Sending interactions — guide for the app team

Audience: Android, iOS and web developers. **Status:** built on the backend (branch `fet/feed_baking`, 2026-09-17).

The recommendation model learns from what people do. It can only learn from what the app sends, and **anything not sent today is lost for good**: a watch time nobody recorded cannot be recovered later. Please send everything below from the first release that has the endpoint.

The same action must mean the same thing on every platform. A `VIEW` on Android and a `VIEW` on iOS are counted together; if their definitions differ, the model learns the difference between the apps instead of what people like.

---

1. The endpoint

```
POST /api/v1/feed/interactions/batch
Authorization: Bearer <access token>
Content-Type: application/json
```

```
{
  "sessionId": "s-8812e0c4",
  "client": { "platform": "ANDROID", "appVersion": "2.4.1", "networkType": "CELLULAR" },
  "events": [
    { "clientEventId": "c-000184", "action": "IMPRESSION", "targetType": "POST", "targetId":
"4f2a77e1-0c8b-4d2f-9a11-7b6c3e5d8f90",
      "occurredAt": "2026-09-17T10:22:31.900Z",
      "context": { "surface": "REELS", "position": 3, "feedSessionId": "fs-77aa", "source":
"RECO" } },
    { "clientEventId": "c-000185", "action": "VIEW", "targetType": "POST", "targetId":
"4f2a77e1-0c8b-4d2f-9a11-7b6c3e5d8f90",
      "occurredAt": "2026-09-17T10:22:41.100Z",
      "watchMs": 9000, "mediaDurationMs": 6000, "loopCount": 1, "soundOn": true, "mediaIndex":
0,
      "context": { "surface": "REELS", "position": 3, "feedSessionId": "fs-77aa", "source":
```

```
"RECO" } }
  ]
}
```

Answer: `202`

```
{ "data": { "accepted": 2, "dropped": 0, "rejected": [] } }
```

- `rejected` lists events that were invalid, each with a `reason`. Fix the app; **do not resend them**.
- `dropped` counts valid events the server did not keep (rate limit or a temporary broker problem). **Do not resend them** either.
- Only resend a batch when the request itself failed (no connection, timeout, `5xx`). Resending is safe: the server recognises repeats by `clientEventId`.
- Never put an account id in the body. The account always comes from the token; any id in the body is ignored.
- Unknown fields are ignored, so an older server never rejects a newer app.

---

2. When to send

Rule	Value
Send every	10 seconds
...or as soon as the buffer holds	50 events
...and always when	the app goes to the background
Maximum per request	100 events (extra events are rejected with `BATCH_LIMIT_100`)
Offline	keep up to 500 unsent events on disk; beyond that drop the **oldest**
Rate limit	600 events per minute per account; normal use never reaches it

Never send one request per event.

---

3. Fields

Batch

Field	Required	Notes
`sessionId`	recommended	A new id each time the app comes to the foreground after ≥ 30 min away. ≤ 64 characters
`client.platform`	recommended	`ANDROID`, `IOS`, `WEB`
`client.appVersion`	recommended	e.g. `2.4.1`
`client.networkType`	recommended	`WIFI`, `CELLULAR`, `OFFLINE`, `UNKNOWN`. A clip abandoned on a slow connection means something different from one skipped on wifi

Event

Field	Required	Notes
`clientEventId`	**yes**	Unique per event on this device, ≤ 64 characters. A counter or a UUID. Reuse it when resending the same event
`action`	**yes**	See §4
`targetType`	yes, except `SEARCH`	`POST`, `PRODUCT`, `SHOP`, `EVENT`, `PROFILE`. A reel is a `POST`
`targetId`	yes, except `SEARCH`	The item's UUID
`occurredAt`	recommended	When it happened on the device, ISO-8601 UTC (`2026-09-17T10:22:31.900Z`). Omitted → the time the server received it. More than 24 h old or more than 5 min ahead is pulled into that window
`dwellMs`	for `VIEW` of images, products, events, shops, profiles	Time the item was on screen (≥ 50 % visible)
`watchMs`	for `VIEW` / `SKIP` of video	Total time actually played, **including repeats**. 9 s on a 6 s clip that looped once is `9000`
`mediaDurationMs`	for video	The clip's length
`loopCount`	for video	How many times it restarted from the beginning
`soundOn`	for video	Whether sound was on while playing
`mediaIndex`	for carousels	Which item (0-based) of a post with several images or videos
`context.surface`	**recommended**	`FEED`, `REELS`, `SEARCH`, `PRODUCT_PAGE`, `SHOP_PAGE`, `EVENT_PAGE`, `PROFILE_PAGE`, `RECOMMENDATION`, `NOTIFICATION`, `CHAT`

Field	Required	Notes
`context.position`	**recommended in lists**	0-based rank on screen. Without it the model cannot tell a tap earned by relevance from a tap earned by being first
`context.feedSessionId`	when the item came from the feed	The feed page's session id (the feed API will return it)
`context.source`	when the item came from the feed	`FOLLOWING`, `CELEBRITY`, `RECO`, `INTEREST`, `FALLBACK`, `SPONSORED`, `TRENDING`, `SEARCH`, as returned by the feed API
`context.viaPostId`	when acting on something attached to a post	The post's id, e.g. tapping a product tagged in a post
`context.query`	for `SEARCH`	What was searched; trimmed, cut at 200 characters

Send numbers as whole milliseconds. Leave a field out rather than sending `0` when it was not measured.

