

Rocket.Chat Developer

Rocket.Chat Developer

These developer guides and APIs will help you start developing on Rocket.Chat quickly. To learn more about the Rocket.Chat platform, please consult our [documentation](#).

→ [Rocket.Chat Developer](#)

/

Guides

Developer Guides

→ **Contributors instant start on Gitpod**

/guides/developer/contributors-instant-start-on-gitpod

→ **Quick Start on Linux**

/guides/developer/quick-start-on-linux

→ **Mobile apps**

/guides/developer/mobile-apps

→ **Developing on Windows 10**

/guides/developer/developing-on-windows-10

Contributors instant start on Gitpod

If you are contributing to Rocket.Chat and will likely be working less than 50 hours a month on it, the quickest way to get started is via Gitpod.

Rocket.Chat Everywhere Development Environment on Gitpod

Gitpod runs a shared environment SaaS for developers working on open source projects. Please be respectful of other developers' need and support Gitpod on their commercial side if you are able to.

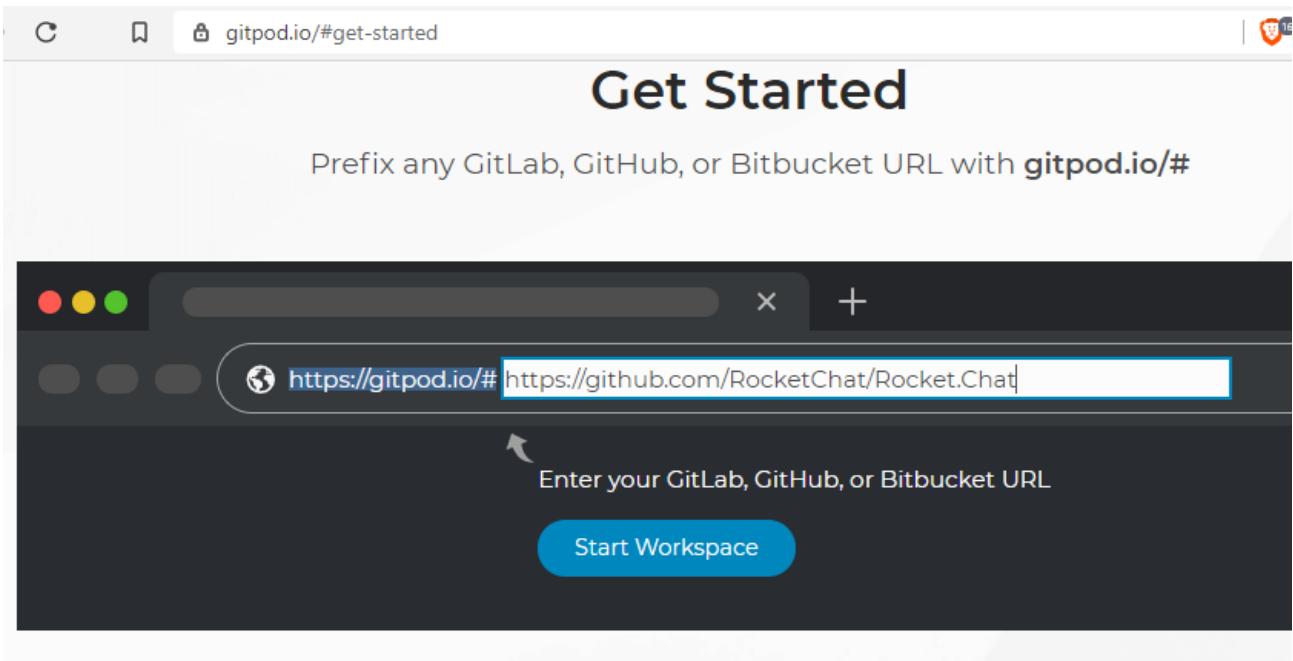
Gitpod hosts the entire development environment and you will be able to contribute to Rocket.Chat wherever and whenever you have access to a browser; even from Internet Cafes and Chromebooks.

Step by step

Go to <https://gitpod.io> and enter the Rocket.Chat Github project URL

<https://github.com/RocketChat/Rocket.Chat>

(you can also supply your Github fork of Rocket.Chat here)



Start your workspace, linking it to your Github account if necessary.

A workspace will be created and loaded with familiar Visual Studio Code environment.

Next, start a terminal in your workspace. And install meteor.

```
gitpod /workspace/Rocket.Chat $ curl https://install.meteor.com/ | sh
```

There will be warning messages, you do not have and do not need sudo (root access) on Gitpod. Next, add the newly installed meteor to your path.

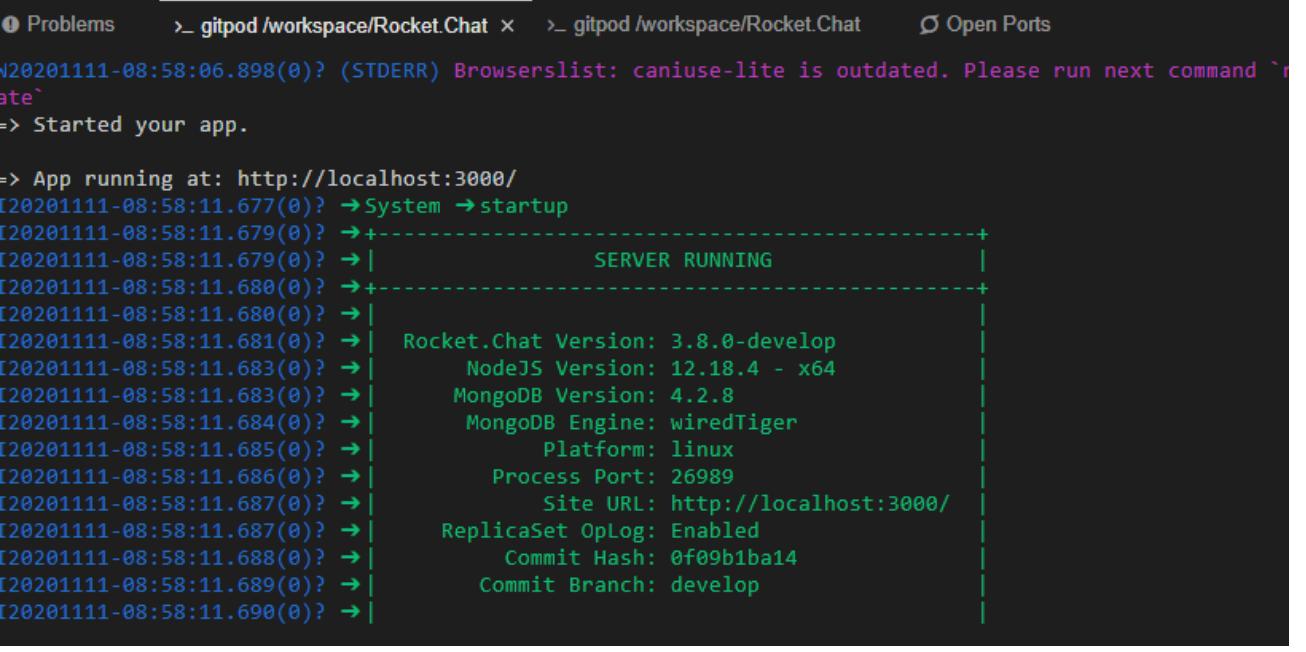
```
gitpod /workspace/Rocket.Chat $ PATH=$PATH:/home/gitpod/.meteor
```

Finally, install the node dependencies and start your server.

```
gitpod /workspace/Rocket.Chat $ meteor npm i
```

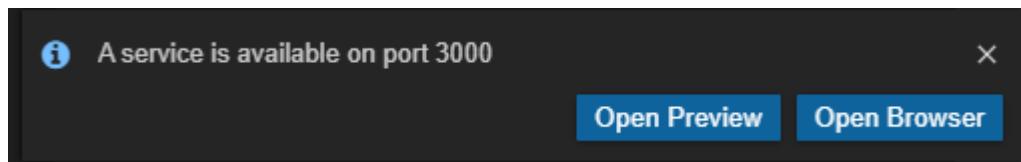
```
gitpod /workspace/Rocket.Chat $ meteor npm start
```

After a few minutes, your development environment should be up and running.



```
Problems gitpod /workspace/Rocket.Chat gitpod /workspace/Rocket.Chat Open Ports
2020-11-11T08:58:06.898(0)? (STDERR) Browserslist: caniuse-lite is outdated. Please run next command `update`  
=> Started your app.  
  
=> App running at: http://localhost:3000/  
[2020-11-11T08:58:11.677(0)? → System → startup  
[2020-11-11T08:58:11.679(0)? → +-----+  
[2020-11-11T08:58:11.679(0)? → | SERVER RUNNING |  
[2020-11-11T08:58:11.680(0)? → +-----+  
[2020-11-11T08:58:11.680(0)? → |  
[2020-11-11T08:58:11.681(0)? → | Rocket.Chat Version: 3.8.0-develop  
[2020-11-11T08:58:11.683(0)? → | NodeJS Version: 12.18.4 - x64  
[2020-11-11T08:58:11.683(0)? → | MongoDB Version: 4.2.8  
[2020-11-11T08:58:11.684(0)? → | MongoDB Engine: wiredTiger  
[2020-11-11T08:58:11.685(0)? → | Platform: linux  
[2020-11-11T08:58:11.686(0)? → | Process Port: 26989  
[2020-11-11T08:58:11.687(0)? → | Site URL: http://localhost:3000/  
[2020-11-11T08:58:11.687(0)? → | ReplicaSet OpLog: Enabled  
[2020-11-11T08:58:11.688(0)? → | Commit Hash: 0f09b1ba14  
[2020-11-11T08:58:11.689(0)? → | Commit Branch: develop  
[2020-11-11T08:58:11.690(0)? → |
```

Once the server starts, you will see a popup window indicating that your server is running on port 3000. Click the **Open Browser** button to view and interact with your Rocket.Chat server instance.



