Part 2D — Agents and live calls

← Contents · Previous: Part 2C — Calls · Next: Part 2E — App actions

People who answer calls. Ringmaster keeps only what it needs to ring them: a `ref` (the client app's user id), a display name, an optional mobile, their departments and their status. How the browser desk works end to end: Part 1 §1.7; the messages the desk receives: Part 3.

All endpoints: ◫ Protected · Bearer API key. The agent's own buttons go **through the client app's backend**, which adds `agentRef` after checking who is logged in — the browser never holds the API key.

```
Set up (once)        I1 department · I7 agent · I5 members
Agent's shift        I12 lobby token → join lobby → I10 status AVAILABLE
A caller arrives     (engine rings: "ring" signal in the lobby)
                     J1 accept → join call room · or J2 decline
On the call          J3 hold · J4 resume · J5 transfer · J6 hang up
After the call       WRAP_UP for wrapUpSec → AVAILABLE (or I11 ready)
End of shift         I10 status OFFLINE → leave lobby
```

I. Departments and agents

I1. Create a department

Endpoint: POST `{base_url}/accounts/{accountId}/departments`

Request JSON Sample:

```
{
  "name": "Mapokezi",
  "strategy": "RING_ALL",
  "ringTimeoutSec": 20,
  "wrapUpSec": 30,
  "maxMissedBeforeAway": 5,
  "ifNobodyOnline": "WAIT",
```

```
  "record": false
}
```

Parameter	Type	Required	Description	Validation / default
`name`	string	Yes	Shown to callers' agents and in reports	not blank
`strategy`	string	No	`RING_ALL` (every free member at once) · `ROUND_ROBIN` (one at a time, in member order, starting after the last one rung)	`RING_ALL`
`ringTimeoutSec`	integer	No	How long one ring lasts before the next try	10–60, default 20
`wrapUpSec`	integer	No	Quiet time after each call before the agent is rung again	0–300, default 0
`maxMissedBeforeAway`	integer	No	Missed or declined rings in a row before the agent is set `AWAY` ; `0` = never	0–20, default 5
`ifNobodyOnline`	string	No	`WAIT` (caller waits in the waiting room) · `EXIT_NO_ANSWER` (the Connect block's `noAnswer` exit at once)	`WAIT`
`record`	boolean	No	Record talk in this department (when the account's recording mode allows)	`true`

Success Response JSON Sample (`201 Created`, `message` "Department created"):

```
{
  "success": true,
  "httpStatus": "201 CREATED",
  "message": "Department created",
  "action": null,
  "action_time": "2026-11-13T20:10:47.69580505",
  "data": {
    "id": "c09c1657-d18a-4a32-94c7-4ffa24e038c3",
    "name": "Mapokezi",
```

```
    "strategy": "RING_ALL",
    "ringTimeoutSec": 20,
    "wrapUpSec": 0,
    "maxMissedBeforeAway": 5,
    "ifNobodyOnline": "WAIT",
    "record": false,
    "members": [],
    "waiting": 0
  }
}
```

Field	Description
`id`	Put it in a flow's Connect block: `"department": "<id>"`
`members`	Agent refs in ring order
`waiting`	Callers waiting now

Errors:

```
{
  "success": false,
  "httpStatus": "422 UNPROCESSABLE_CONTENT",
  "message": "Validation failed",
  "code": "VALIDATION_FAILED",
  "action_time": "2026-11-13T20:10:48.000000000",
  "data": { "ringTimeoutSec": "must be greater than or equal to 10" }
}
```

I2. List departments

Endpoint: `GET` `{base_url}/accounts/{accountId}/departments`

Success Response: `data` = array of departments (I1 shape), by name.

I3. Update a department

Endpoint: `PUT` `{base_url}/accounts/{accountId}/departments/{departmentId}`

Request: same as I1; `name` is required, missing settings keep their value. **Success Response**: the department, `message` "Department updated". **Errors**: `404` "Department not found".

I4. Delete a department

Endpoint: DELETE `{base_url}/accounts/{accountId}/departments/{departmentId}`

Success Response: `200`, `message` "Department deleted", `data` null. Its members are removed from it.

