Hashtag API

Author: Josh, Lead Backend Developer **Last Updated**: 2026-03-02 **Version**: v1.0

Base URL: `https://api.nexgate.com/api/v1`

Short Description: The Hashtag API enables content discovery through hashtags. Users can browse posts by hashtag with optional filters for products, shops, and events, and explore trending hashtags across configurable time windows.

Hints:

- Hashtags are case-insensitive — `#Dar` and `#dar` resolve to the same tag
- All hashtag data is extracted automatically from post content on publish
- Trending is calculated by post count within the selected time window

---

Standard Response Format

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Operation completed successfully",
  "action_time": "2025-09-23T10:30:45",
  "data": {}
}
```

HTTP Method Badge Standards

- **GET** - `GET` - Green

Endpoints

---

1. Get Posts By Hashtag

Purpose: Retrieve paginated posts that contain a specific hashtag, with optional filtering by attachment type

Endpoint: `GET` `{base_url}/e-social/hashtags/{hashtag}/posts`

Access Level: □□ Protected (Requires Bearer Token Authentication)

Path Parameters:

Parameter	Type	Required	Description
hashtag	string	Yes	The hashtag to search (without `#`, e.g. `MbeyaFashion`)

Query Parameters:

Parameter	Type	Required	Default	Description
filter	string	No	`ALL`	Filter by attachment type. Accepted values: `ALL`, `PRODUCTS`, `SHOPS`, `EVENTS`
page	integer	No	`1`	Page number
size	integer	No	`20`	Number of posts per page

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	`Bearer <token>`
Content-Type	string	Yes	`application/json`

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Posts retrieved",
  "action_time": "2026-03-02T10:30:45",
  "data": {
    "content": [
```

```
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "author": {
    "id": "123e4567-e89b-12d3-a456-426614174000",
    "userName": "kibuti_dev",
    "firstName": "Kibuti",
    "lastName": "Dev",
    "profilePictureUrl": "https://cdn.nexgate.com/profiles/kibuti.jpg",
    "isVerified": false
  },
  "content": "New collection just dropped! #MbeyaFashion #Summer",
  "contentParsed": {
    "text": "New collection just dropped! #MbeyaFashion #Summer",
    "entities": [
      {
        "type": "HASHTAG",
        "text": "#MbeyaFashion",
        "startIndex": 29,
        "endIndex": 42,
        "hashtag": "mbeyafashion"
      }
    ]
  },
  "postType": "REGULAR",
  "status": "PUBLISHED",
  "attachments": {
    "products": [
      {
        "id": "789e0123-e89b-12d3-a456-426614174000",
        "name": "Summer Dress",
        "price": 45000.00,
        "imageUrl": "https://cdn.nexgate.com/products/dress.jpg",
        "shopName": "Mbeya Styles",
        "shopId": "abc12345-e89b-12d3-a456-426614174000",
        "inStock": true,
        "socialContext": null
      }
    ],
    "shops": [],
    "events": [],
```

```
          "buyTogetherGroups": [],
          "installmentPlans": [],
          "externalLink": null
        },
        "engagement": {
          "likesCount": 24,
          "commentsCount": 5,
          "repostsCount": 3,
          "bookmarksCount": 8,
          "viewsCount": 120,
          "sharesCount": 2,
          "quotesCount": 1
        },
        "publishedAt": "2026-03-01T14:22:00",
        "createdAt": "2026-03-01T14:20:00"
      }
    ],
    "pageable": {
      "pageNumber": 0,
      "pageSize": 20
    },
    "totalElements": 54,
    "totalPages": 3,
    "last": false,
    "first": true
  }
}
```


Filter Behavior:

Filter Value	Returns
`ALL`	All posts containing the hashtag
`PRODUCTS`	Only posts with the hashtag that have attached products
`SHOPS`	Only posts with the hashtag that have attached shops
`EVENTS`	Only posts with the hashtag that have attached events

Standard Error Types:

- `401 UNAUTHORIZED`: Missing or invalid token
- `500 INTERNAL_SERVER_ERROR`: Unexpected server error

Error Response Example:

Unauthorized (401):

```
{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Token has expired",
  "action_time": "2026-03-02T10:30:45",
  "data": "Token has expired"
}
```

