

Additional Configuration Operations

Additional Configuration Operations

OpenAPI 1.2.0 adds the Configuration/Settings resources and related object operations listed below. All paths are relative to `https://pbx.example.com/pbx/openapi.php`. Supply the API key in `X-API-Key` and select the tenant with `?tenant=TENANT_CODE`.

Collection and object operations

Method and path	Operation
GET /resource	List the selected scope.
POST /resource	Create; returns HTTP 201 and the saved object.
GET /resource/{id}	Read one object.
PUT or PATCH /resource/{id}	Update supplied fields, preserving omitted fields.
DELETE /resource/{id}	Delete one object; returns deleted=true.

These new lists accept `limit` (default 100, maximum 1000), `offset` (default 0), and equality filters using the documented short aliases, except secret, binary and template content fields. Use the returned `id` for item operations. A list returns an array; object responses include the native database fields, `id`, and `object`. Unknown or read-only fields are rejected. Only documented aliases or their corresponding native column names are accepted.

New-item defaults apply to omitted writable fields on creation: installation defaults first, then tenant defaults; explicit request values take precedence. IDs, tenant ownership and worker-maintained fields cannot be supplied. Related IDs are checked in the selected scope. Use a full key for writes; read-only keys can list and read metadata, with credentials and binary downloads

withheld. Quick shortcuts require a global key because they belong to users.

Each mutation uses a checked transaction and requires InnoDB for all tables it modifies. A failure returns an error with `code`, `message` and `rolled_back`. Database errors report the number and, when available, the column, without exposing SQL or secret values. A post-commit activity-log failure returns a successful response with a warning. Use `global=yes` only on the explicitly listed resources, with a global key.

Query-string routing is also supported, for example

`openapi.php?object=manageruser&action=create&tenant=TENANT_CODE` with POST JSON, or `object=manageruser&id=8` with GET/PATCH/DELETE. JSON request bodies contain the object fields, not routing parameters.

Available resources

Collection	Required fields when no new-item default supplies them	Purpose
<code>/managerusers</code>	name, password	manageruser
<code>/quickshortcuts</code>	us_id, name, link	quickshortcut
<code>/pickupgroups</code>	number, name	pickupgroup
<code>/internal routings</code>	regexp, ex_id	internalrouting
<code>/parkingreturns</code>	number, dialback	parkingreturn
<code>/parkinglots</code>	name, start, end	parkinglot
<code>/callprivacyrules</code>	callerid, action	callprivacy
<code>/autoanswers</code>	field	autoanswer
<code>/dialbynamedirectories</code>	name	dialbynamedirectory
<code>/dialbynameentries</code>	dbn_id, type, type_id, number	dialbynameentry
<code>/digitalfaxes</code>	number	digitalfax
<code>/alarmbindings</code>	callerid	alarmbinding
<code>/mailtofaxes</code>	name, host, username	mailtofax
<code>/mailtocalls</code>	name, host, username, authtype	mailtocall
<code>/scheduledcalls</code>	tobecalled	scheduledcall
<code>/dashboards</code>	name	dashboard
<code>/dashboarditems</code>	td_id, code	dashboarditem
<code>/linemonitors</code>	name	linemonitor

Collection	Required fields when no new-item default supplies them	Purpose
/xmltemplates	name, xmltemplate	xmltemplate
/causecodes	code	causecode
/cnamcacheentries	number, name	cnamcacheentry
/extensiondialings	from, to	extensiondialing
/translationrooms	name, basenumber	translationroom
/translationlanguages	tr_id, number, language	translationlanguage
/customdestinationinfos	cu_id	customdestinationinfo
/customdestinationfiles	cu_id, name, binaryfile	customdestinationfile
/provisioningvalues	ph_id, type, al_id	provisioningvalue
/phonebuttonlayouts	ph_id, bl_id	phonebuttonlayout
/phonebookassignments	ph_id, pb_id	phonebookassignment
/featuremappings	code	featuremapping

managerusers

Manager users are owned by the selected tenant. Read-only keys do not receive passwords.

Writable fields: `name, password, description`.

quickshortcuts

Global API key required, including reads. `user_id` is a real user ID, or 0 for default shortcuts. Links must be relative or HTTP(S).

Writable fields: `us_id, name, order, icon_stack1, icon_stack2, icon_stack3, link, user_id`.

pickupgroups

number identifies the pickup/call group; the name is unique to that tenant and number.