Errors:

```
{
  "success": false,
  "httpStatus": "409 CONFLICT",
  "message": "Department is used by live script versions",
  "code": "DEPARTMENT_IN_USE",
  "action_time": "2026-11-13T20:11:00.000000000",
  "data": { "usedBy": ["recorded-music v2"] }
}
```

`409 DEPARTMENT_BUSY` "Callers are still waiting in this department" — try again when the queue is empty.

I5. Set a department's members

Purpose: Replace the whole member list; the order is the round-robin order. Callers already waiting are offered to new free members at once.

Endpoint: PUT `{base_url}/accounts/{accountId}/departments/{departmentId}/members`

Request JSON Sample:

```
{ "agentRefs": ["asha", "juma", "neema"] }
```

Parameter	Type	Required	Description
`agentRefs`	string[]	Yes	Existing agents of the account; `[]` empties the department; repeats are ignored

Success Response: the department, `message` "Members set".

Errors: `404` "Agent not found: ghost" — nothing is changed.

I6. Callers waiting in a department

Endpoint: **GET** `{base_url}/accounts/{accountId}/departments/{departmentId}/queue`

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "200 OK",
  "message": "Waiting callers",
  "action": null,
  "action_time": "2026-11-13T20:10:47.790000000",
  "data": [
    { "callId": "8a5e282d-93a1-443b-81f3-4eb7294a0639", "from": "+255745051250", "position":
1, "waitingSeconds": 27 }
  ]
}
```

Order: callers put back at the head (a failed transfer, an agent who dropped) first, then by waiting time.

I7. Add or update an agent

Endpoint: **PUT** `{base_url}/accounts/{accountId}/agents/{ref}`

Request JSON Sample:

```
{ "displayName": "Asha (mapokezi)", "mobile": "+255712345678" }
```

Parameter	Type	Required	Description	Validation
`{ref}` (path)	string	Yes	The client app's user id	
`displayName`	string	Yes	Shown in the desk, wallboard and reports	not blank

Parameter	Type	Required	Description	Validation
`mobile`	string	No	Rung when the number's ring mode includes `MOBILE`; `""` removes it; missing keeps it	a phone number

Success Response JSON Sample (`message` "Agent saved"; a new agent starts `OFFLINE`):

```
{
  "success": true,
  "httpStatus": "200 OK",
  "message": "Agent saved",
  "action": null,
  "action_time": "2026-11-13T20:10:47.726488473",
  "data": {
    "ref": "asha",
    "displayName": "Asha (mapokezi)",
    "mobile": null,
    "status": "AVAILABLE",
    "online": true,
    "missedInRow": 0,
    "currentCallId": null,
    "departments": ["c09c1657-d18a-4a32-94c7-4ffa24e038c3"]
  }
}
```

Field	Description
`status`	`OFFLINE` · `AVAILABLE` · `RINGING` · `BUSY` · `WRAP_UP` · `AWAY` (Part 1 §1.7)
`online`	The agent's browser is connected to the lobby now
`missedInRow`	Missed or declined rings since the last answered one
`currentCallId`	The call ringing or live with them
`departments`	Department ids they are a member of

Errors: `422` `{"displayName": "must not be blank"}` · `400` "mobile: ..." (not a phone number).

I8. List agents

Endpoint: `GET` `{base_url}/accounts/{accountId}/agents`

Success Response: `data` = array of agents (I7 shape), by display name. Use it for a wallboard together with `agent.status` webhooks.

I9. Remove an agent

Endpoint: `DELETE` `{base_url}/accounts/{accountId}/agents/{ref}`

Success Response: `200`, `message` "Agent removed". They leave every department and are never rung again; a call in progress with them finishes normally. **Errors**: `404` "Agent not found: ...".

I10. Set an agent's status

Purpose: The agent's own buttons: **Available**, **Away** (break), **Offline** (end of shift). `RINGING`, `BUSY` and `WRAP_UP` are set by calls, never by hand.

Endpoint: `PUT` `{base_url}/accounts/{accountId}/agents/{ref}/status`

Request JSON Sample:

