

Botpress Bot CURD Operations

End Point

<https://backend.versalence.online/api/v2/agent-builder/ext>

IndiaMART uses the same x-api-key already provided by Versalence. Its existing read access is retained, with partner lifecycle permissions added server-side. No replacement key is required. Use the authorised workspace ID supplied with that key. Call these APIs from your backend; do not expose the key in a browser or mobile application.

IndiaMART submits seller details, prompt-variable values and knowledge sources. Versalence retains the private `indiamart.json` template, validates and stores the submission, and creates a dedicated Botpress bot after review approval. The template JSON and its internal configuration are not shared with the partner.

Release status 30 October 2026

Capability	Availability
CREATE submission and operation-status polling	Deployed. Review-gated creation verified live.
Internal review, approve and reject APIs	Deployed. Approval was verified through the server service; administrator HTTP/browser acceptance remains outstanding.
Review OFF for new submissions	Implemented and covered by automated tests; not newly live-tested in this release. The deployed setting remains ON.
UPDATE and DELETE	Planned; no callable endpoints yet. Proposed contracts are documented below for integration planning.
Cancel/retry APIs, per-workspace review switch, dashboard review screen and activation readiness	Pending implementation.

Successful creation currently produces a draft, not an active agent. Review OFF queues creation automatically; it does not establish activation or conversational readiness. A

webhook being configured does not mean the bot is active.

Authentication and identifiers

Header	Required	Description
<code>x-api-key</code>	Yes	A valid, active key bound to an active partner and the requested workspace.
<code>Idempotency-Key</code>	CREATE: yes	8-128 ASCII letters, digits or <code>. _ : -</code> . Generate one key for each intended submission; reuse it only to replay the identical request.
<code>Accept: application/json</code>	Recommended	Responses are JSON.
<code>Content-Type</code>	For request bodies	CREATE uses multipart/form-data. Let the HTTP client generate its boundary.

CREATE requires `agents.create`; status polling requires `operations.read`. A key's workspace, partner identity and scopes are all enforced. Sharing an owner label does not grant access. Another partner's operation is not accessible even when its ID is known.

Identifier	Meaning
<code>workspaceId</code>	The authorised Botpress workspace supplied by Versalence.
<code>clientReferenceId</code>	Your stable seller/account reference, unique within your partner and workspace.
<code>operationId</code>	The asynchronous request to track through review and execution.
<code>agentId</code>	The vCX agent identity. It can be allocated before a Botpress bot exists. It is not a Botpress bot ID.
<code>externalSourceId</code>	Your stable identifier for a document/table or website, used to identify results and warnings and, in the planned UPDATE API, source changes.

Create an agent: available

POST `/workspaces/{workspaceId}/agent-requests`

Validates the input, freezes the selected template and stores an encrypted revision and uploaded files on the server. With review ON, no Botpress creation starts until approval. Acceptance is asynchronous: an HTTP 202 is not a completed bot.

Path parameters

Name	Type	Required	Description
------	------	----------	-------------

workspaceId	string	Yes	The workspace authorised for the key.
-------------	--------	-----	---------------------------------------

Multipart form fields

Name	Type	Required	Description
payload	JSON encoded as a text field	Yes	The object described below. Do not send the entire request as application/json.
Each declared uploadField	File part	For each document	Exactly one file per declared field. No undeclared files or reused file parts.

Payload fields

Name	Type	Required	Description
clientReferenceId	string	Yes	1-128 characters; starts with a letter or digit, followed by letters, digits or <code>. _ : -</code> .
template.name	string	Yes	Use the assigned name, for example <code>indiamart</code> , without <code>.json</code> . It must be visible to this workspace.
template.version	string	No	An optional version supplied by Versalence. Currently the first 12 hexadecimal characters of the template SHA-256, not a date or semantic version. If supplied it must match the available template. Omit to resolve the current template at submission. The resolved template is frozen for execution.
agent.name	string	Yes	1-100 characters after trimming.
agent.description	string or null	No	Trimmed; currently truncated to 1,000 characters. Keep within that length to avoid truncation.
vcx.vcxAttachment	string	Yes	Exactly <code>STANDALONE</code> .
vcx.deploymentMode	string	Yes	Exactly <code>DEDICATED_BOT</code> . Other vCX properties and tenant-attachment fields are rejected.
variables	object of strings	When the template declares variables	Provide exactly the template's declared variable names and values. Versalence supplies this contract separately.
knowledge	object	No	May contain documents, websites, notes and JSON text.
knowledge.documents	array	No	Declarations for uploaded files; see below.
knowledge.websites	array	No	Website sources with explicit crawl scope; maximum 20.
knowledge.notes	string or null	No	Rich-text notes converted to safe text/Markdown for knowledge ingestion. Maximum 200,000 characters after conversion.