You can modify the code in Visual Studio Code and see the changes immediately on the server instance thanks to hot code reload.

Start contributing to Rocket.Chat!

Quick Start on Linux

You can run Rocket.Chat for development on a Linux machine or VM. The following instruction has been tested on a new Ubuntu 18.04 LTS installation. Try to find and use a *NEW Ubuntu server installation* with no other un-necessary software installed, not a "Desktop" or "Client" installation.

DO NOT use a system where you already have nodeJS installed to avoid problems.

During the build, memory usage will be nearly 8G, this is the minimum level of RAM recommended for development workstations. (If you are not doing any development, and just deploying a Rocket.Chat server - the RAM required can be as low as 1G.)

IMPORTANT: Note that there is no need to install mongo, nodejs, or npm on the base operating system. If you have any of these already installed; start over, or use another CLEAN system.

Development should be performed under a regular user account (not `root`) on Linux. There should be no need to run `sudo` at all. Running `sudo`, even once, during the installation – may mess up file permissions in irreversible manner.

You may notice build WARNINGS related to *peer dependencies* or other transitive dependencies. They are typically safe to ignore unless you are coding the feature or modules that require them.

1. Install tools required

```
sudo apt install g++ build-essential git curl python-minimal
```

(If you are using other OS (like Ubuntu 20.04 etc) you may run into an error "python-minimal has no installation candidate") To resolve this you can go to [Python Minimal error](#)

2. Install meteor

```
curl https://install.meteor.com/ | sh
```

(Under some circumstances, you may need to install a specific (older) release of Meteor instead of the latest, always check the `.meteor/release` file of the Github code repository to determine if you need to do this before you install meteor)

There is no need to install `node` or `npm`, as meteor already includes them. Verify by:

```
meteor node -v
```

```
meteor npm -v
```

3. Get rocket.chat code

```
git clone https://github.com/RocketChat/Rocket.Chat.git
```

(you may want to fork the code on Github first, and then clone your fork)

4. Install modules

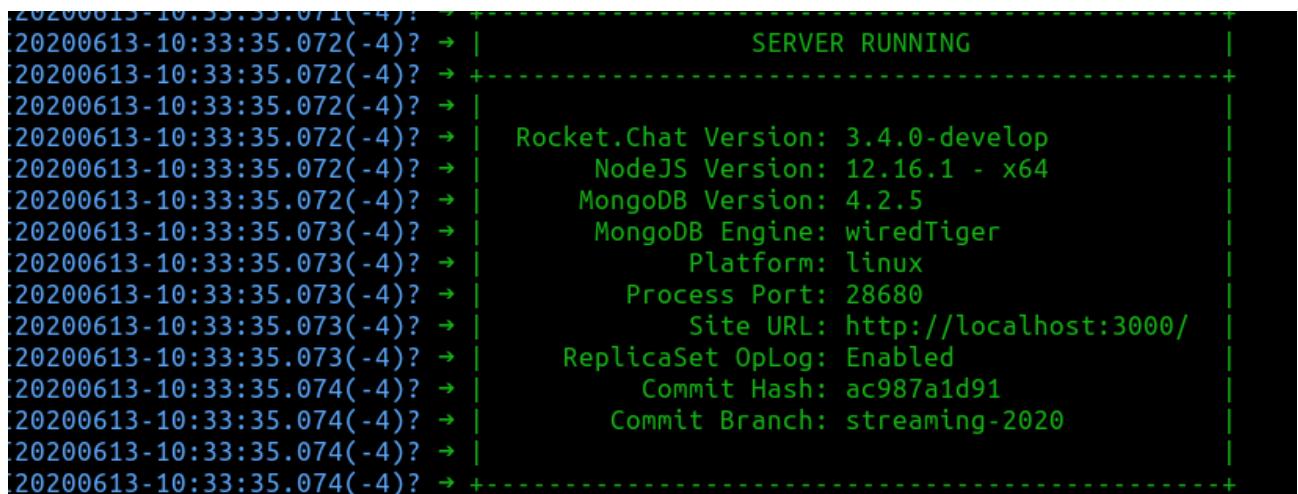
```
cd Rocket.Chat
```

```
meteor npm install
```

5. Start building (the first build can *take 10 or more minutes*, and you may see various warnings or minor errors – please be patient; subsequent dev builds after the first will be 5 minutes or less)

```
meteor npm start
```

When the server is ready, you will see a box with "Server Running" title:



```
20200613-10:33:35.071(-4)? → | SERVER RUNNING
20200613-10:33:35.072(-4)? → | +-----+
20200613-10:33:35.072(-4)? → | | Rocket.Chat Version: 3.4.0-develop
20200613-10:33:35.072(-4)? → | | NodeJS Version: 12.16.1 - x64
20200613-10:33:35.072(-4)? → | | MongoDB Version: 4.2.5
20200613-10:33:35.072(-4)? → | | MongoDB Engine: wiredTiger
20200613-10:33:35.072(-4)? → | | Platform: linux
20200613-10:33:35.072(-4)? → | | Process Port: 28680
20200613-10:33:35.072(-4)? → | | Site URL: http://localhost:3000/
20200613-10:33:35.072(-4)? → | | ReplicaSet OpLog: Enabled
20200613-10:33:35.072(-4)? → | | Commit Hash: ac987a1d91
20200613-10:33:35.072(-4)? → | | Commit Branch: streaming-2020
20200613-10:33:35.072(-4)? → | +-----+
20200613-10:33:35.073(-4)? → | SERVER RUNNING
20200613-10:33:35.073(-4)? → | +-----+
20200613-10:33:35.073(-4)? → | | Rocket.Chat Version: 3.4.0-develop
20200613-10:33:35.073(-4)? → | | NodeJS Version: 12.16.1 - x64
20200613-10:33:35.073(-4)? → | | MongoDB Version: 4.2.5
20200613-10:33:35.073(-4)? → | | MongoDB Engine: wiredTiger
20200613-10:33:35.073(-4)? → | | Platform: linux
20200613-10:33:35.073(-4)? → | | Process Port: 28680
20200613-10:33:35.073(-4)? → | | Site URL: http://localhost:3000/
20200613-10:33:35.073(-4)? → | | ReplicaSet OpLog: Enabled
20200613-10:33:35.073(-4)? → | | Commit Hash: ac987a1d91
20200613-10:33:35.073(-4)? → | | Commit Branch: streaming-2020
20200613-10:33:35.073(-4)? → | +-----+
20200613-10:33:35.074(-4)? → | SERVER RUNNING
20200613-10:33:35.074(-4)? → | +-----+
20200613-10:33:35.074(-4)? → | | Rocket.Chat Version: 3.4.0-develop
20200613-10:33:35.074(-4)? → | | NodeJS Version: 12.16.1 - x64
20200613-10:33:35.074(-4)? → | | MongoDB Version: 4.2.5
20200613-10:33:35.074(-4)? → | | MongoDB Engine: wiredTiger
20200613-10:33:35.074(-4)? → | | Platform: linux
20200613-10:33:35.074(-4)? → | | Process Port: 28680
20200613-10:33:35.074(-4)? → | | Site URL: http://localhost:3000/
20200613-10:33:35.074(-4)? → | | ReplicaSet OpLog: Enabled
20200613-10:33:35.074(-4)? → | | Commit Hash: ac987a1d91
20200613-10:33:35.074(-4)? → | | Commit Branch: streaming-2020
20200613-10:33:35.074(-4)? → | +-----+
```

This means that a Rocket.Chat server is running from your computer. To access the server, navigate to

```
http://localhost:3000
```

Other references:

- [Git](#)
- [Meteor](#)

Editing Rocket.Chat Files

Editing files is relatively simple. After you run `git clone`, the files from the repository are saved on your computer. You can go to the cloned repository folder and edit or add files to Rocket.Chat. When you make changes to Rocket.Chat the server will automatically rebuild.

Sometimes changes can shut down the server, if that happens just run `meteor npm start` again.

The Rocket.Chat code base is very large. You may need to increase this [system parameter](#) on your operating system for the files-change watcher to operate efficiently.

On Windows

Using Windows to develop is not recommended at this time, instead is better to create a Linux virtual machine and follow the steps mentioned above, but if for some reason you really need to build Rocket.Chat on Windows you can find a community supported guide over [here](#).

We look forward to the official release of Windows Subsystem for Linux 2 (WSL 2) when Rocket.Chat development on Linux may finally become viable.

