

Amended Endpoints — what the feed release changes in endpoints you already have

Author: Josh S. Sakweli, Backend Lead Team

Last Updated: 2026-09-18

Version: v1.0 (backend branch `fet/feed_baking`)

Base URL: `https://dev.api.nexgate.co` (staging) · `https://api.nexgate.co` (production)

Short Description: This document lists **only the endpoints that already existed and have now changed**. Nothing here is a new URL. Every path below is one your app already calls today, and every entry says exactly what moved under it, whether the change is visible in the response, and whether you have to do anything about it.

“ Its sibling document is `09_NEW_ENDPOINTS.md` — the endpoints and controllers that did not exist before (home feed, reels, marketplace, events, interactions, and the admin/ops APIs). If a path is not in this file, it was not amended; if it is not in that file, it is not new.

Hints:

- **No path, verb or field was removed.** Every change is additive, a bug fix, or a stricter parameter.
- Two response fields were **added** (both on `ClipResponse`). An app that ignores them behaves exactly as it does today.
- Three endpoints **return fewer items than before for some viewers**. That is the point: they were leaking private content.
- Four old paths are **being deleted, not migrated from** — see [Old endpoints being deleted](#). There is no production data and no grace period: when the frontend has switched, they go.
- Search this file by path. Every heading is the literal URL.

How to read this document

Each entry carries two chips.

Change type — what kind of amendment it is:

Chip	Meaning
SECURITY FIX	The endpoint used to return content the viewer was not allowed to see
RESPONSE	A field was added, or a field that used to be empty is now filled
BEHAVIOUR	Same shape, different contents, ordering or side effect
PARAMS	A query parameter is now validated or clamped
BEING DELETED	Still answers today, but is scheduled for removal — do not build on it

App action — what the frontend must do:

Chip	Meaning
None	Ships safely with no app change
Optional	Works untouched, but there is something worth picking up
Required	The app must handle this or it will misbehave

Index

#	Endpoint	Change type	App action
1	GET /api/v1/e-social/feed	RESPONSE · BEING DELETED	Required
2	GET /api/v1/e-social/clip/feed	SECURITY FIX · RESPONSE · BEHAVIOUR	Required
3	GET /api/v1/e-social/clip/{mediaId}	SECURITY FIX · RESPONSE	Optional
4	GET /api/v1/e-social/clip/user/{userId}/videos	SECURITY FIX · RESPONSE · BEHAVIOUR	Required
5	POST /api/v1/e-social/clip/{mediaId}/view	BEHAVIOUR	Optional
6	GET /api/v1/e-social/hashtags/{hashtag}/posts	SECURITY FIX · PARAMS · BEHAVIOUR · RESPONSE	None
7	GET /api/v1/e-social/hashtags/trending	BEHAVIOUR · PARAMS	None

#	Endpoint	Change type	App action
8	GET /api/v1/e-social/hashtags/search	BEHAVIOUR · PARAMS	☐ None

Standard Response Format

Unchanged. Every endpoint below still answers in the usual envelope:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Operation completed successfully",
  "action_time": "2026-09-18T10:30:45",
  "data": { }
}
```

Errors are unchanged too: 400 bad request, 401 token problems, 403 permission, 404 not found, 422 validation, 500 server. Where an amendment introduces a *new* reason for an existing status, the entry says so.

Endpoints

1. Home feed (old)

Endpoint: GET /api/v1/e-social/feed

Change type: RESPONSE BEING DELETED · **App action:** ☐ Required — move to /api/v1/feed/home

“ **This endpoint is going away.** It is listed here because it still answers today and its contents changed. Do not build anything new on it.

Access Level: ☐ Protected · **Authentication:** Authorization: Bearer <access token>

What changed

a) `topComments` is now always populated.

This endpoint used to call the post mapper with "no top comments", so `topComments` came back empty on every post. The mapper now always fills it (up to 5, most-liked first) because the feed needs it and loading it costs nothing extra in the batched mapper.

b) A backend switch can make it answer from the new pipeline. There is a `feed.legacy.fromPipeline` setting that makes this endpoint serve the new feed in the old shape. It was built for a migration that is no longer needed, and it is **not part of the plan** — it will be deleted together with the endpoint. Mentioned only so nobody is surprised to find it in the code.

Response shape (unchanged)

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Feed retrieved",
  "action_time": "2026-09-18T10:30:45",
  "data": {
    "posts": [ { "postId": "...", "topComments": [ { } ] } ],
    "unseenCount": 12,
    "nextCursor": "eyJ0IjoxNzU4..."
  }
}
```

What the app must do

Move to `GET /api/v1/feed/home`. See `09_NEW_ENDPOINTS.md` A1 and `07_FEED_ENDPOINTS_GUIDE.md`. This path is deleted once you have.

