Part 2B — Call flows, media, knowledge base

← Contents · Previous: Part 2A — Setup · Next: Part 2C — Calls

A **call flow** (API name: *script*) is identified by its `ref` inside an account (`orders`, `account-balance`). Its content lives in numbered **versions** that never change once stored. How to write the document itself (blocks, settings, exits, messages, variables): Part 5. Life of a flow: Part 1 §1.6.

All endpoints: ◫ Protected · Bearer API key. `{accountId}` must be one of the client app's accounts (else `404`).

D. Call flows (scripts)

D1. List flows

Purpose: Every flow of an account with all its versions (deleted flows are not listed).

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

Success Response: `data` = array of flows, sorted by `ref`, each as in D3.

D2. Create or rename a flow

Purpose: Make the flow (before its first version), or change its display name. Calling it again with the same `ref` only renames.

Endpoint: `PUT` `{base_url}/accounts/{accountId}/scripts/{ref}`

Request JSON Sample:

```
{ "name": "Account balance", "direction": "INCOMING" }
```

Parameter	Type	Required	Description	Validation
`{ref}` (path)	string	Yes	The client app's id for the flow	max 120; letters, digits, `-`, `_`
`name`	string	Yes	Display name	not blank
`direction`	string	Yes	`INCOMING` (plays on a number) · `OUTGOING` (plays when a called person answers)	fixed once created

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "200 OK",
  "message": "Script saved",
  "action": null,
  "action_time": "2026-11-13T20:00:15.876328395",
  "data": {
    "id": "d3c063a6-1d76-415e-b623-c5769c1f6e69",
    "ref": "docs-sample",
    "name": "Account balance",
    "direction": "INCOMING",
    "versions": []
  }
}
```

Errors: `409 DIRECTION_FIXED` "A script keeps its direction; duplicate it to use the other direction". A flow that was deleted after taking calls (archived) comes back with its old versions, which stay stopped.

D3. Get a flow

Endpoint: GET `{base_url}/accounts/{accountId}/scripts/{ref}`

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "200 OK",
  "message": "Script",
```

```
  "action": null,
  "action_time": "2026-11-13T20:00:42.882898745",
  "data": {
    "id": "d3c063a6-1d76-415e-b623-c5769c1f6e69",
    "ref": "docs-sample",
    "name": "Account balance",
    "direction": "INCOMING",
    "versions": [
      {
        "versionId": "65a926f5-9fdc-49ed-95a7-93a70e562460",
        "version": 1,
        "runnable": true,
        "testOnly": false,
        "revoked": false,
        "vars": [],
        "texts": ["Bonyeza namba yako ya akaunti.", "Asante, kwaheri."],
        "riskyBlocks": ["acct"],
        "warnings": [],
        "createdAt": "2026-11-13T17:00:15.991897Z"
      },
      {
        "versionId": "15baa310-3997-4d14-a66e-a9c024f909ec",
        "version": 2,
        "runnable": false,
        "testOnly": false,
        "revoked": false,
        "vars": [],
        "texts": ["Bonyeza namba yako ya akaunti.", "Asante, kwaheri."],
        "riskyBlocks": ["acct"],
        "warnings": [],
        "createdAt": "2026-11-13T17:00:42.855216Z"
      }
    ]
  }
}
```


Field	Description
`versions[].versionId`	What numbers (`PUT /v1/numbers/{id}/script`) and call-outs (`scriptVersionId`) point at
`versions[].version`	The number the client app gave it

Field	Description
`runnable`	May run: be put on a number or used for calls out
`testOnly`	For test calls only; never runnable
`revoked`	Stopped for good
`vars`	Variables a call-out must give each person (`name`, `amount`); system variables are not listed
`texts`	Fixed sentences, already turned into audio. Sentences with variables (`Akaunti {input.account}…`) are spoken during the call
`riskyBlocks`	Blocks that make the version need approval (D7)
`warnings`	Only in the answer to a push (D5); empty when read later

Errors: `404 NOT_FOUND` "Script not found: docs-sample".

D4. Delete a flow

Purpose: Remove a flow. Refused while anything uses it. A flow that never ran is erased; a flow that ran calls is **archived**: hidden from lists, all its versions stopped, its call history kept. Creating the same `ref` again (D2) brings it back.