See Also

- [Supporting SSL for Mobile Apps](#)
- [Development Troubleshooting](#)
- [Deployment Methods](#)

Developing on Windows 10

Microsoft finally released Windows Subsystem for Linux 2 ([WSL 2](#)) in June of 2020. Before this time, development of large and complex NodeJS based servers/full-stack applications such as Rocket.Chat on Windows is close to impossible.

WSL 2 is a complete architectural overhaul of Linux on Windows, installing a full genuine Linux kernel (built by Microsoft) alongside the classic Windows kernel. The Linux kernel and Windows kernel can now share system resources, such as memory and CPU, at a granularity not previously possible. It also includes major performance optimization on cross-subsystems file sharing, boot, and other developer-relevant areas.

You must be using **Windows 10, version 2004 or later** to take advantage of WSL 2, and to setup Rocket.Chat development.

Info	
PID	457
Running Instances	1
OpLog	Enabled
Commit	
Hash	793e15d4a9f20c98079085df0dd356cd9c18bd3
Date	Mon Jul 13 18:52:41 2020 -0300
Branch	develop
Tag	3.4.0
Author	Guilherme Gazzo
Subject	[IMPROVE] Message action styles (#18190)
Runtime Environment	
OS Type	Linux
OS Platform	linux
OS Arch	x64
OS Release	4.19.104-microsoft-standard
Node Version	v12.16.1
Mongo Version	4.2.5
Mongo Storage Engine	wiredTiger
OS Uptime	2 hours, 6 minutes, 4 seconds
OS Load Average	1.70, 0.97, 0.45
OS Total Memory	12.41 GB
OS Free Memory	6.07 GB
OS CPU Count	4
Build Environment	
OS Platform	linux
OS Arch	x64
OS Release	4.19.104-microsoft-standard
Node Version	v12.16.1
Date	July 14, 2020 5:55 AM

Before you start

The following are prerequisites for developing Rocket.Chat on Windows 10:

1. Make sure you have **Windows 10, version 2004** or later
2. Install and configure **WSL 2** by following [Microsoft documentation](#), making sure to select **Ubuntu 20.04 LTS** distribution as your choice of Linux
3. Download and install the latest [Linux Kernel Updates](#)

Machine requirement

Building Rocket.Chat code requires a minimum of 8 GB of RAM memory on the Linux subsystem. (If you are not doing any development, and just deploying a Rocket.Chat server - the RAM required can be as low as 1GB.) On version 2004, about 4 GB of RAM appears to be reserved for the Windows subsystem. You will need a Windows machine with the following minimal requirement to develop Rocket.Chat:

- 12 GB of RAM memory (16+ GB highly recommended)
 - 4 or more cores on CPU (at least 3 GHz boosted, 4.2 GHz or higher recommended)
 - 80 GB of available fast SSD storage (PCIe 4.0 NVMe SSD recommended)
-

Setting up development environment

IMPORTANT: Note that there is no need to install mongo, nodejs, or npm separately

NOTE: Development should be performed under a regular user account, not Administrator.

NOTE: During build, you may notice *WARNING* related to *peer dependencies* or other transitive dependencies. They are typically safe to ignore unless you are coding the feature or modules that require them.

1. Open a **WSL 2 shell** (not Powershell). Update Linux

```
sudo apt-get update sudo apt-get dist-upgrade
```

2. Install tools required

```
sudo apt-get install build-essential git curl python2-minimal pkg-config
```

3. Install meteor

```
curl https://install.meteor.com/ | sh
```

(Under some circumstances, you may need to install a specific (older) release of Meteor instead of the latest, always check the `.meteor/release` file of the Github code repository to determine if you need to do this before you install meteor)

There is no need to install `node` or `npm`, as meteor already includes them. Verify by:

```
meteor node -v
```

```
meteor npm -v
```

4. Make sure you are on the WSL 2 filesystem, `pwd` should return `/home/yourusername`

Get rocket.chat code:

```
git clone https://github.com/RocketChat/Rocket.Chat.git
```

(you may want to fork the code on Github first, and then clone your fork)

5. Install modules

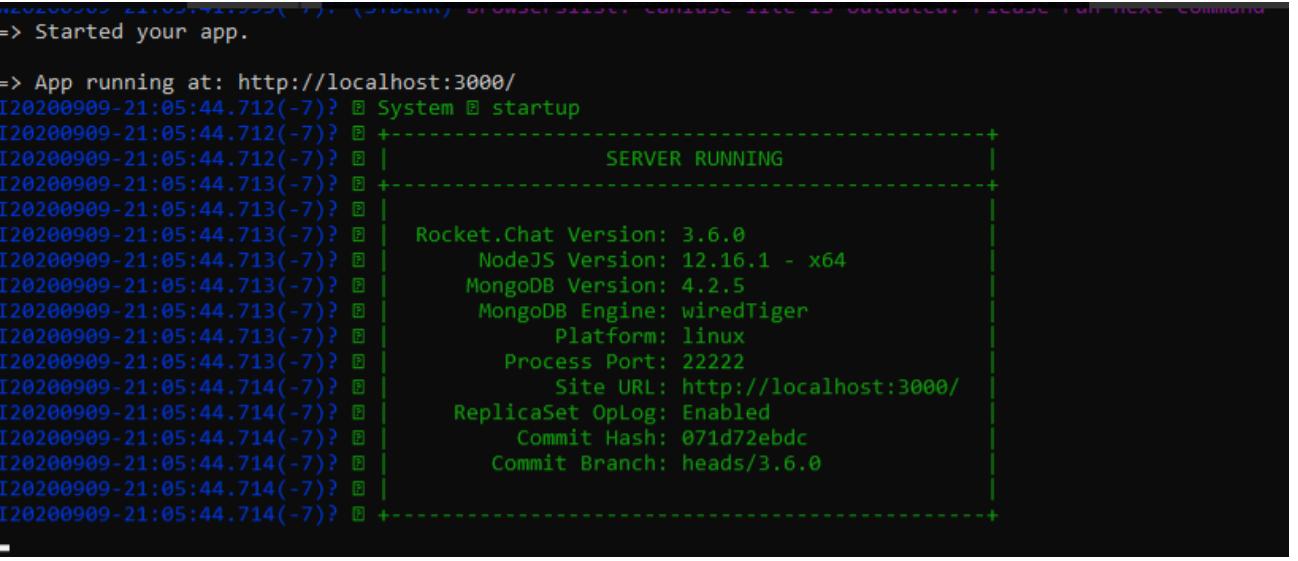
```
cd Rocket.Chat
```

```
meteor npm install
```

6. Start building (the first build can *take 10 or more minutes*, and you may see various warnings or minor errors – please be patient; subsequent dev builds after the first will be 5 minutes or less)

```
meteor npm start
```

When the server is ready, you will see a box with "Server Running" title:



```
[20200909-21:05:44.703(-7)]: (SYNCHRONOUS) BROWSER-SILENT. Continue life is outdated. Please run next command
=> Started your app.

=> App running at: http://localhost:3000/
[20200909-21:05:44.712(-7)?] System startup
[20200909-21:05:44.712(-7)?] +-----+
[20200909-21:05:44.712(-7)?] | SERVER RUNNING |
[20200909-21:05:44.713(-7)?] +-----+
[20200909-21:05:44.713(-7)?] |
[20200909-21:05:44.713(-7)?] | Rocket.Chat Version: 3.6.0
[20200909-21:05:44.713(-7)?] | NodeJS Version: 12.16.1 - x64
[20200909-21:05:44.713(-7)?] | MongoDB Version: 4.2.5
[20200909-21:05:44.713(-7)?] | MongoDB Engine: wiredTiger
[20200909-21:05:44.713(-7)?] | Platform: linux
[20200909-21:05:44.713(-7)?] | Process Port: 22222
[20200909-21:05:44.714(-7)?] | Site URL: http://localhost:3000/
[20200909-21:05:44.714(-7)?] | ReplicaSet OpLog: Enabled
[20200909-21:05:44.714(-7)?] | Commit Hash: 071d72ebdc
[20200909-21:05:44.714(-7)?] | Commit Branch: heads/3.6.0
[20200909-21:05:44.714(-7)?] +-----+
```

This means that a Rocket.Chat server is running from your computer. To access the server, use Chrome, Brave, or Firefox browser, and navigate to

```
http://localhost:3000
```

Other references:

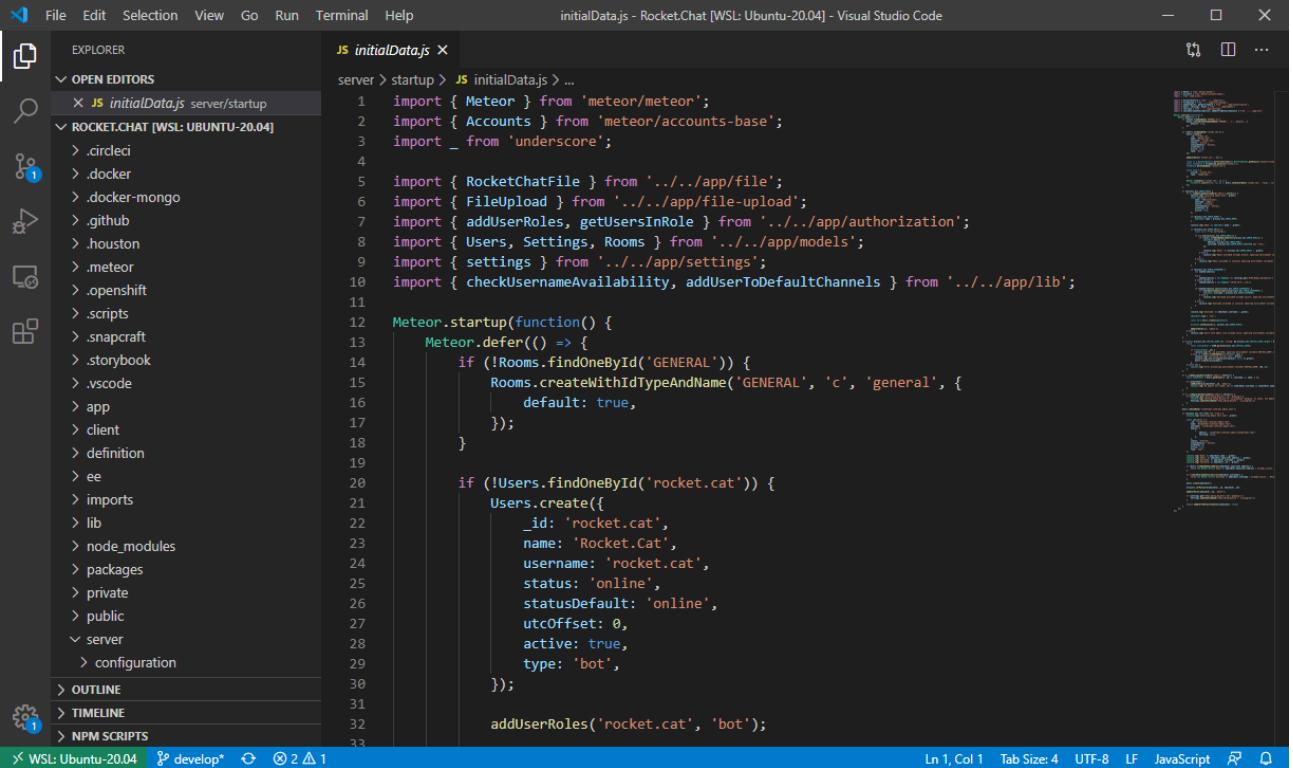
- [Git](#)
- [Meteor](#)

Editing Rocket.Chat Files

On Windows 10, the best coding environment to use is Visual Studio Code. Install [Visual Studio Code](#) from Windows. Make sure you also install the VS Code extension named [Remote - WSL](#). Optionally install the [Windows Terminal](#) extension

Editing files is relatively simple. After you run `git clone`, the files from the repository are saved on your computer. You can go to the cloned repository folder and edit or add files to Rocket.Chat. From a WSL shell, you can start Visual Studio for Code using command

```
code . .
```



```
JS initialData.js x
server > startup > JS initialData.js > ...
1 import { Meteor } from 'meteor/meteor';
2 import { Accounts } from 'meteor/accounts-base';
3 import _ from 'underscore';
4
5 import { RocketChatFile } from '../app/file';
6 import { FileUpload } from '../app/file-upload';
7 import { addUserRoles, getUsersInRole } from '../app/authorization';
8 import { Users, Settings, Rooms } from '../app/models';
9 import { settings } from '../app/settings';
10 import { checkUsernameAvailability, addUserToDefaultChannels } from '../app/lib';

11 Meteor.startup(function() {
12   Meteor.defer(() => {
13     if (!Rooms.findOneById('GENERAL')) {
14       Rooms.createWithIdTypeAndName('GENERAL', 'c', 'general', {
15         default: true,
16       });
17     }
18   })
19
20   if (!Users.findOneById('rocket.cat')) {
21     Users.create({
22       _id: 'rocket.cat',
23       name: 'Rocket.Cat',
24       username: 'rocket.cat',
25       status: 'online',
26       statusDefault: 'online',
27       utcOffset: 0,
28       active: true,
29       type: 'bot',
30     });
31     addUserRoles('rocket.cat', 'bot');
32   }
33 });

Ln 1, Col 1 Tab Size: 4 UTF-8 LF JavaScript ⚙️ 🔍
```

When you make changes to Rocket.Chat the server will automatically rebuild.

Sometimes changes can shut down the server, if that happens just run `meteor npm start` again.

Development Workflow

1. Start working on an issue you're assigned to. If you're not assigned to any issue, find the issue with the highest priority you can work on, by relevant label and assign it to yourself. Priority is given by milestones. You should always check issues in the current milestone, then short-term, middle-term and long-term in that order.
 2. You are responsible for the issue that you're assigned to. If you are not able to do it or you believe you won't make it to the selected milestone, talk to your team, lead or manager to have it reassigned or postponed.
 3. Once your code has been deployed in the release-candidate environment, please verify that your changes work as intended. We have seen issues where bugs did not appear in development but showed in production (e.g. due to merge issues).
-

Choosing something to work on

Start working on things with the highest priority in the current milestone. The priority of items are defined under labels in the repository, but you are able to sort by priority.

After sorting by priority, choose something that you're able to tackle and falls under your responsibility. Take a good look on our [labels](#) and see which ones apply for your position.

To filter very precisely, you could filter all issues for:

- Milestone: Whichever is the current version
- Assignee: Unassigned
- Label: Your label of choice. For instance `feat: api` , `feat: integration / plugin` ,
`feat: webrtc` , `subj: ui/ux` , `subj: security`
- Sort by priority

If you're in doubt about what to work on, ask your team, lead or manager. They will be able to tell you.

Creating a Pull Request

Anytime you start to work on something different, make sure you create or switch to a branch specific to the feature you're working on. You can choose to create a Pull Request anytime during your development phase, just make sure you add the label

`stat: in progress` while the PR is not ready for merge (and remember to remove the label when it is). When naming your Pull Request, please start the name with one of the following tags for identifying changes:

- [NEW] for new features (eg.: `[NEW] WhiteBoard integration`)
- [FIX] for bug fixes. You should include the issue number(s) in parenthesis, whenever possible. (eg.: `[FIX] OTR timeout problems (#629, #2535)`)
- [BREAK] for giving proper attention to changes that will break previous versions of Rocket.Chat (eg.:
`[BREAK] Change notification setting type from boolean to string`)

Getting your pull request reviewed, approved, and merged

There are a few rules to get your pull request accepted:

1. All checks have passed
2. Travis CI runs automatically when you push your pull request. If Travis fails, take a look at the reasons for failing. If it fails for no apparent reason, try running it again.
3. You must sign the [Contributor License Agreement \(CLA\)](#)
4. At least one team member must approve the Pull Request. If you don't know who to ask for an approval, let your team, lead or manager know you need one, and someone will be assigned as reviewer.

Finishing a release branch

When the state of the `release-candidate` branch is ready to become a real release, some actions need to be carried out. First, the release branch is merged into `master` (since every

commit on `master` is a new release by definition). Next, that commit on `master` must be tagged for easy future reference to this historical version. Finally, the changes made on the `release-candidate` branch need to be merged back into `develop`, so that future releases also contain these bug fixes.

The release process is as follows:

```
1 $ git checkout master
2 Switched to branch 'master'
3 $ git merge --no-ff release-candidate
4 Merge made by recursive.
5 (Summary of changes)
6 $ git tag -a [version number]
```

To keep the changes made in the release branch, we need to merge those back into `develop`, though. In Git:

```
1 $ git checkout develop
2 Switched to branch 'develop'
3 $ git merge --no-ff release-candidate
4 Merge made by recursive.
5 (Summary of changes)
```

To create the new `release-candidate` based on `develop`:

```
1 $ git checkout release-candidate
2 Switched to branch 'release-candidate'
3 $ git merge --no-ff develop
4 Merge made by recursive.
5 (Summary of changes)
```

Every team member should strive to thoroughly test the `release-candidate` branch, especially regarding issues they've worked on. Any bug-fixes applied to `release-candidate` should immediately be cherry-picked into `develop`.

You may read all about our branching model on [A successful Git branching model](#), by [Vincent Driessen](#)

Hotfixes

Hotfix branches are very much like release branches in that they are also meant to prepare for a new production release, albeit unplanned. They arise from the necessity to act immediately upon an undesired state of a live production version. When a critical bug in a production version must be resolved immediately, a hotfix branch may be branched off from the corresponding tag on the master branch that marks the production version.

Hotfix branches are created from the `master` branch:

```
1 $ git checkout -b hotfix-[current-version].[sub-version] master
2 Switched to a new branch "hotfix-X.X.X"
```

Then, fix the bug and commit the fix. Later, merge the bugfix into `master` and also into `release-candidate` (and `develop` if the bugfix cannot wait for the release branch to be finished).

```
1 $ git commit -m "Fixed severe production problem"
2 [hotfix-X.X.X abc12d3] Fixed severe production problem
3 5 files changed, 32 insertions(+), 17 deletions(-)
4
5 $ git checkout master
6 Switched to branch 'master'
7 $ git merge --no-ff hotfix-X.X.X
8 Merge made by recursive.
9 (Summary of changes)
10 $ git tag -a [current-version].[sub-version]
11
12 $ git checkout release-candidate
13 Switched to branch 'release-candidate'
14 $ git merge --no-ff hotfix-X.X.X
15 Merge made by recursive.
16 (Summary of changes)
```

Error reporting

We use [GitHub Issues](#) as our error reporting tool. When creating a new issue, always state the version you're working on (write the version number, not `latest`). If you don't know the version you're working on, access `/api/info` on your instance. Include as much detailed information as you can, such as animated gifs and/or screenshots of error, code samples and reproduction steps.