Until then it keeps working as it does today.

2. Clip feed

Endpoint: `GET /api/v1/e-social/clip/feed`

Change type: `SECURITY FIX` `RESPONSE` `BEHAVIOUR` · **App action:** `Required`

Access Level: `Protected` · **Authentication:** `Authorization: Bearer <access token>`

What changed

a) It no longer leaks private videos. The feed used to return clips from followers-only and mentions-only posts to anyone, and clips from accounts the viewer had blocked. It now filters by the post's visibility, the viewer's follow relationship, mentions, and blocks — in the database, before paging.

b) Moderation labels apply. A clip whose post or author carries a moderation label is either dropped from the page or marked for an interstitial, per the rules in `01_ARCHITECTURE.md` §9.2. Followers are treated more leniently than strangers.

c) Two new fields in each clip (see the table below).

d) A page can be shorter than `limit` while `hasMore` is still `true`. The cursor is taken from the last row the database returned, *before* labels remove anything. So `clips.length < limit` no longer means "end of feed" — only `hasMore === false` and `nextCursor === null` mean that.

e) `limit` is clamped to 1-20 (it was previously used as given).

Query Parameters

Parameter	Type	Required	Description	Validation	Default
<code>cursor</code>	string	No	Opaque page cursor from the previous response	Invalid cursor → <code>400</code>	none (first page)
<code>limit</code>	integer	No	Clips per page	Clamped to 1-20 (new)	10

New response fields

Field	Type	Values	Description
<code>visibility</code>	string	<code>ALLOW</code> , <code>INTERSTITIAL</code>	<code>INTERSTITIAL</code> means: do not autoplay. Show a "sensitive content — tap to view" cover first. Defaults to <code>ALLOW</code>
<code>media.durationMs</code>	integer	milliseconds	Clip length. Useful for the progress bar, and needed to report watch time correctly in the interactions batch

Success Response JSON Sample

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Clip feed fetched",
  "action_time": "2026-09-18T10:30:45",
  "data": {
    "clips": [
      {
        "postId": "4f2a77e1-0c8b-4d2f-9a11-7b6c3e5d8f90",
        "visibility": "ALLOW",
        "media": {
          "id": "9b1c2d3e-4f5a-6b7c-8d9e-0f1a2b3c4d5e",
          "durationMs": 14200,
          "variants": { "hls": "https://.../master.m3u8", "thumb": "https://.../thumb.jpg" }
        },
        "engagement": { },
        "userInteraction": { }
      }
    ],
    "nextCursor": "MjAyNi0wOS0xN1QxMDoyMjozMTo0ZjJh...",
    "hasMore": true
  }
}
```

What the app must do

1. **Stop using** `clips.length < limit` **as the end-of-feed test.** Use `hasMore`.
2. **Honour** `visibility`. `INTERSTITIAL` clips must not autoplay behind no cover.
3. Expect **fewer clips than before** for some viewers. That is correct, not a regression.

3. Single clip

Endpoint: `GET` `/api/v1/e-social/clip/{mediaId}`

Change type: `SECURITY FIX` `RESPONSE` · **App action:** ☐ Optional

Access Level: ☐ Protected · **Authentication:** ☐ Authorization: Bearer <access token>

What changed

- **A clip the viewer may not see now returns** `404`, with the message `Clip not found`. It used to be returned in full to anyone holding the id. This covers followers-only and mentions-only posts, blocked authors, and clips dropped by a moderation label.
- Same two new fields as the feed: `visibility` and `media.durationMs`.

Error Response JSON Sample

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Clip not found",
  "action_time": "2026-09-18T10:30:45",
  "data": "Clip not found"
}
```

What the app should do

Treat `404` here as "this clip is gone or not for you" and return the user to the previous screen — it is no longer safe to assume a `404` means deleted.

