

Clips/Short Videos Mng API

Author: Josh S. Sakweli, Backend Lead Team

Last Updated: 2026-06-22

Version: v1.0

Base URL: `{base_url}/api/v1/e-social/clip`

Short Description: The Clips API powers the short-form vertical video surface on the platform. A clip is not a separate content type — it is a video media item attached to a post, where `shortClip = true`. This API provides the feed, individual clip access, view tracking, and user video profiles. Liking, commenting, and saving a clip are handled by the existing Posts interaction API using the clip's parent `postId`.

Why Clips Are Not Separate Posts:

“ **Clip** = a `post_media` row where `shortClip = true` and `mediaType = VIDEO`. **Post** = the parent container that owns the clip's caption, engagement counters (likes, comments, shares, saves), and attachments.

This design avoids duplicating the entire post infrastructure. A clip inherits everything social from its parent post. The only clip-specific data is the per-media `viewsCount` (each video tracks its own views independently) and the `clip_views` table used for view deduplication. A single post can contain multiple clips — both appear as separate items in the clip feed with their own view counts but share the same caption and like/comment counts.

Hints:

- The clip feed is paginated using an opaque cursor — pass `nextCursor` from each response back as `cursor` on the next request
- `POST /{mediaId}/view` is idempotent per user per clip — the first call counts the view, subsequent calls increment a personal `viewCount` only (no double-counting on the public counter)
- Liking a clip: `POST /api/v1/e-social/posts/{postId}/like` — use the `postId` from the clip response
- Commenting on a clip: `POST /api/v1/e-social/posts/{postId}/comments`
- Saving a clip: `POST /api/v1/e-social/posts/{postId}/bookmark`
- `GET /clip/user/{userId}/videos` returns ALL videos from a user (both long and short), not just clips — use this for the profile videos tab

- The `variants` map in the media object is fully dynamic — keys are whatever FileThunder generated for that file (see variants section below)
- View tracking requires authentication — anonymous views are silently ignored (no error returned)
- `currentUserIsPendingCollaborator: true` means the viewer has a pending collaboration invite on this clip — prompt them to accept or decline using the post collaboration endpoints with `postId`
- `collaborators[]` only contains accepted co-authors — pending or declined invites are never exposed to viewers

Standard Response Format

Success Response Structure

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Operation completed successfully",
  "action_time": "2026-06-22T10:30:45",
  "data": {}
}
```

Error Response Structure

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Clip not found",
  "action_time": "2026-06-22T10:30:45",
  "data": "Clip not found"
}
```

Standard Response Fields

Field	Type	Description
<code>success</code>	boolean	<code>true</code> for successful operations, <code>false</code> for errors

Field	Type	Description
<code>httpStatus</code>	string	HTTP status name (OK, NOT_FOUND, etc.)
<code>message</code>	string	Human-readable message describing the result
<code>action_time</code>	string	ISO 8601 timestamp of when the response was generated
<code>data</code>	object/string	Response payload on success, error details on failure

Standard Error Types

- `400 BAD_REQUEST`: Invalid cursor format or malformed request
- `401 UNAUTHORIZED`: Missing or invalid JWT token
- `404 NOT_FOUND`: Clip not found, deleted, or not published
- `500 INTERNAL_SERVER_ERROR`: Unexpected server error

Core Concepts

What Is a Clip?

A clip is a `post_media` record that satisfies all three conditions:

- `mediaType = VIDEO`
- `shortClip = true`
- `status = READY` (FileThunder has finished processing)

The parent post must also be `status = PUBLISHED` and `isDeleted = false`. Any clip whose parent post is deleted or unpublished is treated as non-existent — the API returns 404.

Clip Identity

Identifier	Used For
<code>id</code> (media ID)	Clip identity — share links, view tracking, deep links
<code>postId</code>	Social interactions — like, comment, save, share

The clip `id` is a UUID that uniquely identifies the video media item. The `postId` is used for all social actions since engagement (likes, comments, saves) belongs to the post, not to the individual video.