Labelling issues

To allow for asynchronous issue handling, we use milestones and labels. Leads and product managers handle most of the scheduling into milestones. Labelling is a task for everyone.

Most issues will have labels for at least one of the following:

- `Feat` (`feat: accessibility`, `feat: embed`, `feat: livechat`, etc.)
- `Subject` (`subj: mobile`, `subj: security`, `subj: ui/ux`, etc.)
- `Type` (`type: bug`, `type: new feature`, `type: discussion`, etc.)

All labels are listed on the [labels page](#).

Labels for community contributors

Issues that are beneficial to our users, 'nice to haves', that we currently do not have the capacity for or want to give the priority to, are labeled as `contrib:`, so the community can make a contribution. We categorize them into `easy`, `intermediate` and `experts needed` to help contributors pick an issue to work on when joining the project, based on how difficult is the work we believe the issue demands.

Code Styleguide

CSS Styleguide

- [Comments](#)
 - [Formatting](#)
 - [Selectors](#)
 - [Properties](#)
 - [Colors](#)
-

Comments

- Prefer line comments (`//`) to block comments.
 - Prefer comments on their own line. Avoid end-of-line comments.
 - Write detailed comments for code that isn't self-documenting:
 - Uses of z-index
 - Compatibility or browser-specific hacks
-

Formatting

- Use tabs for indentation
- Use dashes over camelCasing in class names.
- Prefer to not use ID selectors
- When using multiple selectors in a rule declaration, give each selector its own line.
- Put a space before the opening brace `{` in rule declarations
- In properties, put a space after, but not before, the `:` character.
- Put closing braces `}` of rule declarations on a new line
- Put blank lines between rule declarations, but not in nesting selectors.
- Put nesting after all properties.
- Try to not use `!important`.
- Lowercase, always `:`

Bad

```
1 blockquote {
2    .clearfix;
3    margin: .5em 0;
4    &:first-child {
5      margin-top: 0;
6    }
7    &:last-child {
8      margin-bottom: 0;
9    }
10   padding-left: 10px;
11   position: relative;
12   &:before {
13     content: ' ';
14     width: 4px;
15     position: absolute;
16     border-radius: 2px;
17     left: 0;
18     top: -1px;
19     bottom: -1px;
20   }
21 }
22 .login-terms {
23   font-size: smaller;
24   width: 520px;
25   padding: 10px;
26   max-width: 100%;
27   margin: auto;
28   a {
29     font-weight: bold !important;
30     text-decoration: underline;
31   }
32 }
```

Good

```
1 blockquote {
2    .clearfix;
3    margin: .5em 0;
4    padding-left: 10px;
5    position: relative;
6
7    &:first-child {
8      margin-top: 0;
9    }
10 }
```

```
11     &:last-child {
12         margin-bottom: 0;
13     }
14
15     &::before {
16         content: ' ';
17         width: 4px;
18         position: absolute;
19         border-radius: 2px;
20         left: 0;
21         top: -1px;
22         bottom: -1px;
23     }
24 }
25
26 .login-terms {
27     font-size: smaller;
28     width: 520px;
29     padding: 10px;
30     max-width: 100%;
31     margin: auto;
32
33     a {
34         font-weight: bold;
35         text-decoration: underline;
36     }
37 }
```

Selectors

Create an empty line before selectors

Bad

```
1 .class {
2     border: 0;
3 }
4 .second-class: {
5     text-align: right;
6 }
```

Good

```
1 .class {  
2     border: 0;  
3 }  
4  
5 .second-class: {  
6     text-align: right;  
7 }
```

Use a single space after selectors

Bad

```
1 .class{  
2     border: 0;  
3 }
```

Good

```
1 .class {  
2     border: 0;  
3 }
```

Add an empty line before nested selectors

Bad

```
1 .class {  
2     border: 0;  
3     &.another-class {  
4         color: #ffffff;  
5     }  
6 }
```

Good

```
1 .class {  
2     border: 0;  
3  
4     &.another-class {  
5         color: #ffffff;  
6     }  
7 }
```

Use an empty line before non-nested selectors

Bad

```
1 .class {  
2     border: 0;  
3  
4     &.another-class {  
5         color: #ffffff;  
6     }  
7 }  
8 .logo {  
9     display: block;  
10 }
```

Good

```
1 .class {  
2     border: 0;  
3  
4     &.another-class {  
5         color: #ffffff;  
6     }  
7 }  
8  
9 .logo {  
10    display: block;
```

```
11 }
```

Don't add whitespaces inside of the brackets within attribute selectors

Bad

```
1 .input[ type="text" ] {  
2     border: 0;  
3 }
```

Good

```
1 .input[type="text"] {  
2     border: 0;  
3 }
```

Don't add a whitespace between operators within attribute selectors

Bad

```
1 .input[type = "text" ] {  
2     border: 0;  
3 }
```

Good

```
1 .input[type="text"] {  
2     border: 0;  
3 }
```

Always use a single space between the combinator of selectors

Bad

```
1 .class>.button {  
2     font-size: 1rem;  
3 }
```

Good

```
1 .class > .button {  
2     font-size: 1rem;  
3 }
```

Add a newline after the commas of selector lists

Bad

```
1 .class, .another, .another-class {  
2     padding: 1.5rem;  
3 }
```

Good

```
1 .class,  
2 .another,  
3 .another-class {  
4     padding: 1.5rem;  
5 }
```

Don't add a whitespace inside of the parentheses within pseudo-class

selectors

Bad

```
1 .class:not( .another ) {  
2     margin: 5px;  
3 }
```

Good

```
1 .class:not(.another) {  
2     margin: 5px;  
3 }
```

Double colon for applicable pseudo-elements

Bad

```
1 .class:before {  
2     border-width: 2px;  
3 }  
4  
5 .class:first-child {  
6     color: #000000;  
7 }
```

Good

```
1 .class::before {  
2     border-width: 2px;  
3 }  
4  
5 .class:first-child {  
6     color: #000000;
```

```
7 }
```

Properties

Don't add empty line in first/last property or in between properties

Bad

```
1 .room-list {  
2  
3   border: 0;  
4   padding-left: 0;  
5  
6   color: #ddfc32;  
7  
8 }
```

Good

```
1 .room-list {  
2   border: 0;  
3   padding-left: 0;  
4   color: #ddfc32;  
5 }
```

Don't add properties in single-line. Use one property per line

Bad

```
.sumbit {color: #ffffff; background-color: #000000;}
```

Good

```
1 .sumbit {  
2     color: #ffffff;  
3     background-color: #000000;  
4 }
```

No empty blocks

Bad

```
.button {}
```

Good

```
1 .button {  
2     float: left;  
3 }
```

Longhand properties must be combined into one shorthand property

Bad

```
1 .class {  
2     padding-left: 12px;  
3     padding-top: 5px;  
4     padding-bottom: 8px;  
5 }
```

Good

```
1 .class {  
2     padding: 5px auto 8px 12px;  
3 }
```

Don't use shorthand properties that override related longhand properties

Example

```
1 .class {  
2     border-color: #ffffff;  
3     border: 1px solid #000000;  
4 }
```

Always add a trailing semicolon in the end of a declaration

Bad

```
1 .another-class {  
2     color: #ffffff;  
3     padding: 2px  
4 }
```

Good

```
1 .another-class {  
2     color: #ffffff;  
3     padding: 2px;  
4 }
```

Add a newline after the colon of declarations

Bad

```
1 .another-class {  
2     box-shadow: 0 0 0 1px #5b9dd9, 0 0 2px 1px rgba(30, 140, 190, 0.8);  
3 }
```

Good

```
1 .another-class {  
2     box-shadow:  
3         0 0 0 1px #5b9dd9,  
4         0 0 2px 1px rgba(30, 140, 190, 0.8);  
5 }
```

Don't duplicate properties within declaration blocks

Bad

```
1 .another-class {  
2     display: block;  
3     margin-top: 2rem;  
4     display: inline-block;  
5 }
```

Good

```
1 .another-class {  
2     margin-top: 2rem;  
3     display: inline-block;  
4 }
```

Don't duplicate selectors along the file

Bad

```
1 .some-class {  
2     display: block;  
3 }  
4 .another-class {  
5     position: absolute;  
6 }  
7 .some-class {  
8     margin-top: 2rem;  
9     display: inline-block;  
10 }
```

Good

```
1 .some-class {  
2     display: block;  
3     margin-top: 2rem;  
4     display: inline-block;  
5 }
```

Add a whitespace before bang (!) declaration

Bad

```
1 .class {  
2     margin-left: 12px!important;  
3 }
```

Good

```
1 .class {  
2     margin-left: 12px !important;  
3 }
```

Use spaces around calc operators

Bad

```
1 .class {  
2     width: calc(~"200px-1rem");  
3 }
```

Good

```
1 .class {  
2     width: calc(~"200px - 1rem");  
3 }
```

Remove units for zero lengths

Bad

```
1 .button {  
2     padding: 10px 0px 0px 2px;  
3 }
```

Good

```
1 .button {  
2     padding: 10px 0 0 2px;  
3 }
```

Use a leading zero for fractional numbers css than 1

Bad

```
1 .message-form {  
2     margin-right: .5rem;  
3 }
```

Good

```
1 .message-form {  
2     margin-right: 0.5rem;  
3 }
```

Don't add trailing zeros in numbers

Bad

```
1 .flex-tab {  
2     padding-bottom: 1.500rem;  
3 }
```

Good

```
1 .flex-tab {  
2     padding-bottom: 1.5rem;  
3 }
```

Don't add newlines in strings

Bad

```
1 .nav-link::before {  
2     content: "sample  
3         text";
```

```
4 }
```

Good

```
1 .nav-link::before {  
2   content: "sample text";  
3 }
```

Colors

Lowercase, always

Long notation

Bad

```
1 .another-class {  
2   background-color: #FFF;  
3 }
```

Good

```
1 .another-class {  
2   background-color: #ffffff;  
3 }
```

Internationalization

When developing in Rocket.Chat, you have the ability to add strings to our translations files in which later will be translated to other languages by our community translators.