knowledge.json	string or null	No	A string containing valid JSON, converted into labelled knowledge text. Limit 2,000,000 characters; transformed text must also fit. This field does not create a table.
----------------	----------------	----	---

Prompt variables

A template prompt such as `You represent {{Brand Name}}.` receives the value from `variables["Brand Name"]`. For example, `"Brand Name": "Example Steel"` produces `You represent Example Steel.` Names are case-sensitive. Values must be strings, satisfy the template's required-variable checks, be at most 10,000 characters, and contain neither `{{` nor `}}`. Unknown variables are rejected. These are prompt substitutions, not arbitrary Botpress environment/configuration overrides.

There is currently no external template-discovery endpoint. Obtain the allowed name, variable names and any version pin from Versalence. The example below illustrates a template declaring `Brand Name`; use the actual assigned variable contract.

Document declarations and JSON-to-table conversion

Field	Type	Required	Description
externalSourceId	string	Yes	1-128 characters; same character rules as clientReferenceId. Unique across documents and websites in the submission.
uploadField	string	Yes	The exact multipart file-field name, for example <code>file_catalogue</code> .
convertToTable	boolean	Yes for every .json file	<code>true</code> creates a Botpress table attached to the bot's knowledge base; <code>false</code> ingests the JSON as a knowledge document. On non-JSON files omit it or send false.

The JSON flag is a literal boolean: `true` or `false`. Missing, null, `"true"`, `"yes"` and `1` are invalid for a .json upload. Filename extension matching is case-insensitive. This requirement also applies to planned UPDATE add/replace operations.

Accepted extensions: `.pdf .txt .md .json .jsonl .csv`. Maximum 50 files; 50 MiB per file (52,428,800 bytes), 200 MiB total (209,715,200 bytes). Empty files and duplicate sanitised filenames are rejected. Multipart text fields have a 16 MiB limit. JSONL and CSV uploads are knowledge documents; this API's table-conversion option is for .json only.

Table JSON must be a single object or a non-empty array of objects, with at most 19 input fields across the rows. A wrapper object is treated as one row, not automatically unwrapped into its nested array. Nested values become JSON text. Additional row-size validation applies; oversized rows are rejected. Conversion creates a table source, not a second duplicate document source.

Example table file: catalogue.json

```
[
  { "itemCode": "DEMO-001", "description": "Sample product", "unit": "piece" },
  { "itemCode": "DEMO-002", "description": "Another sample product", "unit": "piece" }
]
```

Website declarations

Field	Type	Required	Description
externalSourceId	string	Yes	Your unique source reference.
url	string	Yes	The full public HTTP/HTTPS URL, including its intended path. Private/local addresses and URLs containing credentials are not accepted.
scope	string	Yes	page for the supplied page; site for the supported crawl starting at that URL. The starting path is preserved, not silently changed to the domain root.

To import the seller page, send `https://www.24by7.com/sewa-enterprises-newdelhi/` with `scope: "page"`. Website import is asynchronous. A successful submission does not establish that every page was crawled or indexed; inspect the eventual source results and warnings.

