

Jetdocs API Documentation

V0.5 (beta)

Summary

Jetdocs provides read APIs for various resources for an organization using the Jetdocs platform. This document contains descriptions for all supported APIs. The following resources currently have API support:

tickets.read
templates.read
categories.read
users.read
user-groups.read
workspaces.read

Authentication

Jetdocs API access is restricted to only to Organizational Admins currently. Organizational Admins are configured in the “User Management” page inside Jetdocs, under the “Organization Users” tab.

The user API token used for authentication with the Jetdocs API can be found in the “Profile & Integrations” page inside Jetdocs, under “API Key”. This key must be provided with every API request as a request header, in the following format:

```
{ "x-api-key": <api_key> }
```

Pagination

For methods that return lists of objects, the results may be paginated. These methods will always accept the optional “limit” and “cursor” GET parameters. Limit defines the page size of results to fetch (recommended values are 50 to 1000). Cursor is used for fetching subsequent pages, by passing in the next_cursor value from the response. API responses will always contain a “response_metadata” object, which contains “next_cursor”. To fetch the first page of results (including the next_cursor value), the cursor GET parameter should not be included.

Supported Methods

`tickets.read`

Returns a list of tickets submitted by the organization, including per-ticket submission details, workflow data, and historical data for reassigns, status changes, and comments. Results are sorted by ticket creation date descending. Returns results in JSON format. This method uses pagination.

URL: GET <https://teams.jetdocs.io/api/v1/tickets>

Parameters:

- limit
- offset
- templateId (optional)
- ticketId (optional)
- status (filter by specific status, comma separated. Valid statuses are NEW,OUTSTANDING,COMPLETED,REJECTED,CANCELLED)
- asc (set to 1 to sort by oldest tickets first, or don't include for newest first)

Example: GET

<https://teams.jetdocs.io/api/v1/tickets?limit=100&cursor=XYZ&status=OUTSTANDING&asc=1>

Response Parameters:

response_metadata: { next_cursor, num_results }

tickets : [List of fetched tickets]

Object (ticket) parameters:

id

workspace_idx: workspace the ticket & template belong to

category_id: ID of category the template belongs to

template_id: ID of template

name: Ticket name

submitter: email of submitting user

current_responder: email or user group the ticket is currently assigned to

current_responder_emails: array of emails the ticket is assigned to (if assigned to a user group, then the emails for that user groups)

status

title

form_values: list of form inputs, type of input, and submitted value

current_step: current step the ticket is on (first incomplete step)

workflow_steps: list of ticket workflow steps and relevant metadata

archived

sla_breach_time: Timestamp for ticket SLA breach (if enabled)

tags

reassign_timeline: historical timeline of ticket re-assigns

status_change_timeline: historical timeline of status changes

comment_timeline: historical timeline of comments

workflow_timeline: historical timeline of workflow status updates

templates.read

Returns a list of templates used by the organization, including details of the form inputs, workflow steps, and additional data configured in the Catalog Builder UI. Returns results in JSON format.

URL: GET <https://teams.jetdocs.io/api/v1/templates>

Parameters: None

Example: GET <https://teams.jetdocs.io/api/v1/templates>

Response Parameters:

response_metadata: { num_results }

templates: [List of fetched templates]

Object (template) parameters:

id

workspace_idx: workspace the template belongs to

category_id: ID of category the template belongs to

name

description

long_description

fields: list of input fields used by the template, including input instruction, type of field, and options

workflow_steps: list of workflow steps used by the template, including instruction, responder, and additional options

default_priority

sla_settings: configuration data for template SLA, if enabled

additional_responders: additional responders for the template

view_only_users: view only users for the template

categories.read

Returns a list of categories used by the organization. Returns results in JSON format.

URL: GET <https://teams.jetdocs.io/api/v1/categories>

Parameters: None

Example: GET <https://teams.jetdocs.io/api/v1/categories>

Response Parameters:

response_metadata: { num_results }
categories: [List of fetched categories]

Object (category) parameters:

id
workspace_idx: workspace the category belongs to
name
description
long_description

users.read

Returns a list of an organization's users. Returns results in JSON format.

URL: GET <https://teams.jetdocs.io/api/v1/users>

Parameters: None

Example: GET <https://teams.jetdocs.io/api/v1/users>

Response Parameters:

response_metadata: { num_results }
users: [List of fetched users]

Object (user) parameters:

email
role: Organizational role of the user, "ADMIN" for Organizational Admin, and "MEMBER" for Member
has_seat: boolean, true if the user has a seat assigned to them within Jetdocs

user-groups.read

Returns a list of an organization's user groups, across all workspaces. Returns results in JSON format.

URL: GET <https://teams.jetdocs.io/api/v1/user-groups>

Parameters: None

Example: GET <https://teams.jetdocs.io/api/v1/user-groups>

Response Parameters:

response_metadata: { num_results }
user_groups: [List of fetched user groups]

Object (user group) parameters:

id

workspace_idx: workspace the user group belongs to

name

description

users: list of user emails that belong to the user group

workspaces.read

Returns a list of an organization's workspaces, including workspace configuration and a list of workspace members, and their workspace role. Returns results in JSON format.

URL: GET <https://teams.jetdocs.io/api/v1/workspaces>

Parameters: None

Example: GET <https://teams.jetdocs.io/api/v1/workspaces>

Response Parameters:

response_metadata: { num_results }

workspaces: [List of fetched workspaces]

Object (user group) parameters:

workspace_idx: index/ID of the workspace

name

include_all_members: boolean, true if the workspace automatically includes all organization's members

external: boolean, true if the workspace is configured to be used for external user tickets

members: list of workspace member emails, and their workspace role (Admin, Editor or Member)