Endpoint: **DELETE** `{base_url}/accounts/{accountId}/scripts/{ref}`

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "200 OK",
  "message": "Script deleted",
  "action": null,
  "action_time": "2026-11-13T20:00:43.15694553",
  "data": { "ref": "docs-sample", "outcome": "DELETED" }
}
```

Field	Description
`outcome`	`DELETED` (never ran: gone) · `ARCHIVED` (ran calls: hidden, history kept)

Error Response JSON Sample (still in use):

```
{
  "success": false,
  "httpStatus": "409 CONFLICT",
  "message": "The script is still in use: move these to another script first",
  "code": "SCRIPT_IN_USE",
  "action_time": "2026-11-13T18:09:38.582694597",
  "data": {
    "numbers": ["+255677000003"],
    "callouts": ["Callback +255745051250"],
    "scripts": ["account v4"]
  }
}
```


`data` key	Means
`numbers`	Numbers playing one of its versions → put another flow on them (2A C7)
`callouts`	Call-outs not finished (draft, scheduled, running, paused) using it → cancel or let them finish
`scripts`	Live versions of other flows that transfer to it or schedule callbacks with it → push new versions without that

---

D5. Push a version

Purpose: Store a new version. The engine checks it first (Part 5 §5.9): errors → nothing stored. Fixed sentences are turned into audio straight away, and an AI Agent's knowledge is prepared in the background.

Endpoint: **POST** `{base_url}/accounts/{accountId}/scripts/{ref}/versions`

Request JSON Sample:


```
{
  "version": 2,
  "runnable": false,
  "definition": {
    "direction": "INCOMING",
    "start": { "call": "acct" },
    "blocks": {
```

```
      "acct": { "type": "collect", "msg": { "src": "text", "text": "Bonyeza namba yako ya
akaunti." },
                "saveAs": "account", "minDigits": 4, "maxDigits": 4,
                "x": { "ok": "read", "failed": "bye", "later": "bye" } },
      "read": { "type": "say", "msg": { "src": "text", "text": "Akaunti {input.account} ina
salio la shilingi elfu kumi." },
                "x": { "next": "bye" } },
      "bye":  { "type": "goodbye", "msg": { "src": "text", "text": "Asante, kwaheri." } }
    }
  }
}
```

Parameter	Type	Required	Description	Validation
`version`	integer	Yes	The version number; the client app counts	≥ 1, unique per flow
`definition`	object	Yes	The flow document	must pass the check; `direction` must match the flow; a `layout` key is dropped
`runnable`	boolean	No	`true` = may run at once. Push `false` when the version has `riskyBlocks` and waits for approval	default `false`
`testOnly`	boolean	No	For test calls only (never runnable)	default `false`

Success Response JSON Sample (`201 Created`):

```
{
  "success": true,
  "httpStatus": "201 CREATED",
  "message": "Version stored",
  "action": null,
  "action_time": "2026-11-13T20:00:16.018139374",
  "data": {
    "versionId": "65a926f5-9fdc-49ed-95a7-93a70e562460",
    "version": 1,
    "runnable": true,
    "testOnly": false,
    "revoked": false,
```

```
    "vars": [],
    "texts": ["Bonyeza namba yako ya akaunti.", "Asante, kwaheri."],
    "riskyBlocks": ["acct"],
    "warnings": [
      { "level": "WARNING", "block": "acct", "code": "EXIT_UNCONNECTED", "message": "Exit
'later' is not connected: the call ends there" }
    ],
    "createdAt": "2026-11-13T17:00:15.991897467Z"
  }
}
```


Sending the same version again with the same document (e.g. a retry after a timeout) is safe: `200 OK`, `message` "Version already stored", same `data`. Secrets in Send request blocks are encrypted before storing and do not change the comparison.

Error Response JSON Samples:


```
{
  "success": false,
  "httpStatus": "422 UNPROCESSABLE_CONTENT",
  "message": "Script version is not valid",
  "code": "VALIDATION_FAILED",
  "action_time": "2026-11-13T20:00:16.178937467",
  "data": [
    { "level": "ERROR", "block": "a", "code": "EXIT_TARGET_MISSING", "message": "Exit 'next'
points to a missing block: zz" }
  ]
}
```