```
{ "status": "AVAILABLE" }
```

`status`	Effect
`AVAILABLE`	Can be rung; `missedInRow` back to 0; a caller already waiting is offered at once
`AWAY`	Online but not rung (a ring in progress is cancelled)
`OFFLINE`	Not rung

Success Response: the agent, `message` "Status set". An `agent.status` webhook follows.

Errors:

- `400` "Status can be set to AVAILABLE, AWAY or OFFLINE"
- `409 AGENT_BUSY` "Finish the current call first"

> Being **online** (connected to the lobby) is separate from the status: an agent whose browser is gone for more than 30 s (and who is not on a call) is set

> `OFFLINE` by the engine.

I11. Ready (end wrap-up early)

Endpoint: **POST** `{base_url}/accounts/{accountId}/agents/{ref}/ready`

Success Response: the agent (`AVAILABLE` if they were in `WRAP_UP`), `message` "Ready".

I12. Lobby token

Purpose: Lets the agent's browser join their **lobby room**, where the engine sends ring signals. Data only: no microphone. Valid for 10 minutes; ask for a new one before it runs out (an open connection stays up).

Endpoint: **POST** `{base_url}/accounts/{accountId}/agents/{ref}/lobby-token`

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "200 OK",
  "message": "Lobby token",
  "action": null,
  "action_time": "2026-11-13T20:10:47.815393198",
  "data": {
    "url": "wss://<livekit-host>",
    "room": "lobby-0ea57aeb-c3c9-48f5-b366-7fe3e457ffbd",
    "identity": "agent-0ea57aeb-c3c9-48f5-b366-7fe3e457ffbd:web-e2dc8c22",
    "token": "eyJhbGciOiJIUzI1NiIsInR5…"
  }
}
```


In the browser (LiveKit JS SDK):

```
const room = new LivekitClient.Room();
room.on(LivekitClient.RoomEvent.DataReceived, bytes => onSignal(JSON.parse(new
TextDecoder().decode(bytes))));
```

```
await room.connect(data.url, data.token);           // then PUT …/status {AVAILABLE} through
the backend
```

Make the browser's ring sound start from the agent's "Start shift" click: browsers mute sound that did not start from a click.

Each token has **its own identity** (`agent-{id}:web-{random}`), so a second tab, or the browser next to Agent Phone (`…:dev-{deviceId}`), never pushes the first one out. The agent is **online while any of their connections is in the lobby**; the 30 s grace starts when the last one leaves.

I13. Agent stats

Endpoint: `GET` `{base_url}/accounts/{accountId}/agents/stats`

Query	Type	Required	Description	Default
`from` / `to`	date	No	`YYYY-MM-DD`, account time	the last 7 days

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "200 OK",
  "message": "Agent stats",
  "action": null,
  "action_time": "2026-11-13T20:10:47.759658701",
  "data": [
    { "key": "asha", "values": { "offered": 7, "answered": 3, "missed": 1, "declined": 3,
"declinedWithCustomer": 2,
                                "declinedBreak": 1, "declinedWrongDepartment": 0,
"declinedOther": 0, "talkSeconds": 50 } }
  ]
}
```

`values`	Description
`offered`	Rings
`answered`	Accepted
`missed`	Rang until the ring time ran out

`values`	Description
`declined`	Pressed Decline (all reasons, and declines without a reason)
`declinedWithCustomer` · `declinedBreak` · `declinedWrongDepartment` · `declinedOther`	Declines by the reason the agent tapped (J2)
`talkSeconds`	Time talking to callers

I14. The agent's desk

Purpose: Everything an agent's device shows, in one call: status, departments with their waiting callers and team, today's numbers, recent calls, the business's lines and the Agent Phone switches. Fetch it on start, on reconnect and when a `config` signal arrives; between those the `status`, `team` and `queue` signals keep it live. Agent Phone gets the same through L3.

Endpoint: `GET` `{base_url}/accounts/{accountId}/agents/{ref}/desk`

Success Response JSON Sample:


```
"data": {
  "agent": { "ref": "asha", "name": "Asha", "status": "AVAILABLE", "online": true, "since":
"2026-11-14T07:53:57Z" },
  "settingsVersion": 3,
  "business": "Ringmaster Lab",
  "privacy": { "showTeam": true, "showCallerNumbers": true, "allowDecline": true,
"allowCallBack": true, "allowDialPad": true },
  "lines": [ { "id": "8a8d9c13-bda2-4a18-8410-c5ec49ed2c66", "e164": "+255677000001", "label":
null } ],
  "departments": [ {
    "id": "c09c1657-d18a-4a32-94c7-4ffa24e038c3", "name": "Mapokezi", "ringTimeoutSec": 20,
"wrapUpSec": 0,
    "waiting": 1, "oldestSince": "2026-11-14T08:17:47Z",
    "callers": [ { "callId": "5d382db0-…", "position": 1, "from": "+255745051250", "name":
"John Mushi", "since": "2026-11-14T08:17:47Z" } ],
    "team": [ { "ref": "neema", "name": "Neema", "status": "BUSY", "online": true } ]
  } ],
  "today": { "answered": 9, "missed": 0, "declined": 1, "talkSec": 146 },
  "recent": [ { "callId": "fb40f1bf-…", "direction": "INBOUND", "number": "+255745051250",
"name": "John Mushi",
```

