REST API: Paging, Filters, Errors and Limits

REST API · https://mriacrm.com/docs/rest-api-paging-filters-errors-and-limits/

How every list in the API is paged, filtered and sorted, what an error looks like, and the sizes and rates you will hit. Every value on this page was read from the OpenAPI specification (REST API: Reference) or measured against a test site on 2026-11-06; the Atlassian platform figures are quoted from the page named next to them.

1. Paging

Every list — `GET /leads`, `POST /leads/search`, a record's timeline, files and links, and the same for contacts, companies, deals and products — answers one envelope:

```
{
 "values": [ … ],
 "startAt": 0,
 "maxResults": 50,
 "total": 1970,
 "isLast": false
}
```

Field	Meaning
`startAt`	0-based offset of the first value.
`maxResults`	Page size actually used. Default `50` when you send none; 1 to `200` for record lists, timelines and files; 1 to `50` for link lists (Jira work items, Confluence pages, web links, spaces, boards).
`total`	Records matching the request, all pages.
`isLast`	`true` on the last page.

Requesting the next page

Add `maxResults` to the `startAt` you were given and repeat until `isLast` is `true`. On the GET lists both are query parameters; on the search endpoints they go in `pagination`:

```
curl -s "https://api.atlassian.com/svc/jira/<cloudId>/apps/f3bb4fc8-7911-405b-959b-
002b3292a3d0_<envId>/v1/leads?maxResults=2"            # page 1
curl -s "https://api.atlassian.com/svc/jira/<cloudId>/apps/f3bb4fc8-7911-405b-959b-
002b3292a3d0_<envId>/v1/leads?maxResults=2&startAt=2"  # page 2

curl -s -X POST "https://api.atlassian.com/svc/jira/<cloudId>/apps/f3bb4fc8-7911-405b-959b-
002b3292a3d0_<envId>/v1/leads/search" \
  -H "Authorization: Bearer $ACCESS_TOKEN" -H "Content-Type: application/json" \
  -d '{"pagination": {"startAt": 2, "maxResults": 2}}'
```

Page 2 of a 2-per-page walk over a test site's leads:

```
{
 "values": [ … 2 lead objects … ],
 "startAt": 2,
 "maxResults": 2,
 "total": 1970,
 "isLast": false
}
```

startAt must be a multiple of maxResults. Pages are aligned: walking them in order always satisfies this, but `startAt=3&maxResults=2` is rejected:

```
{
 "error": {
  "code": "VALIDATION_ERROR",
  "message": "Request validation failed",
  "errorId": "74a327a0",
  "fields": [
   {
    "field": "startAt",
    "message": "startAt must be a multiple of maxResults"
   }
  ]
```

```
  }
 }
```

The largest accepted offset is 200,000, and `maxResults` outside 1–200 is a 400 `VALIDATION_ERROR` that names the field:

```
{
 "error": {
  "code": "VALIDATION_ERROR",
  "message": "Request validation failed",
  "errorId": "5f6c6f5a",
  "fields": [
   {
    "field": "maxResults",
    "message": "maxResults must not be greater than 200"
   }
  ]
 }
}
```

A page of 200 leads took about ten seconds on the test site against one to two seconds for a page of 50, so keep pages small when latency matters more than call count.

2. Filters

Each entity has two read endpoints. `GET /{records}` takes a handful of query parameters; `POST /{records}/search` takes a JSON `filter` with every option, including custom fields. Search is a read despite the verb and needs only `read:crm:custom`.

```
# query parameters: comma-separated ids, dates as YYYY-MM-DD or RFC 3339
curl -s "https://api.atlassian.com/svc/jira/<cloudId>/apps/f3bb4fc8-7911-405b-959b-
002b3292a3d0_<envId>/v1/leads?statusIds=<id>,<id>&updatedAfter=2026-11-01&orderBy=-updatedAt" \
  -H "Authorization: Bearer $ACCESS_TOKEN"

# the same as a search body
curl -s -X POST "https://api.atlassian.com/svc/jira/<cloudId>/apps/f3bb4fc8-7911-405b-959b-
002b3292a3d0_<envId>/v1/leads/search" \
  -H "Authorization: Bearer $ACCESS_TOKEN" -H "Content-Type: application/json" -d '{
  "filter": {"statusIds": ["<id>", "<id>"], "updatedAfter": "2026-11-01"},
```

```
  "sort": {"field": "updatedAt", "direction": "desc"},
  "pagination": {"maxResults": 50}
}'
```

