OpenAPI JSON endpoint reference and examples

Overview and examples

`openapi.php` is the MiRTA PBX JSON API endpoint based on the OpenAPI Specification. It supports standard HTTP methods, JSON request bodies, API-key authentication, tenant scoping, and CRUD operations for many PBX configuration objects.

The OpenAPI Specification, often abbreviated as OAS, is a vendor-neutral and programming-language independent description format for HTTP APIs. An OpenAPI document describes the public contract of an API: available paths, supported operations, parameters, request and response bodies, authentication methods, examples, and general service metadata. The document is normally published as JSON or YAML, which makes it readable by people and directly usable by tools.

In MiRTA PBX, the OpenAPI document acts as a machine-readable map of the available API. It lets administrators and integrators inspect the supported endpoints, build repeatable integrations, and keep API usage aligned with the behavior exposed by the PBX server.

Why Use OpenAPI

Advantage	Description
Shared API contract	The specification provides one explicit description of what the API is expected to expose, independently from the language used to implement the server or the client.
Interactive documentation	OpenAPI-compatible tools can render the specification as browsable documentation where developers can inspect endpoints and, when enabled, try requests from the browser.
Client and server generation	Tooling can generate client libraries, SDKs, server stubs, and boilerplate request handling from the same API description.

Advantage	Description
Testing and validation	The API document can be used for contract tests, request and response validation, mock servers, and security checks against the declared API surface.
Better integration workflow	Integrators can import the specification into API clients, testing tools, gateways, and monitoring systems instead of manually recreating every endpoint and payload format.

OpenAPI and Swagger

OpenAPI is the specification. Swagger refers to a family of tools built around that specification, such as Swagger UI for interactive documentation and Swagger Codegen for generating client libraries or server stubs. MiRTA PBX exposes the OpenAPI document; external OpenAPI or Swagger-compatible tools can consume it.

Specification

The OpenAPI document is available from these endpoints:

```
GET /mirtapbx/openapi.php
GET /mirtapbx/openapi.php?spec=1
GET /mirtapbx/openapi.php/openapi.json
GET /mirtapbx/openapi.php/swagger.json
```

The response is an OpenAPI 3.0.3 JSON document.

Authentication

Provide the API key as a query parameter, an `X-API-Key` header, or a bearer token. Tenant API keys require a tenant code. Global keys can list across tenants when no tenant is supplied.

```
curl -H "X-API-Key: TENANT_API_KEY" \
  "https://pbx.example.com/mirtapbx/openapi.php/extensions?tenant=TENANTCODE"

curl -H "Authorization: Bearer TENANT_API_KEY" \
  "https://pbx.example.com/mirtapbx/openapi.php/extension?number=100&tenant=TENANTCODE"
```

API Key Scope

Tenant API Key

When using a tenant-level API key, include the tenant parameter. The tenant can be supplied by tenant code or tenant name.

```
openapi.php/extensions?tenant=TENANTCODE&key=TENANT_API_KEY
```

Global API Key

When using a global API key, the tenant parameter is optional. If tenant is provided, the request is scoped to that tenant. If tenant is omitted, list and get requests can return records across tenants where the object supports global access.

```
openapi.php/extensions?key=GLOBAL_API_KEY
```

Some objects can be edited at global level by adding `global=yes`. This is supported for global settings, global custom destinations, global media files, global music on hold, global caller ID blacklist entries, global cron jobs, global feature codes, and global short numbers.

The tenant, user, user profile, and routing profile objects require an Admin API key. Tenant API keys are rejected for these objects. For users, adding `tenant=TENANTCODE` with an Admin API key filters list and get requests to users assigned to that tenant.

```
openapi.php/featurecodes?global=yes&key=GLOBAL_API_KEY
```

Endpoint Patterns

Action	Pattern
List	`GET /openapi.php/extensions?tenant=TENANTCODE`
Get by ID	`GET /openapi.php/extension?id=ID&tenant=TENANTCODE`
Get extension by number	`GET /openapi.php/extensions/number/100?tenant=TENANTCODE`
Create	`POST /openapi.php/extensions?tenant=TENANTCODE`
Modify	`PATCH /openapi.php/extensions/ID?tenant=TENANTCODE`