Cursor Pagination

The feed uses cursor-based pagination — not page numbers. The cursor is an opaque base64-encoded string that encodes the position of the last item returned. This guarantees stable pagination even when new clips are added between requests.

```
First call:  GET /clip/feed?limit=10
             → returns 10 clips + nextCursor

Second call: GET /clip/feed?cursor={nextCursor}&limit=10
             → returns the next 10 clips after the previous batch
```

When `hasMore: false`, the feed is exhausted. Pull-to-refresh starts a new feed from the top (omit the cursor).

View Deduplication

Views are counted once per user per clip. The dedup table (`clip_views`) enforces a unique constraint on `(mediaId, userId)`.

```
First view by user A on clip X:
  → new row in clip_views
  → post_media.views_count + 1
  → posts.views_count + 1

Second view by user A on clip X:
  → clip_views.view_count + 1 (personal counter only)
  → NO change to post_media.views_count or posts.views_count
```

Anonymous users (no JWT) are ignored — `POST /{mediaId}/view` returns `204` but records nothing.

Media Variants

The `variants` map is dynamic — keys are whatever FileThunder generated during video processing. For short clips the expected keys are:

Key	Type	Description
<code>master</code>	URL	HLS adaptive bitrate master playlist — use this for playback
<code>360p_playlist</code>	URL	HLS playlist for 360p stream
<code>720p_playlist</code>	URL	HLS playlist for 720p stream

Key	Type	Description
<code>1080p_playlist</code>	URL	HLS playlist for 1080p stream
<code>poster</code>	URL	Full-size thumbnail frame (.webp)
<code>thumb</code>	URL	Small thumbnail (.webp, 300px)
<code>og</code>	URL	OG/social share thumbnail (1200×630)
<code>blurhash</code>	string	Blurhash string — not a URL, used for blur placeholder
<code>lqip</code>	string	Base64 WebP data URI — inline low-quality placeholder
<code>dominant_color</code>	string	Hex color extracted from the thumbnail (e.g. <code>#2a2a2a</code>)
<code>preview_3s.mp4</code>	URL	6-second muted preview at 2× speed (360×640)
<code>720p_watermarked.mp4</code>	URL	Watermarked MP4 (short clips only)

Playback recommendation: use `master` for HLS adaptive streaming. Fall back to `720p_playlist` if the player does not support HLS. Use `lqip` as the placeholder while the video loads, then `poster` once the video is ready to play.

Endpoints

1. Get Clip Feed

Purpose: Returns a paginated feed of short clips, ordered by most recently published. This is the main scroll surface — each call returns the next batch of clips after the provided cursor.

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

Access Level: Protected (Requires valid JWT)

Authentication: Bearer Token

Request Headers:

Header	Type	Required	Description
<code>Authorization</code>	string	Yes	Bearer <jwt_token>

Query Parameters:

Parameter	Type	Required	Description	Validation	Default
cursor	string	No	Opaque pagination cursor from the previous response. Omit on the first call.	Valid base64 cursor string	null (first page)
limit	integer	No	Number of clips to return per page	Min: 1, Max: 20	10

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Clip feed fetched",
  "action_time": "2026-06-22T10:30:45",
  "data": {
    "clips": [
      {
        "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
        "postId": "1a2b3c4d-0000-0000-0000-000000000001",
        "author": {
          "id": "aaa-111-bbb-222-ccc",
          "username": "fitnessguru",
          "profilePictureUrl": "https://cdn.nexgate.com/users/aaa/profilepic/medium.webp",
          "profilePictureThumbnailUrl":
            "https://cdn.nexgate.com/users/aaa/profilepic/thumb.webp",
          "profilePictureBlurhash": "LK02?U%2Tw=w]~RBVZRi};RPxuwH",
          "verified": true,
          "following": false
        },
        "caption": "morning workout routine 🏋️‍♀️#fitness #gym",
        "media": {
          "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
          "duration": 42,
          "width": 1080,
          "height": 1920,
          "variants": {
            "master": "https://cdn.nexgate.com/posts/aaa/media/xyz/hls/master.m3u8",
```