A text search for a company name, sorted by last update:

```
{"filter": {"search": "Northstar"}, "sort": {"field": "updatedAt", "direction": "desc"},
"pagination": {"maxResults": 5}}
```


```
{
 "values": [
  {
   "id": "4cf0f383-…",
   "name": "REST docs demo: Northstar Labs",
   "contactName": "Tomas Vidal",
   "contactEmail": "tomas.vidal@northstar.example",
   "contactPhoneNumber": null,
   "companyName": "Northstar Labs",
   "companyId": null,
   "contactId": null,
   "assigneeId": null,
   "status": {
    "id": "4e0326fa-…",
    "name": "New",
    "color": "color.text.success",
    "backgroundColor": "color.background.success",
    "sortOrder": 1
   },
   "createdAt": "2026-11-06T17:43:22.000Z",
   "updatedAt": "2026-11-06T17:43:35.000Z",
   "nextActivityDate": null,
   "lastActivityDate": "2026-11-06T17:43:51.000Z"
  }
 ],
 "startAt": 0,
 "maxResults": 5,
 "total": 1,
 "isLast": true
}
```

Filter fields per entity

Names as the specification lists them. The GET column is the subset available as query parameters; everything is available in the search body.

Entity	Search filter	GET parameters
Leads	`search, statusIds, sourceIds, tagIds, assigneeIds, assignedToMe, contactIds, companyIds, productIds, jiraIssueKeys, hasJiraIssueLinks, createdAfter, createdBefore, updatedAfter, updatedBefore, nextActivityPresets, noNextActivity, nextActivityAfter, nextActivityBefore, lastActivityAfter, lastActivityBefore, noActivityInLastDays, noLastActivity, customFilters`	`search, statusIds, sourceIds, tagIds, assigneeIds, createdAfter, updatedAfter`
Contacts	`search, tagIds, companyIds, typeIds, companyTypeIds, createdByMe, withOpenLeads, withOpenDeals, hasCompany, jiraIssueKeys, hasJiraIssueLinks, createdAfter, createdBefore, updatedAfter, updatedBefore, customFilters`	`search, tagIds, companyIds, createdAfter, updatedAfter`
Companies	`search, tagIds, typeIds, country, city, createdByMe, withOpenLeads, withOpenDeals, hasContacts, jiraIssueKeys, hasJiraIssueLinks, createdAfter, createdBefore, updatedAfter, updatedBefore, customFilters`	`search, tagIds, country, city, createdAfter, updatedAfter`
Deals	`search, assignedToMe, pipelineIds, pipelineStageIds, assigneeIds, typeIds, sourceIds, tagIds, productIds, companyIds, contactIds, minAmount, maxAmount, probabilityFrom, probabilityTo, jiraIssueKeys, hasJiraIssueLinks, createdAfter, createdBefore, updatedAfter, updatedBefore, closingAfter, closingBefore, nextActivityPresets, noNextActivity, nextActivityAfter, nextActivityBefore, lastActivityAfter, lastActivityBefore, noActivityInLastDays, noLastActivity, customFilters`	`search, pipelineIds, pipelineStageIds, assigneeIds, tagIds, createdAfter, updatedAfter`
Products	`search, categoryIds, unitTypeIds, priceFrom, priceTo, createdByMe, withOpenLeads, withOpenDeals, companyIds, jiraIssueKeys, hasJiraIssueLinks, createdAfter, createdBefore, updatedAfter, updatedBefore, customFilters`	`search, categoryIds, unitTypeIds, priceFrom, priceTo, createdAfter, updatedAfter`

Rules that apply to every filter

- `search` is free text, at most 100 characters, matched against the names on the record: a company name or a contact's first name finds the lead.
- Id lists (`statusIds`, `tagIds`, `companyIds`…) hold up to 50 UUIDs and match any of them. Ids come from the reference endpoints: `GET /leadStatuses`, `/leadSources`, `/leadTags`, `/dealPipelines` and `/dealPipelines/{id}/stages`, `/dealTypes`, `/recordTypes?entity=COMPANY|CONTACT`, `/productCategories`. A value that is not a UUID is a 400 (“each value in statusIds must be a UUID”).
- Dates — `createdAfter`, `createdBefore`, `updatedAfter`, `updatedBefore`, `closingAfter` / `closingBefore` on deals — take a date (`2026-11-01`) or a date-time (`2026-11-01T09:00:00Z`); anything else is a 400 (“updatedAfter must be a Date instance”). *After* is an inclusive lower bound; a *Before* date includes the whole UTC day.
- Activity windows: `nextActivityPresets` takes any of `OVERDUE`, `TODAY`, `NEXT_7_DAYS`, `THIS_MONTH` (OR-ed, UTC); `noActivityInLastDays` is 1–3650.
- Filters are AND-ed with each other; a filter you leave out does not restrict. A query parameter or body property the endpoint does not know is rejected (“property expand should not exist”), so a typo in a filter name is a 400, not a silently ignored filter.