```
{
  "success": false,
  "httpStatus": "409 CONFLICT",
  "message": "Version 1 already exists with a different definition; versions are immutable",
  "code": "VERSION_EXISTS",
  "action_time": "2026-11-13T20:00:16.123223371",
  "data": null
}
```


Also `400` "definition must be a JSON object" · `404` "Script not found" (create it with D2 first) · `403 FEATURE_OFF` "AI is not enabled for this client app; these blocks use it: ai" (an AI Agent block or smart speech matching without the `AI` feature). Check codes: Part 4.

D6. Get a version (with its blocks)

Purpose: Read a version back: to show it, edit it into the next version, or see what a call ran.

Endpoint: `GET` `{base_url}/accounts/{accountId}/scripts/{ref}/versions/{version}`

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "200 OK",
  "message": "Script version",
  "action": null,
  "action_time": "2026-11-13T20:00:42.918636988",
  "data": {
    "ref": "docs-sample",
    "name": "Account balance",
    "direction": "INCOMING",
    "version": {
      "versionId": "15baa310-3997-4d14-a66e-a9c024f909ec",
      "version": 2,
      "runnable": false,
      "testOnly": false,
      "revoked": false,
      "vars": [],
      "texts": ["Bonyeza namba yako ya akaunti.", "Asante, kwaheri."],
      "riskyBlocks": ["acct"],
      "warnings": [],
      "createdAt": "2026-11-13T17:00:42.855216Z"
    },
    "definition": {
      "direction": "INCOMING",
      "start": { "call": "acct" },
      "blocks": {
        "acct": { "type": "collect", "msg": { "src": "text", "text": "Bonyeza namba yako ya
akaunti." },
                  "saveAs": "account", "minDigits": 4, "maxDigits": 4,
                  "x": { "ok": "read", "later": "bye", "failed": "bye" } },
        "read": { "type": "say", "msg": { "src": "text", "text": "Akaunti {input.account} ina
```

```
salio la shilingi elfu kumi." },
                  "x": { "next": "bye" } },
        "bye":  { "type": "goodbye", "msg": { "src": "text", "text": "Asante, kwaheri." } }
      }
    }
  }
}
```

Field	Description
`version`	Same as in D3
`definition`	The stored document. Keys come back sorted. Secrets of Send request blocks (`auth.token`, `auth.password`, `auth.value`) come back as `"(hidden)"`. Pushed back unchanged in a new version, `"(hidden)"` keeps the stored secret of the same block

Errors: `404` "Version not found: 9".

D7. Approve a version (make it runnable)

Purpose: After the client app's reviewer approves a version pushed with `runnable: false`. Call-outs that were waiting for it start by themselves.

Endpoint: POST `{base_url}/accounts/{accountId}/scripts/{ref}/versions/{version}/runnable`

Success Response: `200`, `message` "Version is runnable", `data` = the version (`runnable: true`).

Errors:

```
{
  "success": false,
  "httpStatus": "409 CONFLICT",
  "message": "A revoked version cannot become runnable",
  "code": "VERSION_REVOKED",
  "action_time": "2026-11-13T20:00:43.054366231",
  "data": null
}
```

D8. Stop a version (revoke)

Purpose: Stop a version for good (it was wrong, or the business rejected it). It can never run again.

Endpoint: `POST` `{base_url}/accounts/{accountId}/scripts/{ref}/versions/{version}/revoke`

Success Response: `200`, `message` "Version revoked", `data` = the version (`revoked: true`, `runnable: false`). Revoking twice is harmless.

> **Move things off it first.** A number still playing a revoked version refuses calls (callers hear that the service is unavailable), and running call-outs using it are paused with `pauseReason: "VERSION_REVOKED"`.

D9. Block stats of a version

Purpose: Where callers go and where they get stuck, per block — for the flow editor's heat map.

Endpoint: `GET` `{base_url}/accounts/{accountId}/scripts/{ref}/versions/{version}/stats`

Query	Type	Required	Description	Default
`from`	date	No	First day, `YYYY-MM-DD` (account's time zone)	6 days before `to`
`to`	date	No	Last day	today

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "200 OK",
  "message": "Block stats",
  "action": null,
  "action_time": "2026-11-13T20:01:13.597312525",
  "data": [
    { "key": "acct", "values": { "visits": 5, "hangups": 0, "noInput": 0, "retries": 2,
                                  "reqOk": 0, "reqFail": 0, "reqTimeout": 0, "reqMsP50": null,
"gaveUp": 0 } },
    { "key": "when", "values": { "visits": 4, "hangups": 0, "noInput": 0, "retries": 1,
                                  "reqOk": 0, "reqFail": 0, "reqTimeout": 0, "reqMsP50": null,
```

```
"gaveUp": 0 } }
  ]
}
```

Field	Description
`key`	Block id
`visits`	Times a call entered the block
`hangups`	Callers who hung up in it
`noInput`	Times nobody answered its question
`retries`	Questions asked again (wrong or no answer)
`reqOk` / `reqFail` / `reqTimeout` / `reqMsP50`	Send request blocks: results and median time in ms
`gaveUp`	Waiting rooms the caller left after the give-up time

Stats are rolled up a few minutes after calls end.

E. Media (audio files)

Recorded greetings, menu prompts and hold music. Upload once, then use the id in a flow message: `{"src": "upload", "media": "<mediaId>"}` (a waiting room's `music` takes the id directly).

E1. Upload audio

Endpoint: **POST** `{base_url}/accounts/{accountId}/media` · `multipart/form-data`

Part / query	Type	Required	Description	Validation
`file` (part)	file	Yes	MP3, WAV, OGG, M4A…	an audio file; ≤ 10 MB; ≤ 5 minutes
`name` (query)	string	No	Display name	default: the file name
`lang` (query)	string	No	`sw` · `en` (for the library filter)	