Action	Pattern
Delete	`DELETE /openapi.php/extensions/ID?tenant=TENANTCODE`

Quick Examples

```
# Create a PJSIP extension
curl -X POST -H "X-API-Key: TENANT_API_KEY" -H "Content-Type: application/json" \
  -d '{"number":"210","name":"API Demo","tech":"PJSIP","password":"change-this-secret"}' \
  "https://pbx.example.com/mirtapbx/openapi.php/extensions?tenant=TENANTCODE"

# Update a queue name
curl -X PATCH -H "X-API-Key: TENANT_API_KEY" -H "Content-Type: application/json" \
  -d '{"name":"Accounting Support"}' \
  "https://pbx.example.com/mirtapbx/openapi.php/queues/12?tenant=TENANTCODE"

# Create a voice routing profile with a global key
curl -X POST -H "X-API-Key: GLOBAL_API_KEY" -H "Content-Type: application/json" \
  -d '{"name":"Docs Demo Voice Routing","description":"Documentation example","type":"VOICE"}'
\
  "https://pbx.example.com/mirtapbx/openapi.php/routingprofiles"

# Move a tenant to Post Paid and assign a routing profile
curl -X PATCH -H "X-API-Key: GLOBAL_API_KEY" -H "Content-Type: application/json" \
  -d '{"te_payment_type":"Post Paid","routing_profile_id":3}' \
  "https://pbx.example.com/mirtapbx/openapi.php/tenant?id=TENANT_ID"
```

Supported Objects

The endpoint maps plural and singular paths for extensions, voicemails, tenants, users, user profiles, routing profiles, conditions, IVRs, custom destinations, hunt lists, DIDs, queues, settings, media files, music on hold, paging groups, conference rooms, flows, tenant variables, DISAs, caller ID blacklists, campaigns, campaign numbers, cron jobs, feature codes, short numbers, and provisioning phones.

Errors

Errors are returned as JSON with an error code and message. Common errors include missing API key, invalid API key, tenant not found, invalid JSON, missing required field, unsupported method, and endpoint not found.

External OpenAPI Resources

- OpenAPI Initiative: What is OpenAPI?
- OpenAPI Initiative: Getting Started introduction
- Swagger Docs: What is OpenAPI?

List Extensions

Endpoint

```
GET openapi.php/extensions
```

Alternative compatibility format:

```
openapi.php?object=extension&action=list
```

Parameters

Name	Required	Description
key	Yes, unless using header authentication	Full or read-only API key
tenant	Required for tenant keys	Tenant code or tenant name

Example Request

```
curl "https://pbx.example.com/openapi.php/extensions?tenant=TENANTCODE&key=APIKEY"
```

Using header authentication:

```
curl \
  -H "X-API-Key: APIKEY" \
  "https://pbx.example.com/openapi.php/extensions?tenant=TENANTCODE"
```

Using bearer authentication:

```
curl \
  -H "Authorization: Bearer APIKEY" \
  "https://pbx.example.com/openapi.php/extensions?tenant=TENANTCODE"
```

Example Response

```
[
  {
    "id": 101,
    "number": "100",
    "name": "Reception",
    "tech": "PJSIP"
  },
  {
    "id": 102,
    "number": "101",
    "name": "Office",
    "tech": "SIP"
  }
]
```

Response Fields

Field	Type	Description
id	integer	Extension internal ID
number	string	Extension number
name	string	Extension display name
tech	string	Extension technology, for example SIP, PJSIP, VIRTUAL, or CUSTOM

Get Extension Information

This endpoint returns detailed information for one extension.

The extension can be selected by internal extension ID or by extension number.

When using the global full API key, the response also includes the extension password when a password exists for the extension technology.

Tenant API keys and read-only API keys do not return the extension password.

Endpoints

Get an extension by ID:

```
GET openapi.php/extension?id=EXTENSION_ID
```

Alternative path format:

```
GET openapi.php/extensions/EXTENSION_ID
```

Get an extension by number:

```
GET openapi.php/extension?number=EXTENSION_NUMBER
```

Alternative path format:

```
GET openapi.php/extensions/number/EXTENSION_NUMBER
```

Alternative compatibility format:

```
openapi.php?object=extension&action=info&id=EXTENSION_ID
openapi.php?object=extension&action=info&number=EXTENSION_NUMBER
```

Parameters

Name	Required	Description
key	Yes, unless using header authentication	Full or read-only API key

Name	Required	Description
tenant	Required for tenant keys	Tenant code or tenant name
id	Required if number is not provided	Extension internal ID
number	Required if id is not provided	Extension number

Use either id or number, not both.