Example payload.json

```
{
  "clientReferenceId": "im-seller-demo-001",
  "template": { "name": "indiamart" },
  "agent": {
    "name": "Example Steel Assistant",
    "description": "Dedicated seller assistant"
  },
  "vcx": {
    "vcxAttachment": "STANDALONE",
    "deploymentMode": "DEDICATED_BOT"
  },
  "variables": { "Brand Name": "Example Steel" },
  "knowledge": {
    "notes": "<p>Answer seller questions using the supplied knowledge.</p>",
    "json": "{\"supportLanguage\":\"English\"}",
    "documents": [
```

```

    { "externalSourceId": "catalogue", "uploadField": "file_catalogue", "convertToTable":
true },
    { "externalSourceId": "company-data", "uploadField": "file_company", "convertToTable":
false },
    { "externalSourceId": "terms", "uploadField": "file_terms" }
  ],
  "websites": [
    {
      "externalSourceId": "seller-page",
      "url": "https://www.24by7.com/sewa-enterprises-newdelhi/",
      "scope": "page"
    }
  ]
}

```

Curl

Examples use a POSIX shell. Replace the workspace and read the API key from your secure server environment. Save the payload above as `payload.json` and supply your actual files. Remove both a declaration and its corresponding file part if it is not needed.

```

curl --request POST \
  'https://backend.versalence.online/api/v2/agent-
builder/ext/workspaces/YOUR_WORKSPACE_ID/agent-requests' \
  --header "x-api-key: $VCX_API_KEY" \
  --header 'Idempotency-Key: im-demo-001-create-20261030' \
  --header 'Accept: application/json' \
  --form 'payload=<payload.json' \
  --form 'file_catalogue=@catalogue.json;type=application/json' \
  --form 'file_company=@company.json;type=application/json' \
  --form 'file_terms=@terms.pdf;type=application/pdf'

```

202 Accepted · `application/json` · illustrative review-ON response

```

{
  "success": true,
  "replayed": false,

```

```

"operation": {
  "operationId": "11111111-1111-4111-8111-111111111111",
  "type": "CREATE",
  "status": "pending_review",
  "stage": null,
  "agentId": "22222222-2222-4222-8222-222222222222",
  "revision": 1,
  "review": { "mode": "on", "decision": "pending", "decidedAt": null, "reason": null },
  "createdAt": "2026-10-30T10:00:00.000Z",
  "updatedAt": "2026-10-30T10:00:00.000Z",
  "warnings": [],
  "result": null,
  "error": null,
  "statusUrl": "/api/v2/agent-builder/ext/workspaces/YOUR_WORKSPACE_ID/operations/11111111-1111-4111-8111-111111111111"
}
}

```

Idempotency and duplicate sellers

- Same key, same payload and same files: HTTP 200, `replayed: true`, original operation with its current status. This does not restart execution.
- Same key with different content, filenames or file bytes: HTTP 409 `IDEMPOTENCY_KEY_REUSED`.
- A different key for a clientReferenceId already in progress or created: HTTP 409 `CLIENT_REFERENCE_EXISTS`.
- If a network failure leaves acceptance uncertain, replay the identical request with the original key. Do not generate a new key merely because the response was lost.
- A failed operation is not restarted by replay. No public retry endpoint is available yet; coordinate recovery with Versalence, especially if cleanup requires attention.

Get operation status: available

GET /workspaces/{workspaceId}/operations/{operationId}

Requires `operations.read`. Use the operationId or relative statusUrl returned by CREATE. Unknown operations and operations outside your partner/workspace are not disclosed. An operation failure is represented inside a successful status lookup; inspect `operation.status`, not only the HTTP status or top-level success.

Curl

```
curl \
  'https://backend.versalence.online/api/v2/agent-builder/ext/workspaces/YOUR_WORKSPACE_ID/operations/11111111-1111-4111-8111-111111111111' \
  --header "x-api-key: $VCX_API_KEY" \
  --header 'Accept: application/json'
```

Statuses

Status	Meaning / action
<code>pending_review</code>	Stored, awaiting internal review. No creation job starts before approval. Poll less frequently while waiting for a human decision.
<code>queued</code>	Approved or accepted with review OFF; awaiting execution.
<code>running</code>	Provisioning is in progress. The optional stage describes the current creation step.
<code>succeeded</code>	Terminal. Creation completed without reported warnings; current lifecycle is draft.
<code>succeeded_with_warnings</code>	Terminal. Creation completed with limitations; inspect warnings before acceptance.
<code>failed</code>	Terminal. Inspect the safe error code and contact Versalence for recovery.
<code>rejected</code>	Terminal. Internal review declined the submission; inspect review.reason.

