

Event Group Tickets ("Buy Together") API

Author: Josh S. Sakweli, Backend Lead Team **Last Updated:** 2026-10-07 **Version:** v1.0

Base URL: `http://localhost:8765/api/v1` (local) — `https://dev.api.nexgate.co/api/v1` (staging)

“ **v1.0** — tested end to end locally on 2026-10-07: starting and filling a group, two buyers racing for the last ticket, moving tickets between groups, a group expiring with refunds, cancelling a group booking, an organizer payout and an event cancel. The JSON samples below are real local responses.

Short Description: An organizer can let people buy a paid ticket type together at a cheaper **group price**. One buyer starts a group (for example "3 tickets at 15,000 instead of 20,000"), friends join with a code, and when the group is full everyone gets their tickets. If the group does not fill in time, everyone is refunded automatically.

Hints:

- Every endpoint needs `Authorization: Bearer <accessToken>`.
 - **Group size counts tickets, not people.** A group of 3 can be one person with 2 tickets plus one person with 1.
 - **No ticket, no QR code until the group is full.** While a group is open the buyer has paid but has no booking. Show "waiting for group: 2 of 3", never a ticket.
 - **Paid tickets only.** Free tickets, donation tickets and door sales cannot be bought as a group.
 - **A group closes at the earliest of:** its time limit, the end of ticket sales, or **2 hours before the event starts**. Show `expiresAt` as the deadline.
 - **Each friend joins with their own account.** Group checkouts are for the buyer only; `otherAttendees` is refused.
 - **Money is held per ticket.** If a group fails, each ticket still in it is refunded on its own, so moving some tickets to another group never leaves money stuck.
 - Times (`expiresAt`, `createdAt`, ...) carry no timezone and are server local time (EAT).
-

What is new — all endpoints at a glance

Group tickets are new; nothing existing was reshaped. Existing endpoints only gained fields.

Endpoint	New / changed	What for
<code>PATCH /e-events/tickets/{ticketId}/group-buying</code>	New	Organizer turns group buying on/off for a ticket type
<code>GET /e-events/tickets/{eventId}/{ticketId}</code> and the ticket list	Changed — new <code>groupBuying</code> object	Organizer sees available / held in groups / paid / open groups / sold
<code>POST /e-events/checkout</code>	Changed — new optional <code>group</code> object	Start a group (<code>groupName</code>) or join one (<code>groupId</code>)
<code>POST /e-events/checkout/{sessionId}/payment</code>	Same	Pays; fills the group if it was the last place
<code>GET /e-events/ticket-groups/ticket/{ticketTypeId}/open</code>	New	Open groups for a ticket
<code>GET /e-events/ticket-groups/code/{groupCode}</code>	New	Look up a shared code
<code>GET /e-events/ticket-groups/{groupId}</code>	New	One group
<code>GET /e-events/ticket-groups/my</code>	New	My groups
All four group reads above	Changed — new <code>participants</code> list	See who is in the group and how many tickets each holds
<code>POST /e-events/ticket-groups/{groupId}/transfer</code>	New	Move any number of my tickets to another group of the same ticket type
<code>POST /e-events/ticket-groups/{groupId}/withdraw</code>	New	Leave with any number of my tickets, full refund
<code>PATCH /e-events/booking-orders/{bookingId}/cancel</code>	Same	Also works for bookings that came from a group

When does a group end? A timer for its exact deadline, plus a check every 10 minutes as a safety net. Tickets are issued in the same moment as the payment that fills the group — there is no gap where a full group has no tickets.

Part 0 — Which part do I need?

I am building...	Read
------------------	------

The organizer's ticket settings screen	Part 1
The ticket page: "Buy together", open groups, join by code	Part 2 and Part 3
"My groups", moving tickets, leaving a group	Part 4
Notifications and where a tap goes	Part 5

Part 1 — Organizer: turn group buying on

PATCH /e-events/tickets/{ticketId}/group-buying

Only the event organizer can call it.

```
{
  "enabled": true,
  "groupPrice": 15000,
  "groupSize": 3,
  "timeLimitHours": 24
}
```

Field	Rule
enabled	Required. false turns it off; groups already running carry on and finish normally
groupPrice	Required when turning on. Above 0 and below the ticket price
groupSize	Required when turning on. At least 2, at most the ticket's total quantity
timeLimitHours	Required when turning on. 1 to 720

Changing price or size later does not affect groups already running — each group keeps the price and size it started with.