Tools Used

For selecting strings for the correct language we use the [TAP:i18n](#) meteor package.

For managing contributions from the translators community we use [lingohub](#). If you are interested in contributing to Rocket.Chat translations please see [Translating](#)

Adding Strings to the Translation Files

To have a string translated you will firstly need a `key` that would be the identifier of the string you want translated, for example the string `This room is read only` should have the key as `room_is_read_only`. Please have in mind when naming a key that spaces should be replaced with underscores (`_`) and it should be named in english, as is the language selected for Rocket.Chat's code. Finally you will have the `key/value` pair will look like this: `"room_is_read_only": "This room is read only"`

You can also specify `placeholders`, that will allow you to change information on the string via parameters when calling the `i18n` method. A parameter will be surrounded by two double underscores (`__ __`), and it will look like this,

`"Conversation_closed": "Conversation closed: __comment__."` where `__comment__` can be replaced by any string provided in the parameters.

After that you will need to add this key to its respective `i18n.json` file under the `i18n` folder.

When your pull request is merged, our contributors at [LingoHub](#) will be notified of the new string and start translating it.

Using Translated Strings on the code

Now that you have added your strings into the translation files, is time to use them in the code!

When writing on a `.js` file you can use the global method `TAPi18n.__()`. Now depending where you are calling this method from it can have different parameters. If you call it from the back-end of Rocket.Chat you will have to specify a language (in which you can normally grab from an user's object). The method will look like this,

`TAPi18n.__('YOUR_KEY_HERE', {}, user.language)`. You can also pass parameters on the second argument to replace placeholders on the translated strings. Now if you are calling from the front end you can simply use `TAPi18n.__('YOUR_KEY_HERE', {})` that will translate the selected string to the user's current selected language.

If the selected key is not present in the respective `.i18n.json` file it will default it to english, if no key is found it will display the key inserted in the method as a string.

Now if you are in a `.html` file, you can simply surround the string with `{{ "{{_ }}}}` for example `{{ "{{_ YOUR_KEY_HERE" }}}}`. It will work the same as the method mentioned above, only with the convenience of adding it directly to the `.html` file.

Sometimes you can find some methods that requires an object with a `i18nLabel` or `i18nDescription`. In these cases you only need to insert the key of the string, the method will do the rest.

Mobile apps

→ **Whitelabeling mobile apps**

/guides/developer/mobile-apps/whitelabeling-mobile-apps

→ **Supporting SSL for development on Rocket.Chat**

/guides/developer/mobile-apps/supporting-ssl-for-development-on-rocket.chat

→ **Analytics & Data Usage**

/guides/developer/mobile-apps/analytics-and-data-usage

Whitelabeling mobile apps

In this guide we will cover how to rebrand Rocket.Chat Mobile Apps to suit your styling.

Here we will show you how to customize:

- The App Icons
 - Splash Screens
 - App Name
 - Colors
-

Important

- This document is updated after every release, so we can guarantee it's stable for production
 - `develop` branch might be different from this
 - Keep in mind that you will need an **intermediate** knowledge of Android/iOS development and basic Javascript knowledge
 - Our repo contains targets/build flavors to build both our Experimental and Official apps
 - Both apps are equal, but released at different pace on the stores
 - If you see an Experimental folder, don't be scared of breaking anything. It's just a folder containing the assets for the non-official app
-

Repo

- Make sure you have both iOS and Google developer accounts and the respective development environments working
 - You can follow this guide: <https://reactnative.dev/docs/getting-started>
 - Clone <https://github.com/RocketChat/Rocket.Chat.ReactNative>
 - Checkout `single-server` branch (git clone -b single-server <https://github.com/RocketChat/Rocket.Chat.ReactNative>)
-

General

- Create an account on <https://www.bugsnag.com/>
 - Set `BUGSNAG_API_KEY` on `config.js`
 - Set `server` , `appGroup` and `appStoreId` on `app.json`
 - `appGroup` must be the same App Group created for the iOS app
 - Change app colors on `colors.js`
-

Firebase

Creating a new project on Google Cloud Platform

- Visit <https://console.cloud.google.com/home/dashboard>
- By the text Google Cloud Platform there is a dropdown, open and then “Create project”
- In the dialog give an project name then “Create”
- Wait the creating process, you can follow in notifications by your avatar
- You will be redirected to the project page after creation

Creating new Firebase project

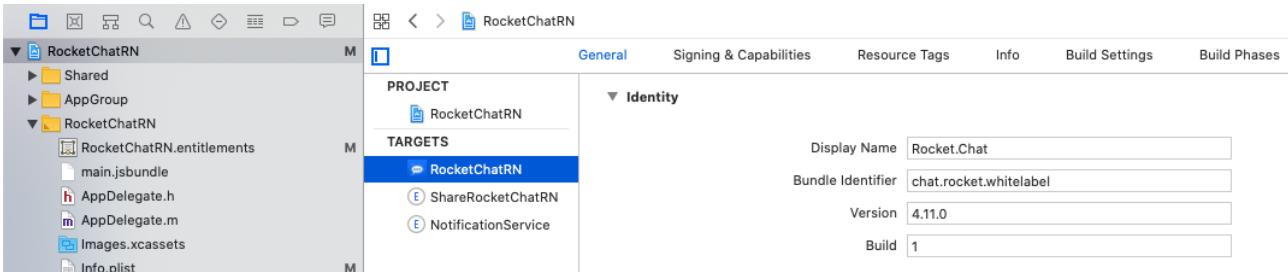
- Visit <https://console.firebaseio.google.com/>
 - Click on “Add Project”
 - Enter the project name you created on previous step
 - Follow the wizard until Firebase project is created
 - We’re going to create the apps later on the tutorial
-

iOS

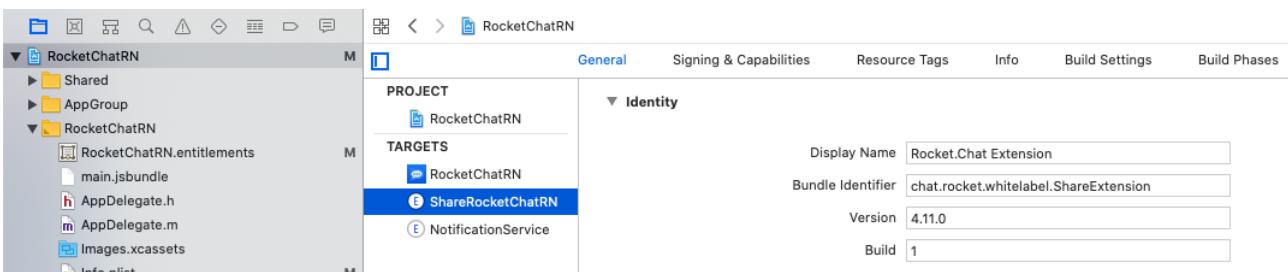
General setup

- Open `RocketChatRN.xcworkspace` on Xcode (11.7 or newer)

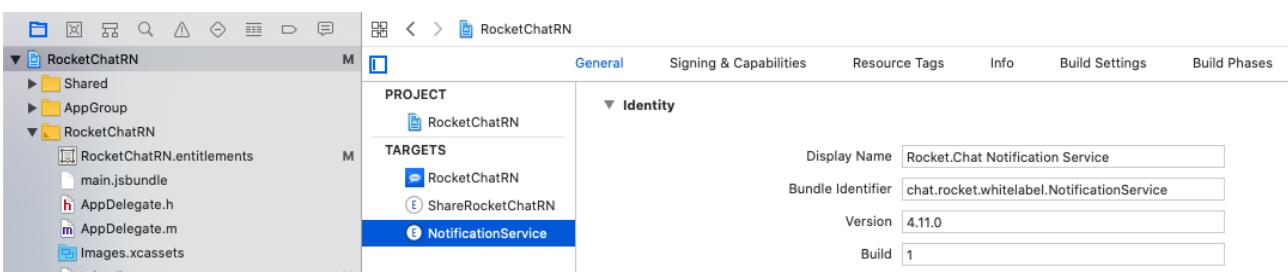
- On General tab, select “RocketChatRN” and change Display Name, Bundle Identifier, Version and Build
 - Note: as explained on Important section, we have two targets and we’re going to cover the default one on this doc, which is the Experimental app.