Non-terminal status responses currently include `Retry-After: 5`. Wait at least that many seconds between polls and back off on network/server errors. Stop polling on a terminal state. Do not treat a null stage or a repeated response as failure. Creation can take longer than a normal HTTP request.

200 Successful lookup · illustrative completion with one text file and one table

```
{
  "success": true,
  "operation": {
    "operationId": "11111111-1111-4111-8111-111111111111",
    "type": "CREATE",
    "status": "succeeded",
```

```

"stage": null,
"agentId": "22222222-2222-4222-8222-222222222222",
"revision": 1,
"review": {
  "mode": "on", "decision": "approved",
  "decidedAt": "2026-10-30T10:02:00.000Z", "reason": null
},
"createdAt": "2026-10-30T10:00:00.000Z",
"updatedAt": "2026-10-30T10:03:00.000Z",
"warnings": [],
"result": {
  "agentId": "22222222-2222-4222-8222-222222222222",
  "lifecycle": "draft",
  "knowledgeFiles": 1,
  "tables": [{ "externalSourceId": "catalogue", "rows": 2, "status": "indexed" }],
  "websites": []
},
"error": null
}
}

```

The completion above illustrates a separate minimal submission, not the multi-source CREATE example. Counts and source results depend on the actual input. `knowledgeFiles` is not a count of all tables and websites. Source warnings refer to `externalSourceId`; retain your own mapping to filenames and URLs.

Status responses omit variable values, uploaded filenames, template contents, Botpress bot/KB IDs and webhook credentials. The vCX agentId cannot be substituted for a botId on the existing bot-credentials API. Integration credentials and activation handover must be coordinated with Versalence; no new partner activation or agent-details endpoint is included in this release.

Validation and failure handling

HTTP / code	Meaning	Action
400 / VALIDATION_FAILED	Malformed payload, fields, identifiers or Idempotency-Key.	Correct the request using error.details when present.
400 / VCX_CONFIG_INVALID	Wrong standalone values or tenant/unsupported vCX fields.	Send only the documented vcx object.
400 / VARIABLE_REQUIRED, VARIABLE_INVALID, VARIABLE_UNKNOWN	Prompt variables do not match the template contract.	Correct names/types/values.

400 / TEMPLATE_NOT_READY	The server template has unresolved preparation requirements.	Contact Versalence; do not work around it in the payload.
400 / CONVERT_TO_TABLE_INVALID, CONVERT_TO_TABLE_NOT_JSON, TABLE_INPUT_INVALID	Incorrect JSON flag or unsuitable table input.	Correct the flag or file structure.
400 / DOCUMENT_FILE_MISSING, DOCUMENT_UNREFERENCED	Multipart files do not match declarations.	Match each uploadField to exactly one part.
400 / document or website validation code	Unsupported type, empty/oversized file, duplicate source or invalid URL/scope.	Correct the reported field.
401 / 403	Authentication or permission denied.	Check key validity, partner activation and assigned scopes.
404 / TEMPLATE_NOT_FOUND, TEMPLATE_VERSION_NOT_FOUND, OPERATION_NOT_FOUND	Resource unavailable within the caller's authorised scope.	Check assigned workspace, template/version or operation ID.
409 / IDEMPOTENCY_KEY_REUSED, CLIENT_REFERENCE_EXISTS	Conflicting request or existing seller creation.	Replay the original identical request or track its operation.
413 / PAYLOAD_TOO_LARGE	Multipart text field exceeds the limit.	Reduce payload size.
5xx	Server/storage/service error.	Preserve the key and request; replay safely after recovery.

Typical validation response

```
{
  "success": false,
  "error": {
    "code": "CONVERT_TO_TABLE_INVALID",
    "message": "The submission is not valid",
    "details": [{ "code": "CONVERT_TO_TABLE_INVALID", "field": "catalogue" }]
  }
}
```

Authentication/guard failures can use the existing envelope

`{"success":false,"message":"...", "status":"failed"}` instead. Clients must tolerate both envelopes and rely on HTTP status when error.code is absent. Gateway responses may also differ.

After acceptance, failures appear on the operation. Individual file rejection can produce a warning while other sources continue; systemic upload failures fail creation. If all supplied knowledge files are rejected, creation fails. Indexing that has not completed can produce a warning rather than proof of readiness. Botpress errors and credentials are not echoed to the partner. Never equate an accepted upload with a verified useful answer from the bot.