Writable fields: `number, name`.

internal routings

extension_id must belong to the selected tenant. regexp must be a valid regular expression; deldigits is non-negative.

Writable fields: `regexp, ex_id, adddigits, deldigits, extension_id`.

parkingreturns

number and dialback are strings; leading zeros are preserved.

Writable fields: `number, dialback, comment`.

parkinglots

The tenant-code prefix is automatically added to name. start and end are numeric and inclusive (maximum 10001 spaces per API-created range). hosted must name an existing server. moh is 0 or tenant/global music on hold.

Writable fields: `name, start, end, hosted, timeout, defaultreturnnumber, moh`.

callprivacyrules

action: allow, kill, torture or deny. data_base64 optionally stores the recorded privacy message. CallerID must be unique per tenant.

Writable fields: `callerid, action, media, datetime, data_base64`.

autoanswers

Global rows are supported with a global key and global=yes. The shared scope is -1.

Writable fields: `field, value, comment`.

dialbynamedirectories

Deleting a directory removes its entries and extension membership links. Routing references block deletion.

Writable fields: `name, description`.

dialbynameentries

directory_id and type_id must be in the selected tenant; type is EXT or VOICEMAIL. number is the directory lookup number.

Writable fields: `dbn_id, type, type_id, number, directory_id`.

digitalfaxes

Use global=yes with a global key for installation-wide records (scope -1).

Writable fields: `number, comment`.

alarmbindings

Bindings apply to callerid in the selected tenant.

Writable fields: `callerid, emailnotification, relayserver`.

mailtofaxes

protocol is /imap or /pop3. transport is /notls, /ssl, /tls, /ssl/novalidate-cert or /tls/novalidate-cert. Defaults without a configured new-item value: /imap, port 143, /notls, folder INBOX and SENDEREMAIL authentication. rp_id is 0 or a response path in this tenant.

Writable fields: `name, email, protocol, host, port, transport, username, password, folder, cidnum, faxquality, authtype, authdata, disabled, language, cidname, pagesize, unauthenticated, automaticrotate, parsedestination, alertemail, rp_id`.

mailtocalls

Mailbox connection fields follow Mail to Fax. authtype is SENDEREMAIL or SENDEREMAILSECRET; the latter requires authsecret (16-128 printable characters, no spaces). An empty secret preserves the existing one. The hash is never returned. Runtime failure counters are read-only. Delete its scheduled calls before deleting a mailbox.

Writable fields: `name, email, protocol, host, port, transport, username, password, folder, cidnum, authtype, authdata, disabled, cidname, unauthenticated, parsedestination, engine, language, voiceapiusername, voiceapipassword, voiceapihost, maxattempts, delaybetweenattempts, requestconfirm, maxdialings, authsecret`.

scheduledcalls

Create with `tobecalled` and `data_base64` containing PCM WAV audio (mono, 8000 Hz, 16 bit). The API computes the audio hash, format and duration; text is descriptive and does not trigger text-to-speech generation. New calls default to READY, three attempts, three dials and a 60-second delay unless new-item defaults override these settings. `notbefore` uses YYYY-MM-DD HH:MM:SS. Only READY/FAILED calls without a held or pending dispatch can be edited or deleted. Worker counters and receipts cannot be set.

Writable fields: `callerid`, `calleridname`, `tobecalled`, `notbefore`, `maxattempts`, `requestconfirm`, `calldelay`, `text`, `mediafile`, `mediafileext`, `mediatimelen`, `maxdials`, `senderemail`, `callstatus`, `data_base64`.

dashboards

Delete removes dashboard items and the corresponding dashboard setting. `global=yes` requires a global key and uses scope -1. The built-in Default dashboard cannot be deleted.

Writable fields: `name`.

dashboarditems

`dashboard_id` determines ownership; there is no writable tenant ID. `code` is an existing dashboard widget code, such as ONLINEEXTENSIONS. `order` controls display order. `global=yes` selects the global dashboard scope.

Writable fields: `td_id`, `code`, `order`, `dashboard_id`.

linemonitors

`start_url`, `answer_url` and `end_url` must be HTTP(S) URLs or empty. `hosted` is an existing server name or empty.

Writable fields: `name`, `start_url`, `answer_url`, `end_url`, `hosted`.

xmltemplates

`xmltemplate` contains the template text. Read-only keys receive metadata without template content.

Writable fields: `name, description, xmltemplate`.

causecodes

destinations is an ordered array of existing destination codes, such as EXT-100. Supply [] to clear the routing list; omit it to preserve routing. disposition: empty, CHANUNAVAIL, CONGESTION, BUSY, ANSWER, NOANSWER or CANCEL. global=yes selects scope 0 with a global key.