```
    "360p_playlist": "https://cdn.nexgate.com/posts/aaa/media/xyz/hls/360p/360p.m3u8",
    "720p_playlist": "https://cdn.nexgate.com/posts/aaa/media/xyz/hls/720p/720p.m3u8",
    "1080p_playlist":
"https://cdn.nexgate.com/posts/aaa/media/xyz/hls/1080p/1080p.m3u8",
    "poster": "https://cdn.nexgate.com/posts/aaa/media/xyz/poster.webp",
    "thumb": "https://cdn.nexgate.com/posts/aaa/media/xyz/thumb.webp",
    "og": "https://cdn.nexgate.com/posts/aaa/media/xyz/og.webp",
    "blurhash": "LK02?U%2Tw=w]~RBVZRi};RPxuwH",
    "lqip": "data:image/webp;base64,UklGRlYAAABXRUJQ...",
    "dominant_color": "#1a1a2e",
    "preview_3s.mp4": "https://cdn.nexgate.com/posts/aaa/media/xyz/preview_3s.mp4",
    "720p_watermarked.mp4":
"https://cdn.nexgate.com/posts/aaa/media/xyz/720p_watermarked.mp4"
  }
},
"hashtags": ["fitness", "gym"],
"mentionedUsers": [
  {
    "id": "ddd-333-eee-444-fff",
    "username": "coachmark",
    "displayName": "Coach Mark",
    "profilePictureUrl": "https://cdn.nexgate.com/users/ddd/profilepic/thumb.webp"
  }
],
"mentionedShops": [
  {
    "id": "shop-bbb-222",
    "name": "GymGear Store",
    "shopSlug": "gymgear-store",
    "logoUrl": "https://cdn.nexgate.com/shops/gymgear/logo.webp"
  }
],
"collaborators": [
  {
    "id": "eee-555-fff-666-ggg",
    "username": "coachmark",
    "displayName": "Coach Mark",
    "profilePictureUrl": "https://cdn.nexgate.com/users/eee/profilepic/thumb.webp"
  }
],
```

```
"currentUserIsCollaborator": false,
"currentUserIsPendingCollaborator": true,
"attachments": {
  "products": [
    {
      "id": "prod-aaa-111",
      "name": "Nike Air Max 270",
      "price": 120.00,
      "discountPrice": 89.99,
      "imageUrl": "https://cdn.nexgate.com/products/nike/medium.webp",
      "shopName": "Nike Store",
      "inStock": true
    }
  ],
  "shop": {
    "id": "shop-aaa-111",
    "name": "Nike Store",
    "logoUrl": "https://cdn.nexgate.com/shops/nike/logo.webp"
  },
  "externalLink": null
},
"engagement": {
  "viewsCount": 50200,
  "likesCount": 1204,
  "commentsCount": 87,
  "sharesCount": 43,
  "savesCount": 210
},
"userInteraction": {
  "hasLiked": false,
  "hasSaved": false,
  "hasShared": false
},
"createdAt": "2026-06-20T08:30:00"
}
],
"nextCursor":
"MjAyNi0wNi0yMFQw0DozMDowMDozZmE4NWY2NC01NzE3LTQ1NjItYjNmYy0yYzk2M2Y2NmFmYTY=",
"hasMore": true
}
```