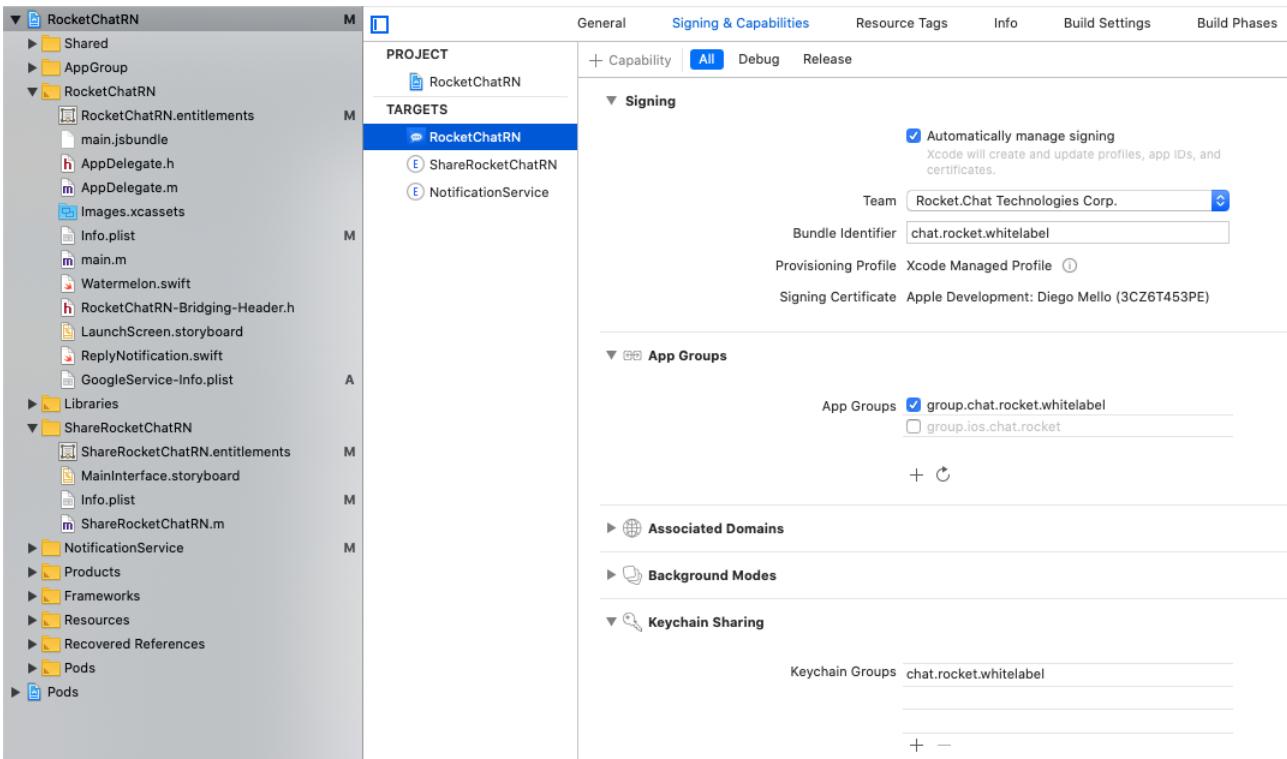
- Select “ShareRocketChatRN” and change the same properties
 - Display Name and Bundle Identifier are different from the previous target
 - Version and Build must be the same on all targets



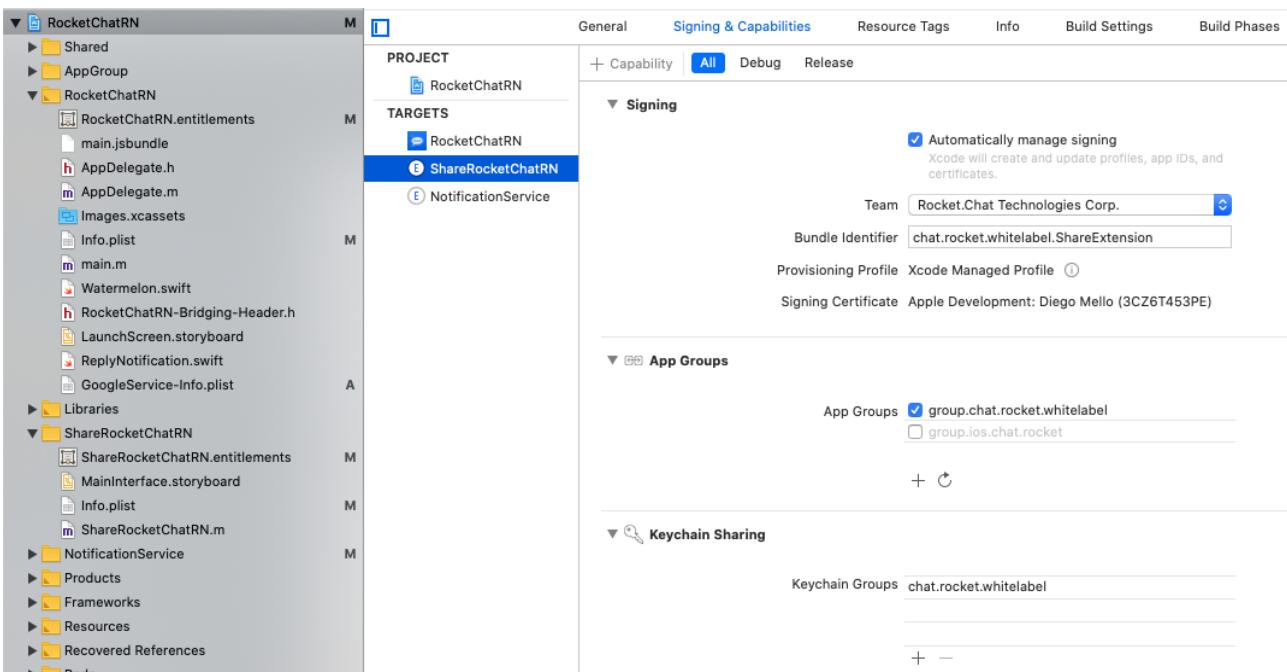
- Select “NotificationService” and change the same properties
 - Display Name and Bundle Identifier are different from the previous target
 - Version and Build must be the same on all targets



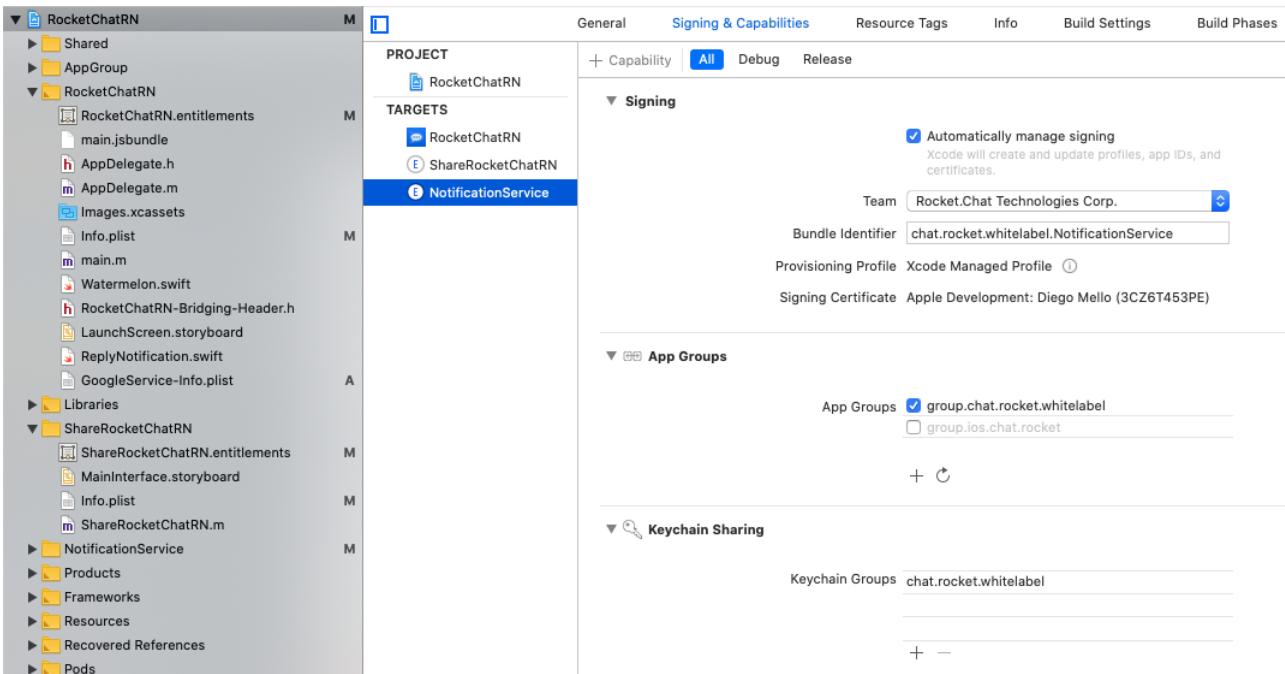
- On Signing and Capabilities, check “Automatically manage signing”, select your app group and add a keychain group