```
                "at": "2026-11-14T08:17:21Z", "seconds": 56, "result": "HANDLED_BY_AGENT" } ]
}
```


Field	Description
`departments[].callers`	Up to 10 waiting callers in order; `name` when the contacts URL knew them
`departments[].team`	Other members; `null` when the account hides the team (`showTeam: false`)
`recent`	The agent's last 10 calls (in and out)
numbers	Masked as `+2557•••••250` everywhere when `showCallerNumbers` is `false`

I15. Search for the dial pad

Endpoint: `GET` `{base_url}/accounts/{accountId}/agents/{ref}/desk/search?digits=john`

Up to 20 matches: first the client app's contacts (asked through its contacts URL, by name or number), then numbers from this agent's and their departments' calls.


```
"data": [ { "number": "+255745051250", "name": "John Mushi", "label": "VIP customer",
"callId": "fb40f1bf-…",
            "direction": "INBOUND", "at": "2026-11-14T08:17:21Z", "calls": 22 } ]
```


`callId` is the latest call with that number (use J8); `null` for a contact never called. When numbers are hidden, masked numbers match only on the digits the agent can see, and contacts are left out.

J. Live call controls

Every control takes the agent: `{"agentRef": "asha"}` (sent by the client app's backend). Controls for a call that is not ringing for, or not live with, that agent are refused.

J1. Accept a ringing call

Endpoint: `POST` `{base_url}/calls/{callId}/accept`

Request JSON Sample:

```
{ "agentRef": "asha" }
```

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "200 OK",
  "message": "Call accepted",
  "action": null,
  "action_time": "2026-11-13T19:43:46.998000000",
  "data": {
    "url": "wss://<livekit-host>",
    "room": "call-_+255745051250_uqL6Uaw9VmC9",
    "identity": "agent-0ea57aeb-c3c9-48f5-b366-7fe3e457ffbd",
    "token": "eyJhbGciOiJIUzI1NiIsInR5…"
  }
}
```

The browser joins `room` with `token` and turns the microphone on; the caller's waiting music stops and they are connected as soon as the agent is in the room (the engine waits up to 15 s; if the agent never arrives, the caller goes back to the head of the queue). Other agents rung for the same call get `ring_cancel` (`answered_elsewhere`).

Errors:

- `409 NOT_RINGING` "This call is not ringing for you any more" (ring time ran out, the caller left, or it was cancelled)
- `409 ALREADY_TAKEN` "Another agent already answered this call"
- `404` "Call not found" · "Agent not found: …"

J2. Decline a ringing call

Endpoint: **POST** `{base_url}/calls/{callId}/decline`

Request: `{"agentRef": "asha", "reason": "WITH_CUSTOMER"}`. `reason` is optional: `WITH_CUSTOMER`, `BREAK` (also sets the agent `AWAY`), `WRONG_DEPARTMENT`, `OTHER`.

Success Response: `200`, `message` "Call declined". The engine rings the next agent; this agent is not rung for this call again for 30 s. If no other agent could take it (reachable, not `AWAY`/`OFFLINE`, not declined), the caller leaves the queue at once by the Connect block's `noAnswer` exit. The decline counts as a missed ring (I1 `maxMissedBeforeAway`) and sends `call.ring_declined`.

Errors: `409 NOT_RINGING` · `409 DECLINE_NOT_ALLOWED` (the account turned `agentPhone.allowDecline` off).

J3. Hold

Purpose: The caller hears hold music; the agent's microphone is muted for the caller.

Endpoint: **POST** `{base_url}/calls/{callId}/hold`

Request: `{"agentRef": "asha"}`. **Success Response**: `200`, `message` "On hold". Holding twice is harmless.