```
}
```

Success Response Fields:

Field	Description
<code>clips</code>	Array of clip objects for this page
<code>clips[].id</code>	Media ID — the unique identity of this clip. Use this for view tracking and share links
<code>clips[].postId</code>	Parent post ID — use this for like, comment, save, and share interactions
<code>clips[].author.id</code>	Author's account ID
<code>clips[].author.username</code>	Author's public username
<code>clips[].author.profilePictureUrl</code>	Medium-size profile picture URL
<code>clips[].author.profilePictureThumbnailUrl</code>	Thumbnail profile picture URL
<code>clips[].author.profilePictureBlurhash</code>	Blurhash string for the profile picture placeholder
<code>clips[].author.verified</code>	Whether the author has a verified badge
<code>clips[].author.following</code>	Whether the authenticated user follows this author
<code>clips[].caption</code>	Raw caption text of the parent post
<code>clips[].media.id</code>	Same as <code>clips[].id</code> — the media item ID
<code>clips[].media.duration</code>	Video duration in seconds
<code>clips[].media.width</code>	Video width in pixels
<code>clips[].media.height</code>	Video height in pixels
<code>clips[].media.variants</code>	Dynamic map of variant keys to fully-assembled URLs. See variants table above
<code>clips[].hashtags</code>	Array of hashtag strings extracted from the caption (without <code>#</code> prefix)
<code>clips[].mentionedUsers</code>	Array of users @mentioned inline in the caption text
<code>clips[].mentionedUsers[].id</code>	Mentioned user's account ID
<code>clips[].mentionedUsers[].username</code>	Mentioned user's public username
<code>clips[].mentionedUsers[].displayName</code>	Mentioned user's display name (may be <code>null</code> if not set)
<code>clips[].mentionedUsers[].profilePictureUrl</code>	Mentioned user's thumbnail profile picture URL
<code>clips[].mentionedShops</code>	Array of shops inline-mentioned in the caption via <code>@ShopName</code>
<code>clips[].mentionedShops[].id</code>	Shop ID
<code>clips[].mentionedShops[].name</code>	Shop display name
<code>clips[].mentionedShops[].shopSlug</code>	Shop URL slug

Field	Description
<code>clips[].mentionedShops[].logoUrl</code>	Shop logo URL
<code>clips[].collaborators</code>	Array of accepted co-authors on this post. Empty array if no collaborators
<code>clips[].collaborators[].id</code>	Collaborator's account ID
<code>clips[].collaborators[].username</code>	Collaborator's public username
<code>clips[].collaborators[].displayName</code>	Collaborator's display name (may be <code>null</code> if not set)
<code>clips[].collaborators[].profilePictureUrl</code>	Collaborator's thumbnail profile picture URL
<code>clips[].currentUserIsCollaborator</code>	<code>true</code> if the authenticated user is an accepted collaborator on this post
<code>clips[].currentUserIsPendingCollaborator</code>	<code>true</code> if the authenticated user has a pending collaboration invite on this post — show an accept/decline prompt
<code>clips[].attachments.products</code>	Lightweight product cards attached to the post (max display: all attached products)
<code>clips[].attachments.shop</code>	The first shop attached to the post, or <code>null</code>
<code>clips[].attachments.externalLink</code>	External link with short URL and preview, or <code>null</code>
<code>clips[].engagement.viewsCount</code>	Total unique views for this specific video (per media item, not per post)
<code>clips[].engagement.likesCount</code>	Total likes on the parent post
<code>clips[].engagement.commentsCount</code>	Total comments on the parent post
<code>clips[].engagement.sharesCount</code>	Total shares on the parent post
<code>clips[].engagement.savesCount</code>	Total saves/bookmarks on the parent post
<code>clips[].userInteraction.hasLiked</code>	Whether the authenticated user has liked the parent post
<code>clips[].userInteraction.hasSaved</code>	Whether the authenticated user has saved/bookmarked the parent post
<code>clips[].userInteraction.hasShared</code>	Whether the authenticated user has shared the parent post
<code>clips[].createdAt</code>	ISO 8601 timestamp of when the parent post was published
<code>nextCursor</code>	Opaque cursor string to pass on the next request. <code>null</code> when <code>hasMore</code> is false
<code>hasMore</code>	<code>true</code> if more clips exist beyond this page, <code>false</code> if the feed is exhausted

Error Response Examples:

Bad cursor (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Invalid cursor",
  "action_time": "2026-06-22T10:30:45",
  "data": "Invalid cursor"
}
```

Unauthorized (401):

```
{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Token has expired",
  "action_time": "2026-06-22T10:30:45",
  "data": "Token has expired"
}
```

2. Get Single Clip

Purpose: Fetches a single clip by its media ID. Used for deep links, share links, and reopening a clip from a notification or profile tap.

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

Access Level: Protected (Requires valid JWT)

Authentication: Bearer Token

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer <jwt_token>

Path Parameters:

Parameter	Type	Required	Description	Validation
mediaId	UUID	Yes	The media ID of the clip	Valid UUID format