Errors (all 400, message shown as-is):

- Group buying is only for paid tickets

- Group price must be lower than the ticket price (20000.00)
- Group size cannot be larger than the ticket's total quantity
- groupPrice, groupSize and timeLimitHours are required to turn group buying on

The groupBuying object on ticket responses

Every ticket response (GET /e-events/tickets/{eventId}/{ticketId}, the event's ticket list, and the PATCH above) now has a groupBuying object. It is **missing** when group buying is off and no group is running.

```
"groupBuying": {
  "enabled": true,
  "groupPrice": 15000.0,
  "groupSize": 3,
  "timeLimitHours": 24,
  "availableToBuy": 14,
  "heldInOpenGroups": 3,
  "paidInOpenGroups": 2,
  "openGroups": 1,
  "sold": 0
}
```

Field	Meaning
availableToBuy	Tickets anyone can still buy right now
heldInOpenGroups	Tickets kept back for groups that are still filling (a whole group's worth is kept from the moment it starts)
paidInOpenGroups	How many of those have actually been paid for
openGroups	Groups running now
sold	Tickets that became real bookings, including from filled groups

Part 2 — Browse groups

GET /e-events/ticket-
groups/ticket/{ticketTypeId}/open

Open groups for a ticket type, soonest deadline first. Use it for the list on the ticket page.

GET /e-events/ticket-groups/code/{groupCode}

Look a group up by the code a friend shared (TG- + 6 characters, case does not matter).

GET /e-events/ticket-groups/{groupId}

One group.

A group looks like this (same shape everywhere):

```
{
  "groupId": "3a6d...",
  "groupCode": "TG-U9LDUE",
  "groupName": "Yoyo Crew",
  "eventId": "1635...",
  "eventTitle": "Group Ticket Test Night",
  "ticketTypeId": "8a31...",
  "ticketTypeName": "VIP Pass",
  "regularPrice": 20000.00,
  "groupPrice": 15000.00,
  "totalSeats": 3,
  "seatsOccupied": 2,
  "seatsRemaining": 1,
  "members": 1,
  "status": "OPEN",
  "initiatorUserName": "mamayoyo",
  "createdAt": "2026-10-07T11:34:55",
  "expiresAt": "2026-10-08T11:34:55",
  "myTickets": 2,
  "participants": [
    {
      "userId": "7cle...",
      "userName": "mamayoyo",
      "userProfileMedia": { "fileId": "...", "status": "READY", "thumbUrl": "https://...",
      "mimeType": "image/jpeg" },
      "tickets": 2,
```

```

    "joinedAt": "2026-10-07T11:34:55",
    "isMe": true
  }
]
}

```

Field	Notes
<code>status</code>	<code>OPEN</code> , <code>COMPLETED</code> (full, tickets issued) or <code>FAILED</code> (ended, refunded)
<code>endReason</code>	Only when ended: <code>FILLED</code> , <code>EXPIRED</code> , <code>EMPTIED</code> (everyone moved out), <code>EVENT_CANCELLED</code> , <code>TICKET_DELETED</code>
<code>endedAt</code>	Only when ended
<code>myTickets</code>	How many of the viewer's tickets are in this group. Missing when none
<code>participants</code>	Who is in the group, first to join first. One entry per person, with how many <code>tickets</code> they hold. No money is shown. Empty once a group has <code>FAILED</code> (everyone was refunded)
<code>participants[].isMe</code>	<code>true</code> on the viewer's own entry, missing on everyone else's — use it to highlight "you"

Part 3 — Start or join a group at checkout

The normal event checkout, with one extra object, `group`.