When using a global API key without the tenant parameter, selecting by extension number can match more than one tenant. In that case the request returns an error and the request should be repeated using the tenant parameter or the extension ID.

Example Requests

Get by ID:

```
curl "https://pbx.example.com/openapi.php/extension?id=101&tenant=TENANTCODE&key=APIKEY"
```

Get by number:

```
curl "https://pbx.example.com/openapi.php/extension?number=100&tenant=TENANTCODE&key=APIKEY"
```

Using header authentication:

```
curl \
  -H "X-API-Key: APIKEY" \
  "https://pbx.example.com/openapi.php/extension?number=100&tenant=TENANTCODE"
```

Example Response

The response includes all fields from the extension table, the normalized fields id, number, name, and tech, the current state when available, the technology username, technology-specific details, and related records.

Related records are returned in the related object. Depending on the extension configuration, this can include state information, registrations, PJSIP contacts, call groups, pickup groups, destinations, queue membership, allowed queue membership, user profile, routing profiles, client rate, caller ID regex, music on hold, parking lot, conditions, media files, and associated phones.

The me_data binary field from media files is not returned in this response.

```
{
  "ex_id": "101",
  "ex_te_id": "1",
  "ex_name": "Reception",
  "ex_number": "100",
  "ex_tech": "PJSIP",
  "ex_tech_id": "55",
  "tenant_code": "TENANTCODE",
  "tenant_name": "Tenant Name",
  "id": 101,
  "number": "100",
  "name": "Reception",
  "tech": "PJSIP",
  "username": "100",
  "state": "NOT_INUSE",
  "st_state": "NOT_INUSE",
  "tech_details": {
    "endpoint": {
      "id": "100",
      "tech_id": "55",
      "te_id": "1"
    },
    "aor": {
      "id": "100",
      "max_contacts": "99"
    },
    "auth": {
      "id": "100",
      "username": "100"
    }
  },
  "related": {
    "state": {
      "st_extension": "100",
      "st_state": "NOT_INUSE"
    },
    "callgroups": [],
    "pickupgroups": [],
    "destinations": [],
    "referenced_by_destinations": [],
```

```
    "queue_members": [],
    "allowed_queue_members": [],
    "userprofile": {
      "up_id": "3",
      "up_name": "Basic user panel"
    },
    "routing_profile": {
      "rp_id": "1",
      "rp_name": "Default"
    },
    "sms_routing_profile": false,
    "client_rate": false,
    "callerid_regex": false,
    "callerid_regex_rules": [],
    "music_on_hold": false,
    "parkinglot": false,
    "conditions": [],
    "mediafiles": [],
    "phones": []
  }
}
```

When the global full API key is used, the response can also include:

```
{
  "password": "extension_password"
}
```

Create Extension

This endpoint creates one extension. A full API key is required. Read-only API keys are rejected.

The request is checked against the installed license before the extension is created.

Endpoint

```
POST openapi.php/extensions
```

Alternative compatibility format:

```
POST openapi.php?object=extension&action=add
POST openapi.php?object=extension&action=create
```

Parameters

Name	Required	Description
key	Yes, unless using header authentication	Full API key
tenant	Yes	Tenant code or tenant name

Request Body

The body is JSON. The API accepts the extension table fields used by the web page, such as ex_number, ex_name, ex_tech, and the technology table fields for sipfriends, ps_endpoints, ps_aors, ps_auths, ce_customextensions, and ve_virtualextensions.

Short aliases are also accepted:

Alias	Field
number	ex_number
name	ex_name
tech	ex_tech
password	SIP secret or PJSIP auth password
username	Technology username
sipusername	Technology username, matching the web page field
mailbox	ex_mailbox and PJSIP mailboxes

Supported technologies are SIP, PJSIP, CUSTOM, and VIRTUAL. If tech is omitted, PJSIP is used.