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Clip fetched",
  "action_time": "2026-06-22T10:30:45",
  "data": {
    "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "postId": "1a2b3c4d-0000-0000-0000-000000000001",
    "author": {
      "id": "aaa-111-bbb-222-ccc",
      "username": "fitnessguru",
      "profilePictureUrl": "https://cdn.nexgate.com/users/aaa/profilepic/medium.webp",
      "profilePictureThumbnailUrl": "https://cdn.nexgate.com/users/aaa/profilepic/thumb.webp",
      "profilePictureBlurhash": "LK02?U%2Tw=w]~RBVZri};RPxuwH",
      "verified": true,
      "following": true
    },
    "caption": "morning workout routine ☑️#fitness #gym",
    "media": {
      "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
      "duration": 42,
      "width": 1080,
      "height": 1920,
      "variants": {
        "master": "https://cdn.nexgate.com/posts/aaa/media/xyz/hls/master.m3u8",
        "360p_playlist": "https://cdn.nexgate.com/posts/aaa/media/xyz/hls/360p/360p.m3u8",
        "720p_playlist": "https://cdn.nexgate.com/posts/aaa/media/xyz/hls/720p/720p.m3u8",
        "1080p_playlist": "https://cdn.nexgate.com/posts/aaa/media/xyz/hls/1080p/1080p.m3u8",
        "poster": "https://cdn.nexgate.com/posts/aaa/media/xyz/poster.webp",
        "thumb": "https://cdn.nexgate.com/posts/aaa/media/xyz/thumb.webp",
        "og": "https://cdn.nexgate.com/posts/aaa/media/xyz/og.webp",
        "blurhash": "LK02?U%2Tw=w]~RBVZri};RPxuwH",
        "lqip": "data:image/webp;base64,UklGRlYAAABXRUJQ...",
        "dominant_color": "#1a1a2e",
        "preview_3s.mp4": "https://cdn.nexgate.com/posts/aaa/media/xyz/preview_3s.mp4",
        "720p_watermarked.mp4":
      }
    }
  }
}
```

```
},
"hashtags": ["fitness", "gym"],
"mentionedUsers": [],
"mentionedShops": [],
"collaborators": [
  {
    "id": "eee-555-fff-666-ggg",
    "username": "coachmark",
    "displayName": "Coach Mark",
    "profilePictureUrl": "https://cdn.nexgate.com/users/eee/profilepic/thumb.webp"
  }
],
"currentUserIsCollaborator": true,
"currentUserIsPendingCollaborator": false,
"attachments": {
  "products": [],
  "shop": null,
  "externalLink": {
    "shortUrl": "https://nexgate.link/abc123",
    "preview": {
      "title": "Best Workout Plans 2026",
      "imageUrl": "https://external-domain.com/og-image.jpg",
      "domain": "workoutplans.com"
    }
  }
}
},
"engagement": {
  "viewsCount": 50200,
  "likesCount": 1204,
  "commentsCount": 87,
  "sharesCount": 43,
  "savesCount": 210
},
"userInteraction": {
  "hasLiked": true,
  "hasSaved": false,
  "hasShared": false
},
"createdAt": "2026-06-20T08:30:00"
}
```

```
}
```

Success Response Fields: Same field definitions as the feed endpoint above. This returns a single clip object (not wrapped in a feed).

Error Response Examples:

Clip not found (404):

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Clip not found",
  "action_time": "2026-06-22T10:30:45",
  "data": "Clip not found"
}
```

Media exists but is not a clip (404):

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Media is not a clip",
  "action_time": "2026-06-22T10:30:45",
  "data": "Media is not a clip"
}
```

Clip exists but is still processing (404):

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Clip is not available",
  "action_time": "2026-06-22T10:30:45",
  "data": "Clip is not available"
}
```

“ **Note:** A deleted or unpublished parent post also returns `404 Clip not found` — the API does not distinguish between "media doesn't exist" and "media exists but is hidden" to prevent enumeration.

3. Record a View

Purpose: Records that the authenticated user has watched a clip. The first call increments the public `viewsCount`. Subsequent calls by the same user only increment a personal view counter — the public count is never double-counted. Anonymous users are silently ignored.

