

# Get reports via API

Jan 17, 2025

API provided by the HappyFox helpdesk is a RESTful web service. It supports operations like creating a ticket, adding updates to a ticket, listing tickets and users of the Helpdesk. It supports JSON, Form Url encoded and Multipart Form Data formats as payload.

## **Requirements:**

The API requires following skills in any programming language.

- Making HTTP requests (using GET and POST HTTP methods as a minimum requirement).
- Doing HTTP Basic Authentication.
- Generating and reading data in the JSON format.
- Optionally making HTTP POST requests using content type of "multipart/form-data" (needed for ticket attachments)

## **Documentation Conventions:**

The documentation indicates parameters that need to be replaced with actual values using the format . The entire string including the enclosing < and > should be replaced.

For example, if the parameter *email* is it should be replaced with the required email address

**Endpoint url format :** /api/1.1/<module>/

The above tail url has to be suffixed with your HappyFox account url as shown below

Eg. <https://customersupport.happyfox.com/> (<https://customersupport.happyfox.com/>)api/1.1/

*Note: If you are using a custom domain, please use the custom domain url*

Note: All timestamps in API responses are in UTC

*API key and auth code are to be passed in Basic HTTP authentication format only.*

<module> - refers the module which is being accessed viz., /reports/, /users/ etc A full constructed example url is shown below.



<https://customersupport.happyfox.com/> (<https://customersupport.happyfox.com/api/1.1/json/reports/>)

This knowledge base article consists of all operations that can be done via the reports endpoint.

1. Get list of all reports
2. View report summary
3. View tabular view of a report
4. View response stats
5. View Staff Performance
6. View Staff Activity
7. View Contact Activity
8. View SLA Performance
9. Runtime filtering and sorting of reports

1. Get list of all reports

Provides a paginated list of all reports that have been created and saved in the Main Menu --> All Reports module.

**API Endpoint:** /api/1.1/json/reports/

**Method:** GET

**Example Response:**

```
{
  "last_index": 7,
  "rows": [
    {
      "description": "",
      "name": "Test report",
      "id": 7
    },
    {
      "description": "All tickets that are created in the last week",
      "name": "Tickets created in last one week",
      "id": 3
    }
  ],
  "page_count": 1,
  "start_index": 1,
  "end_index": 7
}
```

## 2. View Report Summary

To see the summary of a report showing ticket count, completed count, assigned count etc, in one shot.

**API Endpoint:** /api/1.1/json/report/<id of report>/

The id of a report can be obtained from the /reports/ endpoint.

**Method:** GET

## Example Response

```
{  
  "ticket_count": 0,  
  "completed_count": 0,  
  "assigned_count": 0,  
  "pending_count": 0,  
  "unassigned_count": 0  
}
```

### 3. View Tabular View of a report

This endpoint allows access to the most frequently used option in the reports module from the UI, Tabular view.

**API Endpoint:** /api/1.1/json/report/<id of report>/tabulardata/

**Method:** GET

**Example Response:**

```

{
  "rows": [
    {
      "display_id": "#NCC00003439",
      "due_date": "24 Apr 2019",
      "status_id": 5,
      "status_name": "open",
      "assignee": "Pradeep CE",
      "id": 3439,
      "subject": "Test Ticket"
    },
    {
      "display_id": "#CFC00003432",
      "due_date": "26 Apr 2019",
      "status_id": 3,
      "status_name": "new",
      "assignee": "-",
      "id": 3432,
      "subject": "Example ticket"
    }
  ],
  "sort_key": "duedate",
  "page_count": 5,
  "end_index": 10,
  "sort_order": "a",
  "last_index": 42,
  "start_index": 1
}

```

The number of results is 10 by default. Maximum page size can be extended to 50 and can be iterated using page\_count field.