Example Request

```
curl \
  -X POST \
  -H "Content-Type: application/json" \
  -H "X-API-Key: APIKEY" \
```

```
  -d '{"number":"100","name":"Reception","tech":"PJSIP","password":"secret-password"}' \
  "https://pbx.example.com/openapi.php/extensions?tenant=TENANTCODE"
```

Nested technology fields can also be supplied:

```
{
  "ex_number": "100",
  "ex_name": "Reception",
  "ex_tech": "PJSIP",
  "ps_aors": {
    "max_contacts": 99
  },
  "ps_endpoints": {
    "allow": "alaw:20;ulaw:20"
  },
  "ps_auths": {
    "password": "secret-password"
  }
}
```

Example Response

The response is the same detailed extension object returned by the get endpoint, with HTTP status 201.

Modify Extension

This endpoint modifies an existing extension. A full API key is required. Read-only API keys are rejected.

The extension can be selected by internal ID or extension number. Use either id or number, not both.

Endpoints

```
PUT openapi.php/extension?id=EXTENSION_ID
PATCH openapi.php/extension?id=EXTENSION_ID
PATCH openapi.php/extensions/EXTENSION_ID
```

Alternative compatibility format:

```
PUT openapi.php?object=extension&action=update&objectid=EXTENSION_ID
PATCH openapi.php?object=extension&action=modify&objectid=EXTENSION_ID
```

Request Body

The body is JSON and can contain the same table fields accepted by the create endpoint. Only fields included in the body are updated.

Changing the extension technology is not supported. To change technology, delete and recreate the extension.

Example Request

```
curl \
  -X PATCH \
  -H "Content-Type: application/json" \
  -H "X-API-Key: APIKEY" \
  -d '{"ex_name":"Reception Desk","ex_callgroup":"1,2","ex_pickupgroup":"1,2"}' \
  "https://pbx.example.com/openapi.php/extension?id=101&tenant=TENANTCODE"
```

Delete Extension

This endpoint deletes an extension and its related technology rows. A full API key is required. Read-only API keys are rejected.

Endpoints

```
DELETE openapi.php/extension?id=EXTENSION_ID
DELETE openapi.php/extensions/EXTENSION_ID
```

Alternative compatibility format:

```
DELETE openapi.php?object=extension&action=delete&objectid=EXTENSION_ID
```

Example Request

```
curl \
  -X DELETE \
  -H "X-API-Key: APIKEY" \
  "https://pbx.example.com/openapi.php/extension?id=101&tenant=TENANTCODE"
```

Example Response

```
{
  "deleted": true,
  "id": 101,
  "number": "100"
}
```

Extension Busy Destination

The busy destination for an extension can be set with the onbusy alias.

Destination values use the same TYPE-ID format returned in related.destinations. For example, use VOICEMAIL-12 for voicemail ID 12 or EXT-45 for extension internal ID 45. Extension destinations use the extension internal ID, not the extension number.

To send busy calls to an existing voicemail:

```
curl \
  -X PATCH \
  -H "Content-Type: application/json" \
  -H "X-API-Key: APIKEY" \
  -d '{"onbusy":["VOICEMAIL-12"]}' \
  "https://pbx.example.com/openapi.php/extension?id=101&tenant=TENANTCODE"
```

The same destination can also be supplied with the explicit destination type:

```
{
  "destinations": {
    "EXT-BUSY": ["VOICEMAIL-12"]
  }
```

```
}
```

To use the extension same-number voicemail and create it automatically when missing:

```
{
  "onbusy": ["SAMENUMBERVM"]
}
```

Sending an empty array clears the busy destination:

```
{
  "onbusy": []
}
```

Additional Configuration Objects

The following objects use the same authentication and HTTP behavior as extensions.

Object	List Endpoint	Single Endpoint	ID Field	Main Table
voicemail	GET openapi.php/voicemails	openapi.php/voicemail?id=ID	uniqueid	voicemail
tenant	GET openapi.php/tenants	openapi.php/tenant?id=ID	te_id	te_tenants
user	GET openapi.php/users	openapi.php/user?id=ID	us_id	us_users
userprofile	GET openapi.php/userprofiles	openapi.php/userprofile?id=ID	up_id	up_userprofiles
routingprofile	GET openapi.php/routingprofiles	openapi.php/routingprofile?id=ID	rp_id	rp_routingprofiles
condition	GET openapi.php/conditions	openapi.php/condition?id=ID	co_id	co_conditions
ivr	GET openapi.php/ivrs	openapi.php/ivr?id=ID	iv_id	iv_ivrs
customdestination	GET openapi.php/customdestinations	openapi.php/customdestination?id=ID	cu_id	cu_customs