---

4. Actions — exact definitions

Action	Send when	Carries
`IMPRESSION`	The item is **≥ 50 % visible for ≥ 300 ms**. **Once per item per feed session**, however often it scrolls back into view	`context`
`VIEW`	Image, product, event, shop, profile: **≥ 50 % visible for ≥ 1 s**. Video: **played ≥ 2 s**. Send once, when the item leaves the screen, with the total time	`dwellMs` or `watchMs`/`mediaDurationMs`/ `loopCount`/`soundOn`, `mediaIndex`
`SKIP`	A video was scrolled away **less than 1 s after it started**	`watchMs`, `mediaDurationMs`
`CLICK`	The detail page, shop or profile was opened from the item	`context`
`SEARCH`	A query was submitted	`context.query`, no target
`NOT_INTERESTED`	"Show fewer like this"	
`HIDE`	"Hide this"	

Do not send `LIKE`, `UNLIKE`, `COMMENT`, `BOOKMARK`, `SHARE`, `ADD_TO_CART`, `REMOVE_FROM_CART`, `PURCHASE`, `BLOCK`, `REPORT` or `MUTE_AUTHOR`. Just call the feature's own API (like a post, add to cart, pay). The backend sends the interaction itself once the change is saved, so it is counted exactly once, even if

the app crashes before its next batch, and never for a like that failed. The batch API rejects these actions with `SENT_BY_SERVER`.

What the backend cannot know is where the tap happened. Keep sending `CLICK` with `context` (surface, position, source, `viaPostId`) when a product, event or profile is opened; that is how the model learns a post led to a purchase.

Clips (reels)

- A clip is a `POST`. Use the clip's `postId` as `targetId`, not its `id` (that is the media id); `mediaIndex` says which video of the post.
- `mediaDurationMs` = `media.durationMs` from the clip response.
- `POST /api/v1/e-social/clip/{id}/view` only counts the view (once per person). It ignores `watchDurationMs` and `source`: watch time reaches the model only through this batch.
- A clip with `visibility: INTERSTITIAL` gets a "sensitive content" cover; play only after the viewer taps through.

Worked examples

A reel watched to the end and replayed partly, sound on, on cellular `VIEW`, `watchMs: 9000`, `mediaDurationMs: 6000`, `loopCount: 1`, `soundOn: true`, surface `REELS`.

A reel flicked past `IMPRESSION` (if it met 50 % / 300 ms), then `SKIP` with `watchMs: 400`, `mediaDurationMs: 15000`.

Third image of a carousel post looked at for 2.5 s `VIEW`, `targetType: POST`, `mediaIndex: 2`, `dwellMs: 2500`.

A product tagged in a post, opened then added to cart `CLICK`, `targetType: PRODUCT`, `context.viaPostId` = the post. The add to cart itself is sent by the backend.

Search `SEARCH`, `context: { surface: "SEARCH", query: "running shoes" }`; then an `IMPRESSION` per result with `surface: SEARCH` and its `position`.

---

5. Rejection reasons

Reason	Meaning
`CLIENT_EVENT_ID_REQUIRED_MAX_64`	Missing or too long
`UNKNOWN_ACTION`	Not in §4
`SENT_BY_SERVER`	An action the backend already sends (like, comment, cart, purchase, …); call the feature API only

Reason	Meaning
`UNKNOWN_TARGET_TYPE` / `TARGET_ID_NOT_UUID`	Target missing or malformed
`DUPLICATE_CLIENT_EVENT_ID`	Same `clientEventId` twice in one batch
`OCCURRED_AT_NOT_ISO_8601`	Unparseable time
`DURATION_OUT_OF_RANGE`	A duration below 0 or above 6 h (12 h for `mediaDurationMs`)
`LOOP_COUNT_OUT_OF_RANGE` / `MEDIA_INDEX_OUT_OF_RANGE`	Out of 0–1000 / 0–100
`UNKNOWN_SURFACE` / `UNKNOWN_SOURCE` / `POSITION_OUT_OF_RANGE` / `VIA_POST_ID_NOT_UUID`	Context fields malformed
`BATCH_LIMIT_100`	More than 100 events in one request
`EMPTY_EVENT`	A `null` in the list

A rejection is a bug in the app. Log it in development builds.

Revision #1
Created 18 September 2026 09:33:00 by Admin Qbit
Updated 18 September 2026 09:33:14 by Admin Qbit