Eg. <https://customersupport.happyfox.com/api/1.1/json/report/1/tabulardata/?size=50&page=1>  
(<https://customersupport.happyfox.com/api/1.1/json/report/1/tabulardata/?size=50&page=1>)

Response:

```
{
  "rows": [
    {
      //row1
    },
    {
      //row2
    },
    ....
    ....
    ....
    {
      //row n
    }
  ],
  "sort_key": "duedate",
  "page_count": 11,
  "end_index": 547,
  "sort_order": "a",
  "last_index": 547,
  "start_index": 1
}
```

#### 4. View Response Stats

**API Endpoint:** /api/1.1/json/report/<id of report>/responsestats/

**Method:** GET

**Example Response:**

```
[
  {
    "avg": 54444,
    "unit": "Responses",
    "average_first_response_time": [
      "79.55% responded after 24 hours",
      "1.14% responded within 24 hours",
      "19.32% responded within first 2 hours"
    ]
  },
  {
    "avg": 52863,
    "unit": "Responses",
    "average_response_time": [
      "76.34% responded after 24 hours",
      "1.08% responded within 24 hours",
      "22.58% responded within first 2 hours"
    ]
  },
  {
    "avg": 2,
    "unit": "Responses",
    "average_no_of_replies": [
      "84.26% responded after 24 hours",
      "3.7% responded within 24 hours",
      "12.04% responded within first 2 hours"
    ]
  },
  {
    "average_no_of_response_to_completed_state": [
      "97.06% responded after 24 hours",
```



```
        "0.0% responded within 24 hours",  
        "2.94% responded within first 2 hours"  
    ],  
    "avg": 2,  
    "unit": "Responses"  
}  
]
```

## 5. View Staff Performance

**API Endpoint:** /api/1.1/json/report/<id of report>/staffperformance/

**Method:** GET

**Example Response:**

```

{
  "last_index": 13,
  "staff_performance": [
    {
      "average_time_spent": 0.0,
      "staff_id": 3,
      "name": "Agent 1",
      "average_no_of_responses": 0.0,
      "average_time_to_ticket_complete": 0,
      "average_first_response_time": 0.0,
      "average_response_time": 0.0,
      "average_no_of_responses_for_completion": 0.0
    },
    {
      "average_time_spent": 0.0,
      "staff_id": 4,
      "name": "Agent 2",
      "average_no_of_responses": 0.0,
      "average_time_to_ticket_complete": 0,
      "average_first_response_time": 0.0,
      "average_response_time": 0.0,
      "average_no_of_responses_for_completion": 0.0
    }
  ],
  "sort_key": "name",
  "page_count": 2,
  "end_index": 10,
  "sort_order": "a",
  "start_index": 1
}

```

## 6. View Staff Activity

This provides a paginated list of agents (in this report) and their ticket activity sorted by agent name.

**API Endpoint:** /api/1.1/json/report/<id of report>/staffactivity/

**Method:** GET

**Example Response:**

```
{  
  "staff_activity": [  
    {  
      "assigned": 68,  
      "staff_id": 3,  
      "name": "Agent 1",  
      "private_notes": 0,  
      "no_of_replies": 0,  
      "completed": 1,  
      "time_spent": 0,  
      "participated": 0,  
      "pending": 67  
    },  
    {  
      "assigned": 0,  
      "staff_id": 4,  
      "name": "Agent 2",  
      "private_notes": 0,  
      "no_of_replies": 0,  
      "completed": 0,  
      "time_spent": 0,  
      "participated": 0,  
      "pending": 0  
    }  
  ],  
  "last_index": 13,  
  "sort_key": "name",  
  "page_count": 2,  
  "end_index": 10,  
  "sort_order": "a",  
  "start_index": 1
```

```
}
```

## 7. View Contact Activity

To view the contact activity listed by contact names, of a report.

**API Endpoint:** /api/1.1/json/report/<id of report>/customeractivity/

**Method:** GET

**Example Response:**

```
{
  "last_index": 36,
  "customer_activity": [
    {
      "no_of_tickets": 2,
      "pending_tickets": 1,
      "name": "John doe",
      "no_of_replies": 0,
      "contact_id": 3,
      "time_spent": 0,
      "completed_tickets": 1,
      "email": "john@example.com"
    },
    {
      "no_of_tickets": 2,
      "pending_tickets": 2,
      "name": "james may",
      "no_of_replies": 0,
      "contact_id": 30,
      "time_spent": 0,
      "completed_tickets": 0,
      "email": "james@example.com"
    }
  ],
  "sort_key": "name",
  "page_count": 4,
  "end_index": 10,
  "sort_order": "a",
  "start_index": 1
}
```