Object	List Endpoint	Single Endpoint	ID Field	Main Table
huntlist	GET openapi.php/huntlists	openapi.php/huntlist?id=ID	hu_id	hu_huntlists
did	GET openapi.php/dids	openapi.php/did?id=ID	di_id	di_dids
queue	GET openapi.php/queues	openapi.php/queue?id=ID	qu_id	qu_queues
setting	GET openapi.php/settings	openapi.php/setting?id=ID	se_id	se_settings
mediafile	GET openapi.php/mediafiles	openapi.php/mediafile?id=ID	me_id	me_mediafiles
musiconhold	GET openapi.php/musiconholds	openapi.php/musiconhold?id=ID	mu_id	mu_musiconholds
paginggroup	GET openapi.php/paginggroups	openapi.php/paginggroup?id=ID	pa_id	pa_paginggroups
conferenceroom	GET openapi.php/conferencerooms	openapi.php/conferenceroom?id=ID	cr_id	cr_conferencerooms
flow	GET openapi.php/flows	openapi.php/flow?id=ID	fl_id	fl_flows
tenantvariable	GET openapi.php/tenantvariables	openapi.php/tenantvariable?id=ID	tv_id	tv_tenantvariables
disa	GET openapi.php/disas	openapi.php/disa?id=ID	ds_id	ds_disas
calleridblacklist	GET openapi.php/calleridblacklists	openapi.php/calleridblacklist?id=ID	bl_id	bl_blacklists
campaign	GET openapi.php/campaigns	openapi.php/campaign?id=ID	ca_id	ca_campaigns
campaignnumber	GET openapi.php/campaignnumbers	openapi.php/campaignnumber?id=ID	cn_id	cn_campaignnumbers
cronjob	GET openapi.php/cronjobs	openapi.php/cronjob?id=ID	cr_id	cr_cronjobs
featurecode	GET openapi.php/featurecodes	openapi.php/featurecode?id=ID	fe_id	fe_features

Object	List Endpoint	Single Endpoint	ID Field	Main Table
shortnumber	GET openapi.php/shortnumbers	openapi.php/shortnumber?id=ID	sn_id	sn_shortnumbers
provisioningphone	GET openapi.php/provisioningphones	openapi.php/provisioningphone?id=ID	ph_id	ph_phones

Generic Endpoints

Use the plural path for list and create operations:

```
GET openapi.php/voicemails
POST openapi.php/voicemails
GET openapi.php/tenants
POST openapi.php/tenants
GET openapi.php/users
POST openapi.php/users
GET openapi.php/userprofiles
POST openapi.php/userprofiles
GET openapi.php/routingprofiles
POST openapi.php/routingprofiles
GET openapi.php/conditions
POST openapi.php/conditions
GET openapi.php/ivrs
POST openapi.php/ivrs
GET openapi.php/customdestinations
POST openapi.php/customdestinations
GET openapi.php/huntlists
POST openapi.php/huntlists
GET openapi.php/dids
POST openapi.php/dids
GET openapi.php/queues
POST openapi.php/queues
GET openapi.php/settings
POST openapi.php/settings
GET openapi.php/mediafiles
POST openapi.php/mediafiles
GET openapi.php/musiconholds
POST openapi.php/musiconholds
```

```
GET openapi.php/paginggroups
POST openapi.php/paginggroups
GET openapi.php/conferencerooms
POST openapi.php/conferencerooms
GET openapi.php/flows
POST openapi.php/flows
GET openapi.php/tenantvariables
POST openapi.php/tenantvariables
GET openapi.php/disas
POST openapi.php/disas
GET openapi.php/calleridblacklists
POST openapi.php/calleridblacklists
GET openapi.php/campaigns
POST openapi.php/campaigns
GET openapi.php/campaignnumbers
POST openapi.php/campaignnumbers
GET openapi.php/cronjobs
POST openapi.php/cronjobs
GET openapi.php/featurecodes
POST openapi.php/featurecodes
GET openapi.php/shortnumbers
POST openapi.php/shortnumbers
GET openapi.php/provisioningphones
POST openapi.php/provisioningphones
```

Use either the singular endpoint with an id parameter or the plural path with the ID:

```
GET openapi.php/voicemail?id=ID
PATCH openapi.php/voicemail?id=ID
DELETE openapi.php/voicemail?id=ID

GET openapi.php/voicemails/ID
PATCH openapi.php/voicemails/ID
DELETE openapi.php/voicemails/ID
```

The same pattern applies to all generic objects listed in the table above.