POST /e-events/checkout

Start a group:

```

{
  "eventId": "1635...",
  "ticketTypeId": "8a31...",
  "ticketsForMe": 2,
  "group": { "groupName": "Yoyo Crew" }
}

```

Join a group:

```
{
  "eventId": "1635...",
  "ticketTypeId": "8a31...",
  "ticketsForMe": 1,
  "group": { "groupId": "3a6d..." }
}
```

The price in the response is already the group price. The response has a new `group` object:

```
"group": { "groupName": "Yoyo Crew", "startsNewGroup": true }
```

(when joining: `{ "groupId": "3a6d...", "startsNewGroup": false }`)

Then pay exactly as today: `POST /e-events/checkout/{sessionId}/payment`.

What happens on payment:

- **Started a group** → the group exists, `status: OPEN`. No booking yet.
- **Joined and the group is not full yet** → you are in, no booking yet.
- **Your payment filled the group** → every member gets their booking and tickets straight away, with QR codes, in **My bookings**.

Checkout errors (`400`):

- Group name is required when starting a group
- An open group named 'yoyo crew' already exists for this ticket (names are unique per ticket, ignoring case)
- Only 1 tickets left in this group
- This group is no longer open
- This group is for a different ticket
- Group buying is not available for this ticket
- Group tickets are bought for yourself; each friend joins the group with their own account
- It is too close to the event to start a group
- You can hold at most 6 of this ticket; 2 are already waiting in groups (the ticket's per-person limit counts tickets waiting in groups too)

Payment error you must handle: This group just filled up or closed. You have not been charged. — two people paid for the last place at the same moment and the other one got it. Nothing was taken from the wallet; send the user back to the group list.

Part 4 — My groups and moving tickets

GET /e-events/ticket-groups/my

Every group the user has (or had) tickets in, newest first, same shape as Part 2. `myTickets` tells you how many are theirs.

POST /e-events/ticket-groups/{groupId}/transfer

Move some or all of **your** tickets from this group (`{groupId}`, must be open) into another open group of the **same event and same ticket type**.

```
{ "targetGroupId": "7cle...", "quantity": 1 }
```

Returns the target group. No money moves — the price is the same.

- If the move fills the target group, everyone in it gets their tickets at once.
- If the move empties the source group, it ends with `endReason: EMPTIED` (nobody left to refund).

Errors (400):

- You have 2 tickets in this group, not 5
- Tickets can only move between groups of the same event and ticket type
- Source and target groups must be different
- The target group does not have 1 free tickets
- The target group is no longer open / Tickets can only move out of an open group

POST /e-events/ticket-groups/{groupId}/withdraw

Leave an open group with some or all of your tickets and get them refunded in full straight away.

```
{ "quantity": 1 }
```

Returns the group with the new count (`myTickets` shows what you still hold). The places open up again for others. If nobody is left, the group ends with `endReason: EMPTIED` and its held tickets go back on sale.

To swap only part of your tickets into another group, withdraw them here and join the other group with a new checkout.

Errors (400): You can only leave a group that is still open, You have 2 tickets in this group, not 5.

Cancelling a booking that came from a group

Unchanged endpoint: PATCH /e-events/booking-orders/{bookingId}/cancel. It refunds exactly the tickets in that booking and puts them back on sale.

Part 5 — Notifications

All four open the group: targetType: "EVENT_TICKET_GROUP", targetId = the group id → GET /e-events/ticket-groups/{targetId}. As with every notification, **an unknown targetType stays on the bell list.**

Type	Who gets it	When
EVENT_TICKET_GROUP_CREATED	The buyer who started it	Group opened — prompt them to share the code
EVENT_TICKET_GROUP_JOINED	Everyone in the group	Someone joined or moved tickets in — refresh the "2 of 3"
EVENT_TICKET_GROUP_COMPLETED	Everyone in the group	Full — tickets are in My bookings
EVENT_TICKET_GROUP_FAILED	Everyone still in the group	Ended without filling — refunded in full

The refund itself also sends the existing ESCROW_REFUNDED notice, once per person (not once per ticket).