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

Access Level: `⏏` Protected (Requires valid JWT — anonymous calls return `204` but are not recorded)

Authentication: Bearer Token

Request Headers:

Header	Type	Required	Description
<code>Authorization</code>	string	Yes	Bearer <code><jwt_token></code>

Path Parameters:

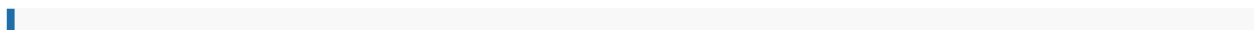
Parameter	Type	Required	Description	Validation
<code>mediaId</code>	UUID	Yes	The media ID of the clip being viewed	Valid UUID format

Request JSON Sample:

```
{
  "watchDurationMs": 38000,
  "source": "FEED"
}
```

Request Body Parameters:

Parameter	Type	Required	Description	Validation
<code>watchDurationMs</code>	long	No	How long the user watched in milliseconds	Any positive long value
<code>source</code>	string	No	Where the user encountered the clip	Recommended: <code>FEED</code> , <code>PROFILE</code> , <code>SHARE_LINK</code>



Note: The request body is optional. Sending `POST /{mediaId}/view` with no body is valid and records the view. `watchDurationMs` and `source` are accepted and stored for future analytics — they do not affect view counting logic today.

Success Response:

`204 No Content` — no response body.

View Counting Logic:

User A watches clip X for the first time:

- New row written to `clip_views` (`mediaId=X`, `userId=A`, `viewCount=1`)
- `post_media.views_count` incremented by 1
- `posts.views_count` incremented by 1

User A watches clip X again:

- `clip_views.view_count` incremented (now 2) — personal history only
- `post_media.views_count` NOT changed — no inflation
- `posts.views_count` NOT changed

Anonymous user (no JWT) watches clip X:

- Returns 204 silently
- Nothing recorded

Error Response Examples:

Clip not found or deleted (404):

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Clip not found",
  "action_time": "2026-06-22T10:30:45",
  "data": "Clip not found"
}
```

4. Get User Videos

Purpose: Returns all videos (both long-form and short clips) published by a specific user, ordered by most recently published. Used for the videos tab on a user's profile page.

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

Access Level: `Protected` (Requires valid JWT)

Authentication: Bearer Token

Request Headers:

Header	Type	Required	Description
<code>Authorization</code>	string	Yes	Bearer <jwt_token>

Path Parameters:

Parameter	Type	Required	Description	Validation
<code>userId</code>	UUID	Yes	The account ID of the user whose videos to fetch	Valid UUID format

Query Parameters:

Parameter	Type	Required	Description	Validation	Default
<code>cursor</code>	string	No	Opaque pagination cursor from the previous response	Valid base64 cursor string	<code>null</code> (first page)
<code>limit</code>	integer	No	Number of videos to return per page	Min: 1, Max: 20	<code>12</code>

“ **Difference from** `/clip/feed`: This endpoint does NOT filter by `shortClip = true`. It returns all processed videos from the user regardless of length, making it suitable for the profile videos grid.

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "User videos fetched",
  "action_time": "2026-06-22T10:30:45",
```

```
"data": {
  "clips": [
    {
      "id": "7cb91a00-0000-0000-0000-000000000001",
      "postId": "8dc02b11-0000-0000-0000-000000000002",
      "author": {
        "id": "aaa-111-bbb-222-ccc",
        "username": "fitnessguru",
        "profilePictureUrl": "https://cdn.nexgate.com/users/aaa/profilepic/medium.webp",
        "profilePictureThumbnailUrl":
"https://cdn.nexgate.com/users/aaa/profilepic/thumb.webp",
        "profilePictureBlurhash": "LK02?U%2Tw=w]~RBVZRi};RPxuwH",
        "verified": true,
        "following": true
      },
      "caption": "Full 30-minute chest day ☑☑#gym #workout",
      "media": {
        "id": "7cb91a00-0000-0000-0000-000000000001",
        "duration": 1842,
        "width": 1920,
        "height": 1080,
        "variants": {
          "master": "https://cdn.nexgate.com/posts/aaa/media/abc/hls/master.m3u8",
          "360p_playlist": "https://cdn.nexgate.com/posts/aaa/media/abc/hls/360p/360p.m3u8",
          "720p_playlist": "https://cdn.nexgate.com/posts/aaa/media/abc/hls/720p/720p.m3u8",
          "1080p_playlist":
"https://cdn.nexgate.com/posts/aaa/media/abc/hls/1080p/1080p.m3u8",
          "poster": "https://cdn.nexgate.com/posts/aaa/media/abc/poster.webp",
          "thumb": "https://cdn.nexgate.com/posts/aaa/media/abc/thumb.webp",
          "blurhash": "LK02?U%2Tw=w]~RBVZRi};RPxuwH",
          "lqip": "data:image/webp;base64,UklGRlYAAABXRUJQ...",
          "dominant_color": "#0d0d1a"
        }
      },
      "hashtags": ["gym", "workout"],
      "mentionedUsers": [],
      "mentionedShops": [],
      "collaborators": [],
      "currentUserIsCollaborator": false,
      "currentUserIsPendingCollaborator": false,
    }
  ]
}
```

```

    "attachments": {
      "products": [],
      "shop": null,
      "externalLink": null
    },
    "engagement": {
      "viewsCount": 8900,
      "likesCount": 320,
      "commentsCount": 14,
      "sharesCount": 8,
      "savesCount": 45
    },
    "userInteraction": {
      "hasLiked": false,
      "hasSaved": false,
      "hasShared": false
    },
    "createdAt": "2026-06-18T14:00:00"
  }
],
"nextCursor":
"MjAyNi0wNi0xOFQxNDowMDowMD03Y2I5MWEwMC0wMDAwLTAwMDAtMDAwMC0wMDAwMDAwMDAwMDE=",
"hasMore": false
}
}