Alternative compatibility format:

```
openapi.php?object=voicemail&action=list
openapi.php?object=voicemail&action=info&id=ID
openapi.php?object=voicemail&action=create
openapi.php?object=voicemail&action=modify&id=ID
openapi.php?object=voicemail&action=delete&id=ID
```

Generic Request Body

The body is JSON. Fields can be supplied using the database column names used by the web pages.

For the user object, password, _password, and us_password are treated as clear text input and are stored using the same SHA-256 hash used by the web page. To provide a precomputed hash, use password_hash or us_password_hash.

Short aliases are supported where they are unambiguous:

Object	Alias	Field
voicemail	number	mailbox
voicemail	name	fullname
tenant	name	te_name
tenant	code	te_code
user	username	us_username
user	password	us_password
user	profile_id	us_up_id
userprofile	name	up_name
userprofile	reserved	up_reserved
routingprofile	name	rp_name
routingprofile	type	rp_type
condition	name	co_name
condition	type	co_type
ivr	name	iv_name
customdestination	name	cu_name
huntlist	name	hu_name
huntlist	number	hu_number
did	number	di_number

Object	Alias	Field
queue	name	qu_name
setting	code	se_code
setting	value	se_value
mediafile	name	me_name
mediafile	data_base64	me_data
musiconhold	name	mu_name
musiconhold	custom	mu_custom
paginggroup	number	pa_number
paginggroup	name	pa_name
conferenceroom	name	cr_name
conferenceroom	number	cr_number
flow	name	fl_name
flow	variable_name	fl_variable_name
tenantvariable	variable_id	tv_al_id
tenantvariable	value	tv_value
disa	name	ds_name
calleridblacklist	callerid	bl_callerid
campaign	name	ca_name
campaign	state	ca_state
campaignnumber	campaign_id	cn_ca_id
campaignnumber	number	cn_number
cronjob	name	cr_name
cronjob	run	cr_run
featurecode	code	fe_code
shortnumber	number	sn_number
provisioningphone	name	ph_name
provisioningphone	mac	ph_mac

Destinations

Objects that use destinations accept a destinations object. Each destination item can be either a string in TYPE-ID format or an object with type and id.

```
{
  "di_number": "390212345678",
  "di_comment": "Main number",
  "destinations": {
    "DID": ["EXT-101"],
    "DID-UNCONDITIONAL": [
      { "type": "VOICEMAIL", "id": 10 }
    ]
  }
}
```

Supported destination keys:

Object	Destination Keys
voicemail	VOICEMAIL-OPERATOR, VOICEMAIL-FOLLOW, VOICEMAIL-BROADCAST
condition	CONDITION, NOTCONDITION, and CONDITION1 through CONDITION20
ivr	IVR_1 through IVR_9, IVR_0, IVR_STAR, IVR_SHARP, IVR_WRONG, IVR_TIMEOUT, IVR_HANGUP, IVR_FEATURE, IVR_EXTENSION, IVR_MEDIAFILE, IVR_OPTIONSMEDIAFILE, and custom keys beginning with CUSTOMIVR_
customdestination	PRIVACY-DONTCALL, PRIVACY-TORTURE, CTONANSWER, CALLBACK-CONNECTED, CTONCALLERHANGUP, SPLITCHANNELACTION-CALLER, SPLITCHANNELACTION-CALLED, and RANDOMDESTINATION keys
musiconhold	MUSICONHOLD. Plain numeric values are treated as media file IDs.
paginggroup	PAGING. Plain numeric values are treated as extension IDs.
huntlist	HUNTLIST, HUNTLIST-TIMEOUT
did	DID, DID-UNCONDITIONAL, DID-SMS, DID-FAXSUCCESS
queue	QUEUE-FULL, QUEUE-TIMEOUT, QUEUE-EXITKEY, QUEUE-ONCALLBACK, QUEUE-NOBODYHOME, QUEUE-NOFREEMEMBER, QUEUE-PERIODICANNOUNCE, QUEUE-BEFORERINGING, QUEUE-ONAUTOPAUSE, QUEUE-ONABANDONEDCALL
flow	FLOW
campaign	CAMPAIGN-ONCONNECT and CAMPAIGN-DONOTCALL. Plain numeric values for CAMPAIGN-DONOTCALL are treated as do-not-call IDs.
cronjob	CRONJOB

Object	Destination Keys
featurecode	FEATURE

Related Records

Get responses include a related object when the object has related configuration.

