### Get Tickets

Gets a list of the tickets that match the search keyword.

#### HTTP Request

GET /api/sonar/tickets

Request using cURL

curl -H "Authorization: Bearer <API\_KEY>" \ https://HOSTNAME/api/sonar/tickets

Request Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Required | Type | Description | Note |
| offset | X | 32-bit integer | Number of records to skip | Default: 0 |
| limit | X | 32-bit integer | Maximum number of records to load | Minimum 0 to maximum 1000. If not specified, 1000 |
| from | X | Date | Start date | yyyy-MM-dd HH:mm:ssZ format. The start date is included in the range. |
| to | X | Date | End date | yyyy-MM-dd HH:mm:ssZ format. The end date is included in the range. |
| statuses | X | List (of strings) | Status | Comma-separated list. Refer to the below status codes. |
| keywords | X | String | Keyword |  |
| priorities | X | List (of 32-bit integers) | Priority list | High (3), Medium (2), Low (1). Comma-sepated list |
| assignees | X | List (of strings) | List of assignees | Comma-separated user GUID list |
| approvers | X | List (of strings) | List of approvers | Comma-separated user GUID list |
| sort\_type | X | String | Sorting type | ASCor DESC |
| sort\_column | X | String | Sorting key | id or created\_at or updated\_at or closed\_at |

Ticket status code

* New (NEW)
* Assigned (ASSIGNED)
* In process (IN\_PROGRESS)
* Submitted (SUBMITTED)
* Approved (APPROVED)
* Rejected (REJECTED)
* Closed (CLOSED)

#### Success Response

{ "total": 15, "tickets": [ { "id": 2, "repo\_guid": "5f0ba741-7551-400d-8bd6-1f14a6e8536d", "repo\_name": "Threat", "guid": "49272877-75f2-4c2f-9301-d21c4f9a106d", "title": "Attempt to collect web server settings: 20.0.31.172", "priority": "LOW", "status": "ASSIGNED", "format": "JSON", "count": 7, "attack": true, "incident": false, "assignees": [ { "company\_guid": "6fbe27b7-f1ae-4d7a-a1a5-76d8fa9aa311", "company\_name": "Logpresso", "user\_guid": "bfd00bb0-be99-4fd5-8380-166f544975fa", "user\_name": "Joshua", "task\_type": "ASSIGNEE", "task\_status": "ASSIGNED", "x\_login": null, "x\_user": null, "x\_dept": null } ], "approvers": [], "created": "2022-09-14 17:34:19+0900", "updated": "2022-09-14 23:55:29+0900", "closed": null, "x\_login": null, "x\_user": null, "x\_dept": null } ]}

* (32-bit integer) total
* (Array) tickets
* **id** (32-bit integer): Ticket ID
* **repo\_guid** (String): Ticket type GUID
* **repo\_name** (String): Ticket type name
* **site\_guid** (String): Site GUID
* **site\_name** (String): Site name
* **guid** (String): Ticket GUID
* **title** (String): Title
* **priority** (String): Ticket priority. One of the following: HIGH, MEDIUM, LOW.
* **status** (String): Ticket status. One of the following: NEW, ASSIGNED, IN\_PROGRESS, SUBMITTED, APPROVED, REJECTED, CLOSED.
* **format** (String): Ticket format. One of the following: JSON, MARKDOWN, PLAIN. Threat detection ticket uses JSON format.
* **count** (32-bit integer): Number of merged duplicate ticket
* **attack** (Boolean): Whether the detection is false or not, recorded after analysis. Logged as true if the detection is true.
* **incident** (Boolean): Whether an incident occurred or not, recorded after analysis. Recorded as true if the incident requires an immediate response, such as an endpoint infection.
* **assignees** (Array): Ticket assignee list
* **company\_guid** (String): Company (tenant) GUID
* **company\_name** (String): Company (tenant) name
* **user\_guid** (String): Assignee identifier
* **user\_name** (String): Assignee GUID
* **task\_type** (String): Always ASSIGNEE
* **task\_status** (String): ASSIGNED or IN\_PROGRESS or CLOSED
* **x\_login** (String): Login user name logged when logged when the assignee account is deleted
* **x\_user** (String): User name logged when logged when the assignee account is deleted
* **x\_dept** (String): Department name logged when logged when the assignee account is deleted
* **approvers** (Array): List of ticker approvers
* **company\_guid** (String): Company (tenant) GUID
* **company\_name** (String): Company (tenant) name
* **user\_guid** (String): Approver GUID
* **user\_name** (String): Approver name
* **task\_type** (String): Always APPROVER
* **task\_status** (String): ASSIGNED or IN\_PROGRESS or CLOSED
* **x\_login** (String): Login user name logged when logged when the approver account is deleted
* **x\_user** (String): User name logged when logged when the approver account is deleted
* **x\_dept** (String): Department name logged when logged when the approver account is deleted
* **created** (String): Date and time of creation (yyyy-MM-dd HH:mm:ssZ)
* **updated** (String): Date and time of last modification (yyyy-MM-dd HH:mm:ssZ)
* **closed** (String): Date and time of ticket closing (yyyy-MM-dd HH:mm:ssZ)
* **x\_login** (String): Login user name logged when logged when the ticket author account is deleted
* **x\_user** (String): User name logged when logged when the ticket author account is deleted
* **x\_dept** (String): Department name logged when logged when the ticket author account is deleted
* **x\_site** (String): Site name logged when logged when the site is deleted

#### Error Responses

offset or limit value is not an integer

HTTP status code 400

{ "error\_code": "invalid-argument", "error\_msg": "'offset' parameter should be int type"}

offset or limit value is negative

HTTP status code 400

{ "error\_code": "invalid-argument", "error\_msg": "'offset' must be greater than or equal to 0."}

Invalid `from`, `to` date formats

HTTP status code 400

{ "error\_code": "invalid-argument", "error\_msg": "'from' parameter should be date format (yyyy-MM-dd HH:mm:ss+0000)"}

Undefined status code

HTTP status code 400

{ "error\_code": "invalid-argument", "error\_msg": "'statuses' should contain elements that is one of NEW, ASSIGNED, IN\_PROGRESS, SUBMITTED, APPROVED, REJECTED, CLOSED."}

Undefined priority value

HTTP status code 400

{ "error\_code": "invalid-argument", "error\_msg": "element of priorities should be one of 1 (LOW), 2 (MEDIUM), 3 (HIGH). input is 4"}

Undefined sorting type

HTTP status code 400

{ "error\_code": "invalid-argument", "error\_msg": "sort\_type should be one of ASC or DESC. input is NONE"}

Invalid sorting key column

HTTP status code 400

{ "error\_code": "invalid-argument", "error\_msg": "sort\_column should be one of id, created\_at, updated\_at, closed\_at."}

List of assignees or approvers has non-GUID value

HTTP status code 400

{ "error\_code": "invalid-argument", "error\_msg": "assignees should contains only guid values."}