Errors:


```
{
  "success": false,
  "httpStatus": "409 CONFLICT",
  "message": "This call is not with you",
  "code": "NOT_YOUR_CALL",
  "action_time": "2026-11-13T20:10:57.000000000",
  "data": null
}
```


J4. Resume from hold

Endpoint: **POST** `{base_url}/calls/{callId}/unhold`

Request: `{"agentRef": "asha"}`. **Success Response**: `200`, `message` "Back from hold". **Errors**: `409 NOT_YOUR_CALL`.

J5. Transfer

Purpose: Hand the caller to another department or a named agent (blind transfer: the first agent leaves at once, the caller hears music meanwhile).

Endpoint: **POST** `{base_url}/calls/{callId}/transfer`

Request JSON Samples:

```
{ "agentRef": "asha", "toDepartmentId": "dbd644af-e460-40fa-9b09-4cb838915bdc" }
```

```
{ "agentRef": "asha", "toAgentRef": "juma" }
```

Parameter	Type	Required	Description
`agentRef`	string	Yes	The agent now on the call
`toDepartmentId`	uuid	One of the two	The caller goes to the **head** of that department's queue
`toAgentRef`	string	One of the two	That agent is rung; if they don't answer, the caller's current department picks them up (they are at its head)

Success Response: `200`, `message` "Transferring". The call's timeline gets a `TRANSFER` event (`{from, to}`).

Errors: `400` "Give toDepartmentId or toAgentRef" · `409 NOT_YOUR_CALL` · `404` "Agent not found: ...".

J6. Hang up

Endpoint: **POST** `{base_url}/calls/{callId}/hangup`

Request: `{"agentRef": "asha"}`. **Success Response**: `200`, `message` "Hanging up". The caller is disconnected; the call ends with `result: HANDLED_BY_AGENT`, `endReason: AGENT_HUNG_UP`, `endedBy: AGENT`; the agent goes to `WRAP_UP` (or straight to `AVAILABLE` when `wrapUpSec` is 0).

Errors: `409 NOT_YOUR_CALL`.

On an **outgoing** call that still rings at the other end, hang up **cancels the dial**: the other phone stops ringing.

> **If the agent's browser drops** during a call (closed tab, lost network), the caller is not cut off: they hear music and go back to the head of the queue.

J7. Join an outgoing call

Purpose: An agent call (click-to-call) started anywhere (your dashboard, Agent Phone's dial pad or a call back) sends `outgoing` to the agent's devices; the device then joins with this.

Endpoint: `POST` `{base_url}/calls/{callId}/join`

Request: `{"agentRef": "asha"}`. **Success Response**: `200`, the call room token (`url`, `room`, `identity`, `token`). Follow `call_answered`, `call_recording` and `call_ended` (3.3): the other phone's audio arrives while it still rings, so never treat audio as "answered".

Errors: `409 NOT_YOUR_CALL` · `409 CALL_ENDED`.

J8. Call someone back

Purpose: Call the other person of an earlier call (a caller, or someone called before) **from the same business line**. It works even when numbers are hidden from agents, because the agent never needs the number.

Endpoint: `POST` `{base_url}/calls/{callId}/call-back`

Request: `{"agentRef": "asha"}`. **Success Response**: `201`, like click-to-call (`callId`, `room`); the agent's devices get `outgoing`.

Errors: `409 CALL_BACK_OFF` (the account turned `agentPhone.allowCallBack` off) · `409 NO_LINE` · the click-to-call errors (`AGENT_BUSY`, `CAN_CALL_REFUSED`).

J9. Take a waiting caller

Purpose: Answer a caller from the queue list **now**, even while the agent is `AWAY` (they see someone waiting and pick them up). Only callers waiting for one of the agent's departments, and not while the agent is on a call or ringing. Other agents' rings for that caller stop; the caller leaves the queue; then it is exactly like J1 (the answer is the call room token).

Endpoint: `POST` `{base_url}/calls/{callId}/take`

Request: `{"agentRef": "asha"}`. A `TAKEN_FROM_QUEUE` event goes on the timeline.

Errors: `409 NOT_WAITING` · `409 NOT_YOUR_DEPARTMENT` · `409 AGENT_BUSY`.

Revision #2
Created 13 November 2026 18:49:29 by Admin Qbit
Updated 14 November 2026 12:25:41 by Admin Qbit