Review and activation behaviour

1. Partner submits a complete CREATE request and stores the returned operationId and agentId.
2. With review ON, input and files remain stored on the server; an internal reviewer examines the exact revision.
3. Approval is bound to the reviewed digest. A stale or tampered revision cannot pass the execution gate.
4. The worker creates the bot from the frozen template, substitutes variables, prepares/attaches knowledge and tables, imports website sources, publishes the draft definition, configures the webhook and registers the vCX agent.
5. The partner polls for completion and checks warnings. The result remains draft pending internal acceptance and activation work.

The current switch is server configuration `AGENT_BUILDER_PARTNER_REVIEW`, default ON. Setting it to OFF affects new submissions only; it does not approve existing pending requests. It is not yet a per-workspace switch or a partner-controlled payload field. Automated expiry and scheduled storage purge are not promised by this release.

Update an agent: planned, unavailable

Design contract only. This endpoint is not implemented or deployed. Do not call it yet.

POST `/workspaces/{workspaceId}/agents/{agentId}/update-requests`

Planned scope: `agents.update`. Planned headers: x-api-key and Idempotency-Key. Multipart payload plus files, using the same uploadField matching and mandatory .json convertToTable boolean as CREATE.

Field	Planned rule
<code>baseRevision</code>	Required integer identifying the latest accepted snapshot. Changes apply to this exact snapshot, including earlier pending changes.
<code>agent</code>	Optional display-name/description changes.
<code>variables</code>	Optional patch of declared prompt-variable values. The merged result must satisfy the template.
<code>knowledge.notes</code> , <code>knowledge.json</code>	Explicit replace with value, or remove. JSON value remains a JSON-encoded string.

knowledge.documents	add, replace or remove by externalSourceId. Add/replace requires a file part; remove does not.
knowledge.websites	add, replace or remove by externalSourceId. Add/replace supplies the exact URL and scope.

Planned update-payload.json

```
{
  "baseRevision": 1,
  "agent": { "name": "Updated Example Steel Assistant" },
  "variables": { "Brand Name": "Updated Example Steel" },
  "knowledge": {
    "notes": { "op": "replace", "value": "<p>Updated seller instructions.</p>" },
    "json": { "op": "remove" },
    "documents": [
      { "op": "replace", "externalSourceId": "catalogue", "uploadField": "file_catalogue_v2",
"convertToTable": true },
      { "op": "add", "externalSourceId": "faq", "uploadField": "file_faq" },
      { "op": "remove", "externalSourceId": "terms" }
    ],
    "websites": [
      { "op": "replace", "externalSourceId": "seller-page", "url":
"https://www.24by7.com/sewa-enterprises-newdelhi/", "scope": "page" }
    ]
  }
}
```

Planned Curl — do not execute until released

```
curl --request POST \
  'https://backend.versalence.online/api/v2/agent-builder/ext/workspaces/YOUR_WORKSPACE_ID/agents/YOUR_VCX_AGENT_ID/update-requests' \
  --header "x-api-key: $VCX_API_KEY" \
  --header 'Idempotency-Key: im-demo-001-update-002' \
  --form 'payload=<update-payload.json' \
  --form 'file_catalogue_v2=@catalogue-v2.json;type=application/json' \
  --form 'file_faq=@faq.md;type=text/markdown'
```

The intended response is an asynchronous operation with type UPDATE and a new revision. Live content must remain unchanged while review is pending. An update creates a complete resulting

snapshot; it must not lose changes from the pending base. A pending unapproved update may be superseded by a later revision; approved/queued/running work must not be silently superseded.

Planned errors include stale revision, existing/unknown source IDs, repeated operations on one source, operation in progress and no changes. Changing convertToTable on replacement switches the source type. Document/table/website replacement and removal against a live bot still require implementation and acceptance tests. Final response examples will be pinned when the endpoint is built.

Delete an agent: planned, unavailable

Design contract only. No DELETE endpoint is deployed. The DELETE approval policy still requires a decision.