Custom fields

`customFilters` is a list of up to 20 items, each `{fieldId, op, …}`. Field ids come from `GET /leadCustomFields` and its siblings. `op` is one of `is_empty, is_not_empty, contains, equals, not_equals, starts_with, gt, gte, lt, lte, between, in_any, in_all, in_none`; the value goes in `textValue` (≤ 2,000 characters), `numberValue` / `numberValueTo`, `dateValue` / `dateValueTo` or `values` (≤ 50). Leads with a number field above zero:

```
{
 "filter": {
  "customFilters": [
   {
    "fieldId": "<field id>",
    "op": "gt",
    "numberValue": 0
   }
  ]
 },
```

```
 "pagination": {
  "maxResults": 1
 }
}
```

```
{
 "values": [ … ],
 "startAt": 0,
 "maxResults": 1,
 "total": 6,
 "isLast": false
}
```

An unknown fieldId is not an error. A custom filter whose `fieldId` matches no field is ignored and the search returns everything, so check the id against `/leadCustomFields` yourself. An unknown `op` is rejected with the list of valid ones.

3. Sorting

On GET lists: `orderBy=<field>` ascending, `orderBy=-<field>` descending. In a search body: `"sort": {"field": "<field>", "direction": "asc"|"desc"}`.

Entity	Sort fields
Leads	`name, contactName, companyName, contactEmail, status, assignee, createdAt, updatedAt, nextActivityDate, lastActivityDate`
Contacts	`name, email, phoneNumber, company, createdAt, updatedAt, countOpenLeads, countOpenDeals`
Companies	`name, country, city, createdAt, updatedAt, countOpenLeads, countOpenDeals, countLinkedContacts`
Deals	`name, pipelineStage, company, primaryContact, assignee, amount, probability, closeDate, lastActivityDate, nextActivityDate, createdAt, updatedAt, lossReason`
Products	`name, price, sku, category, unitType, createdAt, updatedAt`

A field outside the list is a 400 that spells out the choices:

```
{
 "error": {
  "code": "VALIDATION_ERROR",
  "message": "Request validation failed",
  "errorId": "9fab36bc",
  "fields": [
   {
    "field": "sort.field",
    "message": "field must be one of the following values: name, contactName, companyName,
contactEmail, status, assignee, createdAt, updatedAt, nextActivityDate, lastActivityDate"
   }
  ]
 }
}
```


4. Errors

Mria CRM answers every non-2xx of its own with one envelope. `error.code` is stable and is what to branch on; `error.fields` is present on validation errors, one entry per field; `error.errorId` is a short id logged on the server — quote it when you contact support. Creating a lead with only a one-letter name:


```
{
 "error": {
  "code": "VALIDATION_ERROR",
  "message": "Request validation failed",
  "errorId": "1309fe1d",
  "fields": [
   {
    "field": "name",
    "message": "name must be longer than or equal to 2 characters"
   },
   {
    "field": "contactName",
    "message": "contactName must be longer than or equal to 2 characters; contactName should not
be empty"
   },
   {
    "field": "contactEmail",
```

```
    "message": "contactEmail must be an email; contactEmail should not be empty"
   }
  ]
 }
}
```

Two answers come from the Atlassian gateway before the request reaches the app and have a different shape: a missing or invalid token (JSON with `code` and `message`), and a token whose scopes do not cover the install (a plain string). Both are shown on REST API: Authorization with curl.

Status	When	Example
`400` `INVALID_JSON`	The body is not JSON.	`{ "error": { "code": "INVALID_JSON", "message": "Request body is not valid JSON", "errorId": "dc11e5be" } }`
`400` `VALIDATION_ERROR`	A field is missing, too short, too long, the wrong type, unknown, or refers to something that does not exist — an unknown `statusId` on a PATCH is a 400, not a 404. A malformed id in the path is a 400 too.	`{ "error": { "code": "VALIDATION_ERROR", "message": "Request validation failed", "errorId": "fdfd69d8", "fields": [ { "field": "statusId", "message": "Lead status 6b7f0895-… not found" } ] } }`

Status	When	Example

`401`	No token, or an expired, revoked or malformed one. Gateway shape.	<pre>{
		 "code": 401,
		 "message": "Unauthorized"
		}</pre>

`403`	The token lacks the scope for this install (gateway, plain string); the user's CRM role lacks the permission, or a write on a site whose license is inactive (app envelope); or App REST APIs are switched off for the site (`REST APIs are not allowed for appId=…`).	*Role and license cases read from the specification; no read-only user was available on the test site.*

`404` `NOT_FOUND`	No record with that id.	<pre>{
		 "error": {
		  "code": "NOT_FOUND",
		  "message": "The requested resource was not found",
		  "errorId": "45369f7e"
		 }
		}</pre>