For users, related.profile contains the selected main user profile. related.tenants, related.routing_profiles, related.allowed_userprofiles, related.client_rates, and the restriction lists show the assignments used by the user page. Create and modify requests can replace these assignments by sending tenants, routingprofiles, allowed_userprofiles, clientrates, queue_restrictions, extension_restrictions, or provider_restrictions as arrays of IDs.

For user profiles, related.privileges contains the assigned privilege rows. Create and modify requests can replace the assigned privileges by sending privileges as an array of privilege IDs or objects containing ug_id and optional param1 through param5.

For conditions, related.extended_infos contains rows from ce_conditions_extended. Create and modify requests can replace these rows by sending extended_infos.

For queues, related.realtime contains the realtime queue row. Create and modify requests can send realtime queue fields either at the top level or nested under queue or realtime. Queue members can be replaced by sending members, queue_member, or queue_members. Allowed queue members can be replaced by sending allowed_members, aq_allowed_queue_member, or allowed_queue_members.

For hunt lists, create and modify requests update the matching follow-me records used by the dialplan.

For custom destinations, related.extended_infos contains rows from ce_customs_extended and related.binary_files lists stored binary records without returning the binary payload.

For music on hold, related.realtime contains the linked musiconhold row and related.entries contains the generated playlist entries. Create and modify requests can send mediafiles as media file IDs or MEDIAFILE-ID strings. Sending entries replaces the raw musiconhold_entry rows directly.

For conference rooms, related.meetme contains the linked realtime conference row. Creating a conference room also creates the linked meetme row, and deleting it removes that row.

For flows, sending status, state, or st_state updates the matching st_states row when the flow has a number.

For campaigns, related.numbers_count, related.numbers_by_disposition, and related.binary_files summarize linked campaign number and fax file records. Campaign fax files can be replaced by sending binary_files with name and data_base64. Campaign numbers are managed through the

separate campaignnumber object. The list endpoint accepts campaign_id or caid to limit numbers to one campaign.

For cron jobs, setting run to yes schedules a run in the same way as setting cr_run.

For provisioning phones, related.button_layouts and related.phonebooks contain the linked phone layout and phonebook rows.

Examples

Create a voicemail:

```
curl \
  -X POST \
  -H "Content-Type: application/json" \
  -H "X-API-Key: APIKEY" \
  -d '{"mailbox":"200","fullname":"Sales","password":"1234","email":"sales@example.com"}' \
  "https://pbx.example.com/openapi.php/voicemails?tenant=TENANTCODE"
```

Modify a DID destination:

```
curl \
  -X PATCH \
  -H "Content-Type: application/json" \
  -H "X-API-Key: APIKEY" \
  -d '{"destinations":{"DID":["EXT-101"]}}' \
  "https://pbx.example.com/openapi.php/did?id=25&tenant=TENANTCODE"
```

Create a queue with realtime fields:

```
curl \
  -X POST \
  -H "Content-Type: application/json" \
  -H "X-API-Key: APIKEY" \
  -d
'{"qu_name":"Support","qu_number":"700","queue":{"strategy":"ringall","timeout":20},"members":
[{"membername":"100","interface":"Local/AG-000-NF-
101@fromotherpbx/n","state_interface":"Custom:100","member_device":"100","penalty":0,"paused":
0}]}' \
  "https://pbx.example.com/openapi.php/queues?tenant=TENANTCODE"
```

Create a global feature code:

```
curl \
  -X POST \
  -H "Content-Type: application/json" \
  -H "X-API-Key: GLOBAL_APIKEY" \
  -d '{"code":"*56","comment":"Pickup"}' \
  "https://pbx.example.com/openapi.php/featurecodes?global=yes"
```