4. A user's videos

Endpoint: ☐ GET `/api/v1/e-social/clip/user/{userId}/videos`

Change type: ☐ SECURITY FIX ☐ RESPONSE ☐ BEHAVIOUR · **App action:** ☐ Required

Access Level: ☐ Protected · **Authentication:** ☐ Authorization: Bearer <access token>

What changed

Exactly the same three things as the clip feed (entry 2), on a profile's video grid:

- private and blocked-author videos are filtered out; a viewer looking at someone else's profile now sees only what they are entitled to (the owner still sees their own);
- `visibility` and `media.durationMs` added;

- short pages with `hasMore: true` are possible, and `limit` is clamped to **1-20** (default 12).

What the app must do

Use `hasMore`, not the page length, to decide whether to load more in the profile grid.

5. Record a clip view

Endpoint: `POST` `/api/v1/e-social/clip/{mediaId}/view`

Change type: `BEHAVIOUR` · **App action:** `[]` Optional

Access Level: `[]` Protected · **Authentication:** `Authorization: Bearer <access token>`

Response: still `204 No Content`, still no body. The request contract has not changed.

What changed

- **A view is now counted once per viewer**, cheaply, in Redis. Repeat calls from the same viewer are accepted and ignored instead of inflating the count.
- **The `clip_views` table is no longer written.** Per-view rows are gone; the counter is what survives.
- **The request body is accepted and ignored.** Watch time does not belong here — it belongs in the interactions batch, where the recommendation model can read it (`05_INTERACTIONS_CLIENT_GUIDE.md`).

What the app should do

Keep calling this for the view counter, and send watch time as a `VIEW` event with `watchMs` and `mediaDurationMs` to `POST /api/v1/feed/interactions/batch`. Anything sent in this body is thrown away.

6. Posts for a hashtag

Endpoint: `GET` `/api/v1/e-social/hashtags/{hashtag}/posts`

Change type: `SECURITY FIX` `PARAMS` `BEHAVIOUR` `RESPONSE` · **App action:** `[]` None

Access Level: `[]` Public (a signed-in viewer sees more) · **Authentication:** optional bearer token

What changed

- a) It no longer leaks private posts.** The old version loaded every post id carrying the tag into memory and checked only "published and not deleted" — so followers-only and mentions-only posts reached anyone who opened the tag page. Visibility, mentions and blocks are now enforced in the database query itself.
- b) Paging is real.** The whole tag is no longer loaded into memory; the page and its total count are computed in the database.
- c) Ordering is by `published_at` descending** (it was `createdAt`). For a scheduled post these differ: the post now sorts by when it went live, which is what a reader expects.
- d) `page` and `size` are clamped** instead of being trusted.
- e) `topComments` is now populated** on each post, for the same reason as entry 1.
- f) Hashtags are matched in any script.** Tags are normalised the same way at write and read time, so non-Latin tags (Swahili, Arabic, CJK ...) now match instead of silently returning nothing.

Query Parameters

Parameter	Type	Required	Description	Validation	Default
<code>filter</code>	enum	No	<code>ALL</code> , <code>PRODUCTS</code> , <code>SHOPS</code> , <code>EVENTS</code>	unknown value → <code>400</code>	<code>ALL</code>
<code>page</code>	integer	No	1-based page number	Below 1 is treated as 1 (new)	1
<code>size</code>	integer	No	Posts per page	Clamped to 1-50 (new)	20

What the app should do

Nothing. Expect a signed-out viewer to see fewer posts than a signed-in one, and a blocked author's posts never to appear.

7. Trending hashtags

Endpoint: `GET` `/api/v1/e-social/hashtags/trending`

Change type: `BEHAVIOUR` `PARAMS` · **App action:** `[]` None

Access Level: ☐ Public · **Authentication:** none

Response shape: unchanged — a list of `{ hashtag, posts }`.

What changed

a) Trending is now hard to game. Tags are ranked by **how many different accounts** used them, then by how many posts. One account posting the same tag fifty times counts once, and `#sale #sale` inside one post counts once.

b) Private and draft posts never count. Only public, published, non-deleted posts inside the window — so a private post can no longer push its tag onto a public page.

c) Results are cached for 5 minutes per window. The page is public and read far more often than it changes.