---

2. Get Trending Hashtags

Purpose: Retrieve the most used hashtags within a given time window, ordered by post count descending

Endpoint: `GET` `{base_url}/e-social/hashtags/trending`

Access Level: □□ Protected (Requires Bearer Token Authentication)

Query Parameters:

Parameter	Type	Required	Default	Description
window	string	No	`24h`	Time window for trending calculation. Accepted values: `24h`, `7d`, `30d`
limit	integer	No	`15`	Maximum number of trending hashtags to return

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	`Bearer <token>`
Content-Type	string	Yes	`application/json`

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Trending hashtags",
  "action_time": "2026-03-02T10:30:45",
  "data": [
    {
      "hashtag": "mbeyafashion",
      "postCount": 312
    },
    {
      "hashtag": "daressalaamtech",
    "postCount": 278
    },
    {
      "hashtag": "tanzaniastartups",
      "postCount": 195
    },
    {
      "hashtag": "summer2026",
      "postCount": 143
    }
  ]
}
```


Response Fields:

Field	Type	Description
hashtag	string	The hashtag text (always lowercase, without `#`)
postCount	long	Number of posts using this hashtag within the selected time window

Window Behavior:

Window Value	Looks Back
`24h`	Last 24 hours — what's trending right now
`7d`	Last 7 days — weekly trends
`30d`	Last 30 days — monthly trends

Standard Error Types:

- `401 UNAUTHORIZED`: Missing or invalid token
- `500 INTERNAL_SERVER_ERROR`: Unexpected server error

---

3. Search Hashtags

Purpose: Search hashtags by prefix query, used in post composer to suggest similar hashtags while typing. Results are sorted by a weighted score combining recent activity and all-time popularity.

Endpoint: `GET` `{base_url}/e-social/hashtags/search`

Access Level: Protected (Requires Bearer Token Authentication)

Query Parameters:

Parameter	Type	Required	Default	Description
q	string	Yes	—	Prefix query string (e.g. `Mbe` matches `mbeyafashion`, `mbeyatech`)
limit	integer	No	`5`	Maximum number of suggestions to return

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	`Bearer <token>`
Content-Type	string	Yes	`application/json`

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Hashtag suggestions",
  "action_time": "2026-03-02T10:30:45",
  "data": [
    {
      "hashtag": "mbeyafashion",
```

```
      "postCount": 312
    },
    {
      "hashtag": "mbeyatech",
      "postCount": 89
    },
    {
      "hashtag": "mbeyafoods",
      "postCount": 45
    }
  ]
}
```

Response Fields:

Field	Type	Description
hashtag	string	Matched hashtag text (always lowercase, without `#`)
postCount	long	Total number of posts using this hashtag

Scoring & Sort Logic:

Results are sorted by a weighted score combining recency and popularity:

Post Age	Score Weight
Posted within last 7 days	**2 points** per post
Older than 7 days	**1 point** per post

So a hashtag with 50 recent posts can outrank one with 200 old posts — fresh and trending tags float to the top naturally.

Empty Query Behavior: If `q` is empty or blank, returns an empty list immediately without querying the database.

Standard Error Types:

- `401 UNAUTHORIZED`: Missing or invalid token
- `500 INTERNAL_SERVER_ERROR`: Unexpected server error

Error Response Example:

Unauthorized (401):

```
{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Token has expired",
  "action_time": "2026-03-02T10:30:45",
  "data": "Token has expired"
}
```

Quick Reference Guide

Endpoints Summary

Method	Endpoint	Description
GET	`/e-social/hashtags/{hashtag}/posts`	Get posts by hashtag with optional filter
GET	`/e-social/hashtags/trending`	Get trending hashtags by time window

Filter Values

`ALL` `PRODUCTS` `SHOPS` `EVENTS`

Trending Window Values

`24h` `7d` `30d`

Authentication

- **Bearer Token**: Include `Authorization: Bearer your_token` in all headers

Data Format Standards

- **Dates**: ISO 8601 format (`2026-03-02T14:30:00`)
- **IDs**: UUID format (`550e8400-e29b-41d4-a716-446655440000`)
- **Hashtags**: Always lowercase, without `#` symbol in API responses

---

Revision #2
Created 2 March 2026 08:44:38 by Admin Qbit
Updated 2 March 2026 08:53:38 by Admin Qbit