```
curl -X POST "$BASE/accounts/$ACCOUNT/media?name=Salamu%20ya%20duka&lang=sw" \
  -H "Authorization: Bearer $RINGMASTER_KEY" -F "file=@salamu.mp3"
```

Success Response: `202 Accepted`, `message` "Audio received; processing", `data` = the media item with `status: "PENDING"`. The engine converts it for the phone line in the background, then sends

`media.ready` (`{mediaId, durationMs}`) or `media.failed` (`{mediaId, error}`) — Part 3. Use the id in a flow only once it is `READY`.

Errors:

```
{
  "success": false,
  "httpStatus": "422 UNPROCESSABLE_CONTENT",
  "message": "Upload rejected",
  "code": "VALIDATION_FAILED",
  "action_time": "2026-11-13T20:01:13.708314472",
  "data": [ { "field": "file", "message": "Not an audio file" } ]
}
```

Other messages: "File is larger than 10 MB" · "Only audio files are accepted" · "Audio is longer than 5 minutes". `409 STORAGE_QUOTA` — the account has used its 500 MB of audio.

E2. List audio

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

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "200 OK",
  "message": "Audio library",
  "action": null,
  "action_time": "2026-11-13T20:01:13.551808448",
  "data": [
    {
      "id": "eebb855e-2a03-4af7-a8c8-301f85877308",
      "name": "Muziki wa kusubiri",
      "kind": "UPLOAD",
      "status": "READY",
      "durationMs": 24192,
      "lang": "sw",
      "error": null,
      "previewUrl": "https://<ringmaster-host>/media/eebb855e-
```

```
…/eda4fdcb….mp3?t=f702ec7bb40e96291e689462aa8319c9",
      "createdAt": "2026-11-13T09:46:03.541625Z"
    }
  ]
}
```


Field	Description
`kind`	`UPLOAD` (uploaded) · `RECORDING` (recorded in the editor) · `SYSTEM` (engine's own: beeps, default hold music)
`status`	`PENDING` → `PROCESSING` → `READY` · `FAILED` (see `error`)
`durationMs`	Length once ready
`previewUrl`	Signed link to play it in a browser (`<audio src>`), no API key needed; it expires — read it again rather than storing it

---

E3. Get one audio item

Endpoint: `GET` `{base_url}/media/{mediaId}`

Success Response: one item as in E2. **Errors**: `404` "Audio not found".

E4. Delete audio

Endpoint: `DELETE` `{base_url}/media/{mediaId}`

Success Response: `200`, `message` "Audio deleted", `data` null.

Errors: `404` "Audio not found" · `409 MEDIA_IN_USE` "Audio is used by live script versions", `data: {"usedBy": ["orders v3", "reception v2"]}` — push versions without it (or revoke those) first.

F. Knowledge base (for AI Agents)

Documents an AI Agent may answer from. Upload them, then list their ids in the AI Agent block's `kb.docs` (Part 5 §5.7). Each flow version builds its own searchable copy when it is stored, so a version keeps answering the same way even if documents change later; push a new version to use new documents.

F1. Upload a document

Endpoint: **POST** `{base_url}/accounts/{accountId}/kb-docs` · `multipart/form-data`

Part	Type	Required	Description	Validation
`file`	file	Yes	`.txt`, `.md` or `.pdf` with text	≤ 10 MB; text is cut at 200,000 characters; scanned PDFs (images) have no text

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "201 CREATED",
  "message": "Document stored",
  "action": null,
  "action_time": "2026-11-13T20:05:00.000000000",
  "data": { "id": "7c2e…", "name": "bei_2026.pdf", "contentType": "application/pdf", "chars":
18420,
            "status": "READY", "createdAt": "2026-11-13T17:05:00Z" }
}
```

Errors (`400 BAD_REQUEST`): "A file is required" · "Documents are limited to 10 MB" · "Upload a .txt, .md or .pdf document" · "No text found in this document (scanned PDFs are not read)".

F2. List documents

Endpoint: **GET** `{base_url}/accounts/{accountId}/kb-docs`

Success Response: `data` = array of documents as in F1.

F3. Delete a document

Endpoint: DELETE `{base_url}/accounts/{accountId}/kb-docs/{documentId}`

Success Response: `200`, `message` "Document removed", `data` null. Versions already stored keep their copy of its text. **Errors**: `404` "Document not found".

Revision #2
Created 13 November 2026 18:48:04 by Admin Qbit
Updated 13 November 2026 19:01:21 by Admin Qbit