`404` `ROUTE_NOT_FOUND`	No such path, or the path exists but not for this method (there is no 405).	<pre>{
		 "error": {
		  "code": "ROUTE_NOT_FOUND",
		  "message": "No DELETE route matches the requested path",
		  "errorId": "772b6878"
		 }
		}</pre>

Status	When	Example
`409`	The request conflicts with the record's state: `EMAIL_ALREADY_EXISTS` when `createNewContact` names an email a Contact already has; `LEAD_CONVERTED` when you edit a converted lead; `CONTACT_HAS_COMPANY` and `MEETING_NOT_EDITABLE` likewise.	`{ "error": { "code": "EMAIL_ALREADY_EXISTS", "message": "A contact with this email already exists", "errorId": "3e3c85b1" } }`
`413`	Request body of 5 MB or more. Gateway, plain string.	`HTTP 413` `"Request body too long"`
`429`	Rate limited by the Atlassian platform (section 5). Wait for the window to reset, then retry.	*Not reached: 40 parallel reads were all 200.*
`500`	Unexpected. Retry once; if it persists, send `errorId` to support.	*Not seen.*

Writes that replace

A PATCH is a partial update — a field you leave out is unchanged and `null` clears it — but `tagIds` and `productIds` replace the whole set. Sending one tag to a lead that has another leaves it with one tag, not two:

```
{"tagIds": ["<tag id B>"]}
```

```
{
 "tags": [
  {
   "id": "1093f23b-…",
   "name": "Follow-up"
  }
 ]
}
```

```
… the rest of the lead unchanged; the tag sent earlier is gone.
```

There is no endpoint that creates a tag, status or source: ids must already exist in Settings.

5. Limits

What	Limit	Source
Page size	1–200 (links: 1–50); default `50`	Specification; default and 201 measured
Offset	`startAt` ≤ 200,000, page-aligned	Specification; both measured
Search text	100 characters	Specification
Ids per filter list	50	Specification
Custom filters	20 per search; `textValue` 2,000 characters	Specification
Names	Lead, contact, company, deal and product `name`: 2–150 characters	Specification (150); minimum measured: a one-character name answers “name must be longer than or equal to 2 characters”
Note `content`	1–10,000 characters	Specification; measured: 10,000 accepted, 10,001 and empty rejected
Files	10,485,760 bytes (10 MB) per file, checked on the start call (“File exceeds the 10 MB size limit”) and again on upload	Specification; start call measured. See REST API: Attaching Files.
Request body	Under 5 MB. A 4.5 MB body was parsed; 5 MB answered 413 “Request body too long”	Measured; matches the 5 MB invocation payload limit on Atlassian: Forge invocation limits

What	Limit	Source
Access token	3,600 seconds	Measured (`expires_in` on every token response)
Refresh token	Rotates on every use	Atlassian: Implementing the Refresh Token Flow; rotation measured
Rate	Every response carries `X-Ratelimit-Limit: 300`, `X-Ratelimit-Remaining` and `X-Ratelimit-Reset` (Unix seconds, always within a few seconds of now). Atlassian's platform figures: 300 invocations per second and 7,000 per minute per install, 1,200 per minute per user; over the limit the platform answers 429.	Headers measured; figures from Atlassian: Forge invocation limits. 40 parallel reads in 8.3 s: all 200.
Time per call	25 seconds of app runtime, after which the platform stops the invocation. Measured: one to two seconds for a typical call, about ten for a 200-record page	Atlassian: Forge invocation limits

6. No webhooks yet

The API does not push changes. To keep another system in step, poll: every record carries `updatedAt`, and every list accepts `updatedAfter`. Remember the time your last run started, ask for everything updated after it, sort by `updatedAt` and page through:

```
curl -s "https://api.atlassian.com/svc/jira/<cloudId>/apps/f3bb4fc8-7911-405b-959b-
002b3292a3d0_<envId>/v1/leads?updatedAfter=2026-11-06T09:00:00Z&orderBy=updatedAt&maxResults=200"
\
  -H "Authorization: Bearer $ACCESS_TOKEN"
```

Overlap the window by a minute rather than trusting the clocks exactly, and treat a record you have seen before as an update. Deleted records do not appear — compare ids against your copy for those. Webhooks are planned; this page will say so when they ship.

Related Pages

- REST API: Reference — every parameter and schema, including the per-entity filters in full.

- REST API: Authorization with curl — getting the token these calls carry.
- REST API: Attaching Files — the 10 MB upload flow.
- Daily Lead Sync — a polling sync built on search and `updatedAt`.

This API is supported and actively developed, so it can change without notice. It builds on Atlassian's Forge App REST APIs (Preview). Send feedback to the Mria CRM team.