input Validation

- All user input sanitized
- File type verification
- SQL/NoSQL injection prevention
- XSS protection

Authentication

- JWT with httpOnly cookies
- Session blacklisting
- Token expiry (7 days)
- Redis-based session storage

⚡ Performance

Caching Strategy

- Profile data: 1 hour TTL
- Featured posts: 30 min TTL
- Statistics: 1 hour TTL

Database Indexes

- Status + created_at
- Author + created_at
- Category + published_at
- Tags search
- Text search on title/content

Workflow Summary

```
Student → PENDING_REVIEW
   ↓
Editor → ACCEPTED / REJECTED
   ↓
Editor → Upload Design → AWAITING_ADMIN
   ↓
Admin → APPROVED / ADMIN_REJECTED
   ↓
Admin/Publisher → PUBLISHED ✅
```

Status Codes

- `200` - Success
- `201` - Created
- `400` - Bad Request
- `401` - Unauthorized
- `403` - Forbidden
- `404` - Not Found
- `429` - Too Many Requests
- `500` - Server Error