Writable fields: `disposition, code, comment, destinations`.

cnamcacheentries

number is unique within the selected tenant. New manual entries default to manual=on and the current datetime.

Writable fields: `number, name, datetime, manual, note`.

extensiondialings

from and to contain the extension dialing patterns; advanced follows the existing editor value (on or empty).

Writable fields: `from, to, advanced`.

translationrooms

Also available on installations where Translation Rooms is hidden from the menu. Deleting a room removes its language rows.

Writable fields: `name, basenumber, minsilence, silencethreshold`.

translationlanguages

room_id must belong to the selected tenant. number is the room offset; language is the TTS language and speaking is the voice name.

Writable fields: `tr_id, number, language, speaking, welcome_text, room_id`.

customdestinationinfos

customdestination_id must belong to the selected tenant. param1, param2 and param3 hold the custom type's extended data. Their meaning depends on the destination type.

Writable fields: `cu_id, param1, param2, param3, customdestination_id`.

customdestinationfiles

customdestination_id must belong to the selected tenant. data_base64 stores arbitrary binary data, at most 16 MiB decoded. name is a stored label, not a server filesystem path. Full-key detail GET downloads data_base64; lists return data_bytes without content.

Writable fields: `cu_id, name, binaryfile, data_base64, customdestination_id`.

provisioningvalues

type is PHONE, LINE or BUTTON. owner_id is a phone ID for PHONE/LINE and a button layout ID for BUTTON. label_id must have the same type. PHONE uses order=0; LINE uses a line in the phone model range. BUTTON uses a position within the layout range; order=0 is reserved for phonebook mappings on a phonebook-enabled layout. Value strings may contain the template placeholders used by the web editor. Read-only keys do not receive values.

Writable fields: `ph_id, type, order, al_id, value, owner_id, label_id`.

phonebuttonlayouts

phone_id and layout_id must belong to the same selected tenant. order controls assignment order.

Writable fields: `ph_id, bl_id, order, phone_id, layout_id`.

phonebookassignments

phone_id and phonebook_id must belong to the same selected tenant. order controls assignment order.

Writable fields: `ph_id, pb_id, order, phone_id, phonebook_id`.

featuremappings

code must be a standard function from sf_standardfunctions, for example PICKUP. feature_id is a tenant/global feature ID, 0 for the system/global default, or -1 to disable. Deleting a mapping

removes the override; the web page may recreate a default mapping. global=yes requires a global key and uses scope 0.

Writable fields: `code`, `fe_id`, `feature_id`.

Examples

```
curl -X POST -H "X-API-Key: TENANT_API_KEY" -H "Content-Type: application/json" \
-d '{"name":"Reception","number":10}' \
'https://pbx.example.com/pbx/openapi.php/pickupgroups?tenant=TENANT_CODE'

curl -X POST -H "X-API-Key: TENANT_API_KEY" -H "Content-Type: application/json" \
-d '{"phone_id":8,"phonebook_id":12,"order":1}' \
'https://pbx.example.com/pbx/openapi.php/phonebookassignments?tenant=TENANT_CODE'

curl -X POST -H "X-API-Key: TENANT_API_KEY" -H "Content-Type: application/json" \
-d '{"owner_id":8,"type":"BUTTON","order":2,"label_id":5,"value":"100"}' \
'https://pbx.example.com/pbx/openapi.php/provisioningvalues?tenant=TENANT_CODE'

curl -X PATCH -H "X-API-Key: TENANT_API_KEY" -H "Content-Type: application/json" \
-d '{"name":"Updated reception"}' \
'https://pbx.example.com/pbx/openapi.php/pickupgroups/OBJECT_ID?tenant=TENANT_CODE'

curl -X DELETE -H "X-API-Key: TENANT_API_KEY" \
'https://pbx.example.com/pbx/openapi.php/phonebookassignments/OBJECT_ID?tenant=TENANT_CODE'
```

The example parent, label and object IDs must be replaced with IDs from your installation. For BUTTON values the owner is the button layout, not the phone. To create a phonebook mapping, use a phonebook-enabled layout and order=0.