DELETE /workspaces/{workspaceId}/agents/{agentId}

Planned scope: `agents.delete`. Requires Idempotency-Key and `If-Match` containing the quoted latest revision number. The JSON body must confirm the same vCX agentId as the path and provide a reason.

```
{
  "confirmAgentId": "22222222-2222-4222-8222-222222222222",
  "reason": "Seller account closed"
}
```

Planned Curl — do not execute until released

```
curl --request DELETE \
  'https://backend.versalence.online/api/v2/agent-builder/ext/workspaces/YOUR_WORKSPACE_ID/agents/22222222-2222-4222-8222-222222222222' \
  --header 'x-api-key: $VCX_API_KEY' \
  --header 'Idempotency-Key: im-demo-001-delete-001' \
  --header 'If-Match: "2"' \
  --header 'Content-Type: application/json' \
  --data '{"confirmAgentId":"22222222-2222-4222-8222-222222222222","reason":"Seller account closed"}'
```

Deletion is intended to be asynchronous and restricted to the partner's agent in its workspace. The worker must remove the Botpress resource and reconcile the vCX record, record cleanup outcomes, and retain the required audit trail. A partly completed delete must remain visibly

failed/needs attention, not be reported as complete.

The proposed review behaviour is: while deletion awaits approval, hold other queued work for that agent; do not cancel it yet. Rejection, cancellation or expiry releases the hold. When approved deletion is claimed, cancel held work atomically. New updates are refused while deletion is pending. Running operations must be resolved before deletion. These concurrency rules are requirements, not deployed behaviour.

Other planned lifecycle capabilities

Capability	Planned behaviour
POST <code>/workspaces/{workspaceId}/operations/{operationId}/cancel</code>	Cancel eligible unexecuted work. Not available.
POST <code>/workspaces/{workspaceId}/operations/{operationId}/retry</code>	Create a new operation for an eligible approved revision, reconcile existing/removed resources against the ledger, and apply a bounded retry policy. Not available.
Review setting and dashboard queue	Permission-controlled per-workspace setting and review UI showing input, file/table previews, revision differences and status. Not available yet.
Activation readiness	Publication, integration readback and controlled inbound-trigger acceptance before showing active. Not completed by current CREATE success.
Retention, expiry and request quotas	Lifecycle specification defines policies; do not assume scheduled expiry/purge or the proposed quota numbers are enforced by this release.

Internal operator appendix: not partner endpoints

These routes use the base `https://backend.versalence.online/api/v2/agent-builder`, without `/ext`. They require an internal authenticated administrator with verified workspace access. A partner x-api-key does not grant access.

Method / path	Input / purpose
GET <code>/partner-operations?workspaceId=WORKSPACE_ID</code>	List pending-review operations by default; optional status filter. Returns operation, partner, revision digest and summary metadata.
GET <code>/partner-operations/{operationId}/review?workspaceId=WORKSPACE_ID</code>	Read the stored revision snapshot, digest and table previews for review.
POST <code>/partner-operations/{operationId}/approve</code>	JSON: workspaceId, expectedDigest from the reviewed snapshot, optional reason. Wrong/stale digest returns 409; successful approval queues the operation.
POST <code>/partner-operations/{operationId}/reject</code>	JSON: workspaceId and reason. Records the rejection without provisioning.

Only internal administrators issue partner identities and bind keys. A key is bound to its partner at issue; changing partner requires revocation and a new key. Do not include admin credentials, Botpress PATs or webhook secrets in partner request examples.

Integration acceptance checklist

1. Obtain the workspace, scoped key, template name and exact variable contract from Versalence.
2. Submit a disposable seller request with documents, a JSON table and the exact intended website URL; save the idempotency key, operationId and agentId.
3. With review ON, confirm pending_review and no provisioning before approval. Repeat the identical request to verify replay.
4. After internal approval, poll to a terminal state and inspect warnings and each source outcome.
5. Verify knowledge is attached to the destination bot; check table search, website import and representative conversational answers separately.
6. Keep the agent labelled draft until internal activation acceptance is complete. Do not implement partner UPDATE/DELETE calls until their release is announced.

Revision #4

Created 2026-10-30 10:24:03 UTC by Admin

Updated 2026-10-30 10:30:29 UTC by Admin