## 8. View SLA Performance

**API Endpoint:** /api/1.1/json/report/<id of report>/slaentries/

**Method:** GET

**Example Response:**

```
[
  {
    "target": 100,
    "ticketsChecked": 478,
    "ticketsBreached": 464,
    "name": "Sla 1",
    "achieved": 2.93
  },
  {
    "target": 100,
    "ticketsChecked": 9,
    "ticketsBreached": 8,
    "name": "SLA 2",
    "achieved": 11.11
  },
  {
    "target": 90,
    "ticketsChecked": 1,
    "ticketsBreached": 0,
    "name": "Sla 3",
    "achieved": 100.0
  }
]
```

9. Runtime filtering and sorting of reports.

Filters can be applied to the following report views.

- Tabular data
- Response stats
- Staff performance
- Contact performance
- Staff activity
- Contact activity
- SLA performance

## **Filter Parameters**

period\_type - Filter tickets by time of

- cr - Ticket Creation
- as - Ticket Assignment
- str - Staff Reply
- cur - Customer Reply
- prs - Staff Private Notes
- srp - Staff Reply & Private Notes
- cl - Ticket Closure

period\_date\_range\_type - Filter tickets for the period

- sr - selected range

1. period\_start - start date (yyyy-mm-dd)

2. period\_end - end date (yyyy-mm-dd)

- tod - today
- l7d - last 7 days
- mtd - month to date
- ytd - year to date



- pm - previous month

## Examples:

Specific date range:

[https://customersupport.happyfox.com/api/1.1/json/report/3/tabulardata/?](https://customersupport.happyfox.com/api/1.1/json/report/3/tabulardata/?period_start=2019-01-09&period_end=2019-08-05&period_type=cr&period_date_range_type=sr)

[period\\_start=2019-01-09&period\\_end=2019-08-05&period\\_type=cr&period\\_date\\_range\\_type=sr](https://customersupport.happyfox.com/api/1.1/json/report/3/tabulardata/?period_start=2019-01-09&period_end=2019-08-05&period_type=cr&period_date_range_type=sr)

([https://acmewidgetsco.happyfox.com/api/1.1/json/report/3/tabularview/?](https://acmewidgetsco.happyfox.com/api/1.1/json/report/3/tabularview/?period_start=2019-01-09&period_end=2019-08-05&period_type=cr&period_date_range_type=sr)

[period\\_start=2019-01-09&period\\_end=2019-08-05&period\\_type=cr&period\\_date\\_range\\_type=sr](https://acmewidgetsco.happyfox.com/api/1.1/json/report/3/tabularview/?period_start=2019-01-09&period_end=2019-08-05&period_type=cr&period_date_range_type=sr))

Year to date: [https://customersupport.happyfox.com/api/1.1/json/report/3/tabulardata/?](https://customersupport.happyfox.com/api/1.1/json/report/3/tabulardata/?period_type=cr&period_date_range_type=ytd)

[period\\_type=cr&period\\_date\\_range\\_type=ytd](https://customersupport.happyfox.com/api/1.1/json/report/3/tabulardata/?period_type=cr&period_date_range_type=ytd) ([https://customersupport.happyfox.com/api/1.1/](https://customersupport.happyfox.com/api/1.1/json/report/3/tabularview/?period_type=cr&period_date_range_type=ytd)

[json/report/3/tabularview/?period\\_type=cr&period\\_date\\_range\\_type=ytd](https://customersupport.happyfox.com/api/1.1/json/report/3/tabularview/?period_type=cr&period_date_range_type=ytd))

## Sort Parameters

The results fetched using the API endpoints can be sorted based on the requirement using the below sort options.

-- sort\_key - the key by which to sort (available list of sort keys is shown below)

-- sort\_dir - a (ascending, default) / d (descending)

## Available list of sort keys

-- Tabular view

- ticket - sort by ticket id
- status - sort by status order
- created - sort by ticket creation date
- duedate (default) - sort by ticket due date
- assigned - sort by assignee name of ticket

-- Staff activity, Staff performance views

- name - sort by agent's name

-- Contact activity view

- name - sort by contact's name