Generated internal phone book

`GET /internalphonebook?tenant=TENANT_CODE` returns entries with `type`, `id`, `number` and `description`, generated from configuration objects, parking slots and reserved numbers. Use `limit`, `offset` and optional `search` for an exact number or conduit prefix. Editing or deleting an entry means editing or deleting its source object. There is no separate writable internal-directory table. To avoid unbounded expansion, a legacy parking range larger than 100000 spaces produces an explicit error.

Extension technology conversion

`POST /extensions/{id}/technology` changes an extension between SIP, PJSIP, CUSTOM and VIRTUAL in one transaction. The extension ID, number, peer name, routing and queue memberships are preserved. SIP/PJSIP conversions retain the password and codec list unless replacements are supplied. Other technology fields are initialized from the target technology’s new-item defaults. Review target-specific settings after conversion; settings without an equivalent in the new technology are not migrated.

```
curl -X POST -H "X-API-Key: TENANT_API_KEY" -H "Content-Type: application/json" \
  -d '{"technology":"PJSIP"}' \
  'https://pbx.example.com/pbx/openapi.php/extensions/8/technology?tenant=TENANT_CODE'
```

Optional fields: `secret` (or `password`), `allow` (codec string), `custom`, `virtual`, `providers` (CUSTOM provider IDs), and `items` (VIRTUAL member extension IDs). The custom object accepts `ce_dialoption`, `ce_auth_type`, `ce_auth_parameter`, `ce_usecid`, `ce_custom_param1`. The virtual object accepts supported columns among `ve_callerid`, `ve_mailbox`, `ve_timeout`, `ve_strategy`; unavailable installation columns are rejected. Send other extension edits in a separate request.

A technology-only PATCH to `/extensions/{id}` with `{"tech":"PJSIP"}` also converts. Query routing: `POST ?object=extension&action=technology&id=8&tenant=TENANT_CODE`. Shared/custom PJSIP authentication or AOR links block conversion until normalized. An extension already used inside another virtual extension cannot become virtual. The response identifies the previous and new technology; a live PBX refresh failure returns a warning after the database change has committed. A database failure rolls the conversion back.

Voicemail messages and greetings

Resource	Supported methods
<code>/voicemails/{id}/messages</code>	GET list, POST create
<code>/voicemails/{id}/messages/{messageld}</code>	GET, PUT, PATCH, DELETE
<code>/voicemails/{id}/greetings</code>	GET list, POST create
<code>/voicemails/{id}/greetings/{name}</code>	GET, PUT, PATCH, DELETE

The voicemail ID is the mailbox configuration ID, and `messageld` is the recording’s database ID, not its position number. Greeting names: `busy`, `unavail`, `greet`, `temp`. Every operation checks both tenant and mailbox ownership. Use `limit` and `offset` for lists (100 by default, maximum 1000).

POST requires `data_base64`; greeting POST also requires `name`. Audio must be PCM WAV, mono, 8000 Hz, 16 bit, at most 16 MiB decoded, and the installation’s VOICEMAILFORMAT must be wav.

Full-key detail GET returns data_base64; list responses and read-only keys omit audio. An existing greeting is replaced with PUT/PATCH, not another POST.

Message fields: `folder` (INBOX by default; also Old, Work, Family, Friends, Cust1–Cust5), `callerid`, `flag` (empty, Urgent or PRIORITY), `category`, `origtime` (Unix timestamp), and `data_base64`. PATCH folder moves the message within that mailbox. Storage paths, mailbox identity, message positions and analysis fields are generated/read-only. Replacing audio clears stale transcripts and summaries.

```
curl -H "X-API-Key: TENANT_API_KEY" \
  'https://pbx.example.com/pbx/openapi.php/voicemails/8/messages?tenant=TENANT_CODE'

curl -X PATCH -H "X-API-Key: TENANT_API_KEY" -H "Content-Type: application/json" \
  -d '{"folder":"Old"}' \
  'https://pbx.example.com/pbx/openapi.php/voicemails/8/messages/42?tenant=TENANT_CODE'

curl -X PUT -H "X-API-Key: TENANT_API_KEY" -H "Content-Type: application/json" \
  -d '{"data_base64":"BASE64_ENCODED_WAV"}' \
  'https://pbx.example.com/pbx/openapi.php/voicemails/8/greetings/busy?tenant=TENANT_CODE'
```

Query routing is supported using `object=voicemail&action=messages&id=8&item=42` or `action=greetings&id=8&item=busy`. Omit item for collection requests. DELETE requires one item and does not clear the entire mailbox.

Revision #3

Created 2026-09-26 17:46:53 UTC by Admin

Updated 2026-10-18 11:07:38 UTC by Admin