Query Parameters

Parameter	Type	Required	Description	Validation	Default
<code>window</code>	string	No	<code>24h</code> , <code>7d</code> , <code>30d</code> . Anything else is treated as <code>24h</code>	—	<code>24h</code>
<code>limit</code>	integer	No	How many tags	Clamped to 1-50 (new)	15

8. Hashtag suggestions (type-ahead)

Endpoint: **GET** `/api/v1/e-social/hashtags/search`

Change type: **BEHAVIOUR** **PARAMS** · **App action:** ☐ None

Access Level: ☐ Public · **Authentication:** none

Response shape: unchanged — a list of `{ hashtag, posts }`.

What changed

- **Answered from a Redis suggestion index**, not from the database. The old query scanned and grouped the entire `post_hashtags` table **on every keystroke**.
- **Suggestions come from public posts only.** Tags that exist only on drafts or private posts are no longer suggested.

- `limit` is **clamped to 1-20** (default 5).
- An empty or blank `q` returns an empty list instead of an error.

Query Parameters

Parameter	Type	Required	Description	Validation	Default
<code>q</code>	string	Yes	What the user has typed so far	Blank → empty list	—
<code>limit</code>	integer	No	How many suggestions	Clamped to 1-20 (new)	5

Old endpoints being deleted

There is no migration and no grace period. This is a staging platform with no production users and no data worth keeping — the database will be dropped. So these paths are not being "migrated away from" over time; they are **deleted from the codebase in one commit**, as soon as the frontend confirms it has switched.

What that means for you:

- There is **no fallback**. Once the commit lands, these paths return `404`.
- There is **no dual-running period** to lean on, and no backend flag that will quietly keep them alive.
- Tell the backend team when your calls have moved. The deletion happens on that word.

The strings below are literal — copy them into your project search to find every call site.

Search for this	What it is	Replace with	Notes
<code>api/v1/e-social/feed</code>	Old home feed	<code>api/v1/feed/home</code>	Response shapes differ — see 09 A1
<code>api/v1/e-social/feed/stream</code>	Feed SSE stream	<i>nothing</i> — use <code>api/v1/events</code>	The chat/events stream now carries a <code>feed.new</code> hint; on receiving it, call <code>api/v1/feed/home/has-new</code>
<code>api/v1/e-social/feed/heartbeat</code>	SSE keep-alive	<i>nothing</i>	Dies with the stream above
<code>api/v1/e-social/clip/feed</code>	Clip feed	<code>api/v1/feed/reels</code>	Still amended in the meantime (entry 2), so it behaves correctly right up to the day it goes

Not going anywhere — keep calling these: `api/v1/e-social/clip/{mediaId}`, `api/v1/e-social/clip/{mediaId}/view`, `api/v1/e-social/clip/user/{userId}/videos`, and all three `api/v1/e-social/hashtags/...` endpoints.

A one-liner for a frontend repo:

```
grep -rn "e-social/feed|e-social/clip/feed" src/ --include=*.ts,tsx,js,jsx,dart,kt,swift
```

Every hit is a call site that must move. When that grep returns nothing, tell the backend team — that is the signal to delete.

What did *not* change

Worth stating plainly, so nobody goes looking:

- **No existing response field was renamed or removed**, anywhere.
- **PostResponse is unchanged**. Post, profile, search, bookmark and comment endpoints return exactly what they returned before. They are faster (a page of posts went from about 242 database queries to about 24) and they load the viewer's account once instead of twice, but the JSON is the same.
- **Engagement numbers** (likes, views, comments) keep the same field names. Some values were recounted and corrected as part of an earlier release.
- **No new notification** is produced by any endpoint in this document.
- Authentication, the response envelope and every error code behave as before.

Related documents

Doc	What it covers
09_NEW_ENDPOINTS.md	Every new endpoint and new controller
07_FEED_ENDPOINTS_GUIDE.md	The app team's walkthrough of the new feed screens
05_INTERACTIONS_CLIENT_GUIDE.md	What the app must report back, and exact action definitions
01_ARCHITECTURE.md	The design. §A.3 is the migration and the deletion list; §9.2 is the label rule