Create a conference room:

```
curl \
  -X POST \
  -H "Content-Type: application/json" \
  -H "X-API-Key: APIKEY" \
  -d '{"name":"Weekly Meeting","number":"900","pin":"1234","adminpin":"4321","maxusers":10}' \
  "https://pbx.example.com/openapi.php/conferencerooms?tenant=TENANTCODE"
```

Create a music on hold playlist:

```
curl \
  -X POST \
  -H "Content-Type: application/json" \
  -H "X-API-Key: APIKEY" \
  -d '{"name":"Support Hold","mediafiles":[12,15],"default":"on"}' \
  "https://pbx.example.com/openapi.php/musiconholds?tenant=TENANTCODE"
```

Create a campaign number:

```
curl \
  -X POST \
  -H "Content-Type: application/json" \
  -H "X-API-Key: APIKEY" \
  -d '{"campaign_id":5,"number":"390212345678","description":"Lead A"}' \
  "https://pbx.example.com/openapi.php/campaignnumbers?tenant=TENANTCODE"
```

Run a cron job:

```
curl \
  -X PATCH \
  -H "Content-Type: application/json" \
```

```
  -H "X-API-Key: APIKEY" \
  -d '{"run":"yes"}' \
  "https://pbx.example.com/openapi.php/cronjob?id=8&tenant=TENANTCODE"
```

Create a user and assign tenants, routing profiles, and allowed user profiles:

```
curl \
  -X POST \
  -H "Content-Type: application/json" \
  -H "X-API-Key: GLOBAL_APIKEY" \
  -d '{"username":"operator","password":"change-
me","profile_id":3,"tenants":[1],"routingprofiles":[2,4],"allowed_userprofiles":[5]}' \
  "https://pbx.example.com/openapi.php/users"
```

Create a user profile with privileges:

```
curl \
  -X POST \
  -H "Content-Type: application/json" \
  -H "X-API-Key: GLOBAL_APIKEY" \
  -d '{"name":"Helpdesk","description":"Helpdesk operators","privileges":[12,18,25]}' \
  "https://pbx.example.com/openapi.php/userprofiles"
```

Change Logging

Create, modify, and delete operations write a process log entry and a user activity log entry using the OpenAPI user label. Related destination, user assignment, user profile privilege, condition, queue member, and queue realtime changes are also logged where applicable.

Additional Error Responses

Missing Identifier

```
{
  "error": {
    "code": "missing_identifier",
    "message": "Provide an extension id or number."
  }
```

```
}
```

Invalid Identifier

```
{
  "error": {
    "code": "invalid_identifier",
    "message": "Use either id or number, not both."
  }
}
```

Extension Not Found

```
{
  "error": {
    "code": "extension_not_found",
    "message": "Extension not found."
  }
}
```

Multiple Extensions Found

```
{
  "error": {
    "code": "multiple_extensions_found",
    "message": "Multiple extensions match the requested number. Use the tenant parameter or the extension id."
  }
}
```

Error Responses

Errors are returned as JSON.

Missing API Key

```
{
  "error": {
    "code": "missing_api_key",
    "message": "Missing API key."
  }
}
```


Invalid API Key


```
{
  "error": {
    "code": "invalid_api_key",
    "message": "Invalid API key."
  }
}
```


Tenant Not Found


```
{
  "error": {
    "code": "tenant_not_found",
    "message": "Tenant not found."
  }
}
```


Endpoint Not Found


```
{
  "error": {
    "code": "not_found",
    "message": "Endpoint not found."
  }
}
```


Method Not Allowed

Supported methods are GET, POST, PUT, PATCH, and DELETE.

```
{
  "error": {
    "code": "method_not_allowed",
    "message": "Supported methods are GET, POST, PUT, PATCH and DELETE."
  }
}
```

Notes

- Read-only API keys are allowed for list and get endpoints.
- A full API key is required for create, modify, and delete endpoints.
- Sensitive fields such as passwords, API keys, tokens, and media binary data are hidden unless the global full API key is used.
- The endpoint does not create a web session.
- Responses are always JSON.
- CORS is enabled with Access-Control-Allow-Origin: *.
- OPTIONS requests return HTTP 204 for browser preflight support.

Revision #5
Created 2026-06-02 21:58:37 UTC by Admin
Updated 2026-06-03 10:22:15 UTC by Admin