```

Success Response Fields: Same as the feed response. Note that `shortClip` is not part of the response — long videos and short clips look identical in this response. The frontend can distinguish them by `media.duration` if needed (short clips are under 180 seconds).

Error Response Examples:

Unauthorized (401):

```

{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Token has expired",
  "action_time": "2026-06-22T10:30:45",
  "data": "Token has expired"
}

```

How Clip Interactions Work

Clips share the social layer of their parent post. The clip response always includes `postId` so the frontend knows which ID to use for each action.

Action	Endpoint	ID to use
Like a clip	POST /api/v1/e-social/posts/{postId}/like	<code>postId</code> from clip response
Unlike a clip	DELETE /api/v1/e-social/posts/{postId}/like	<code>postId</code> from clip response
Comment on a clip	POST /api/v1/e-social/posts/{postId}/comments	<code>postId</code> from clip response
Save a clip	POST /api/v1/e-social/posts/{postId}/bookmark	<code>postId</code> from clip response
Share a clip	POST /api/v1/e-social/posts/{postId}/share	<code>postId</code> from clip response
Record a view	POST /api/v1/e-social/clip/{mediaId}/view	<code>id</code> (media ID) from clip response
Deep link to clip	GET /api/v1/e-social/clip/{mediaId}	<code>id</code> (media ID) from clip response
Accept collaboration invite	POST /api/v1/e-social/posts/{postId}/collaboration/accept	<code>postId</code> from clip response
Decline collaboration invite	POST /api/v1/e-social/posts/{postId}/collaboration/decline	<code>postId</code> from clip response

“ **Collaboration flow:** when `currentUserIsPendingCollaborator` is `true`, show the user an accept/decline prompt. Use the `postId` from the clip response with the existing post collaboration endpoints above. Once accepted, `currentUserIsCollaborator` becomes `true` and the user appears in the `collaborators` array.

Quick Reference

Endpoint Summary

Method	Path	Auth	Purpose
--------	------	------	---------

GET	/api/v1/e-social/clip/feed	JWT	Scroll feed — paginated clips
GET	/api/v1/e-social/clip/{mediaId}	JWT	Single clip by media ID
POST	/api/v1/e-social/clip/{mediaId}/view	JWT	Record a view
GET	/api/v1/e-social/clip/user/{userId}/videos	JWT	All videos from a user

Key ID Reference

Use Case	Which ID	Field Name
Share link, deep link, view tracking	Media ID	clip.id
Like, comment, save, share	Post ID	clip.postId

Recommended Playback Priority

1. variants["master"] → HLS adaptive bitrate (best quality, adapts to connection)
2. variants["720p_playlist"] → HLS fixed 720p fallback
3. variants["360p_playlist"] → HLS fixed 360p low bandwidth fallback

Recommended Loading Placeholder Priority

1. variants["lqip"] → Show immediately (inline base64, no network request)
2. variants["blurhash"] → Render as CSS blur using a blurhash library
3. variants["dominant_color"] → Solid color fallback

Revision #4

Created 22 June 2026 09:21:29 by Admin Qbit

Updated 22 June 2026 09:54:21 by Admin Qbit