- Select “ShareRocketChatRN”, check “Automatically manage signing”, select your app group and add the same keychain group



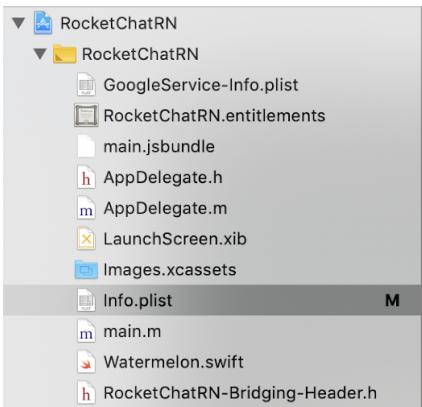
- Select “NotificationService”, check “Automatically manage signing”, select your app group and add the same keychain group



- Set the same app group on `RocketChatRN/Info.plist`,

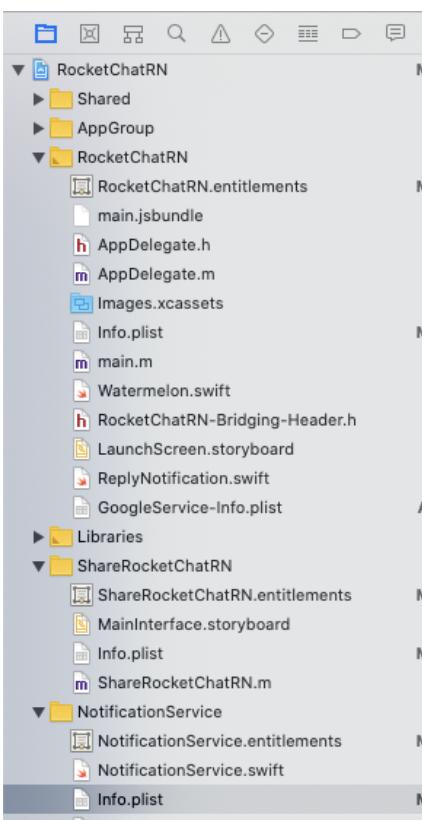
`ShareRocketChatRN/Info.plist` and `NotificationService/Info.plist`

Key	Type	Value
▼ Information Property List	Dictionary	(13 items)
AppGroup	String	group.chat.rocket.whitelabel
Localization native development re...	String	\$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	Rocket.Chat Experimental
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	4.5.1
Bundle version	String	1
► App Transport Security Settings	Dictionary	(1 item)
► NSExtension	Dictionary	(4 items)
► Fonts provided by application	Array	(1 item)



The screenshot shows the Xcode project structure for 'RocketChatRN'. The 'Info.plist' file is selected in the project navigator. The right-hand panel displays the contents of 'Info.plist' as a key-value table.

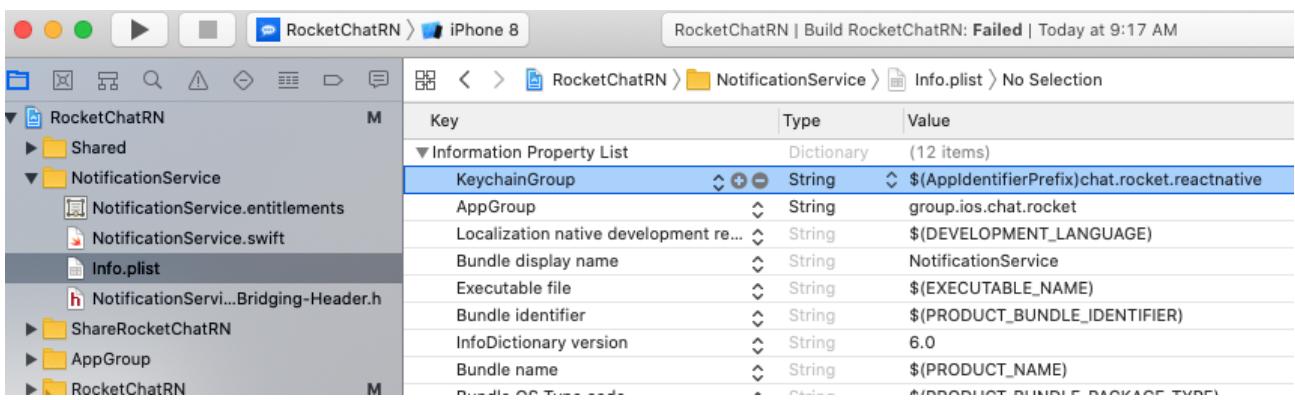
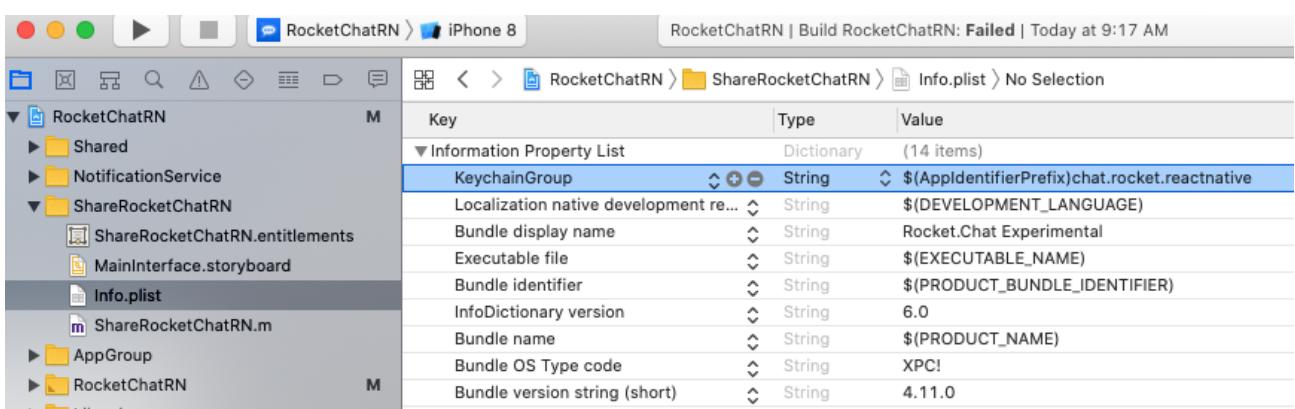
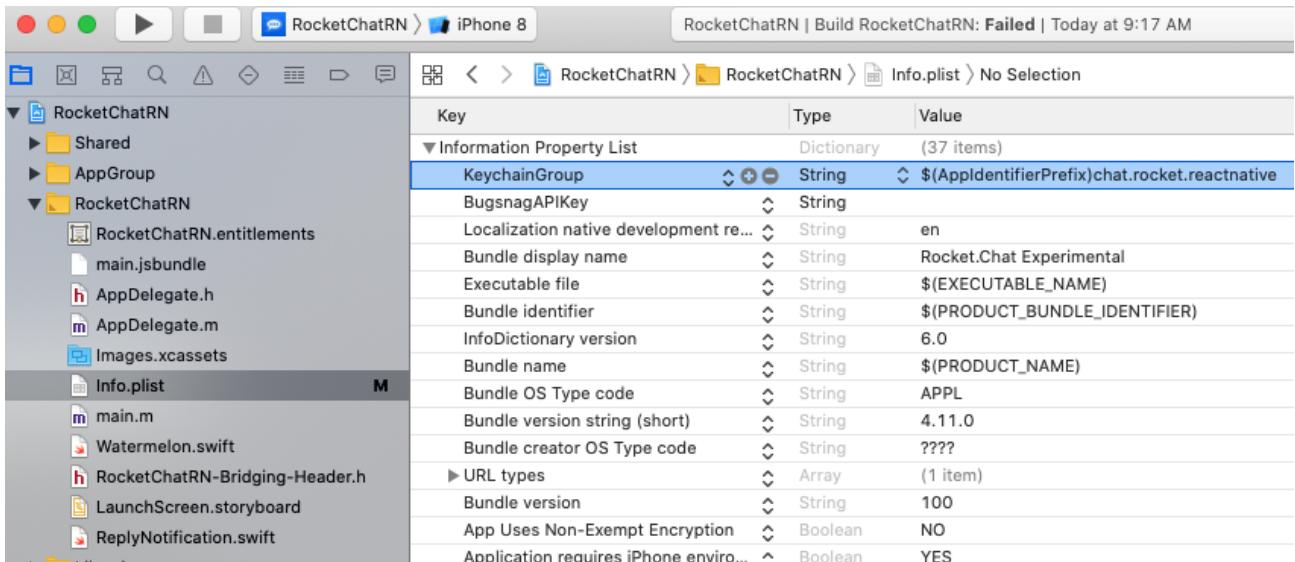
Key	Type	Value
▼ Information Property List	Dictionary	(35 items)
BugsnagAPIKey	String	
AppGroup	String	group.chat.rocket.whitelabel
Localization native development re...	String	en
Bundle display name	String	Rocket.Chat Experimental
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	4.5.1
Bundle creator OS Type code	String	????
▼ URL types	Array	(1 item)



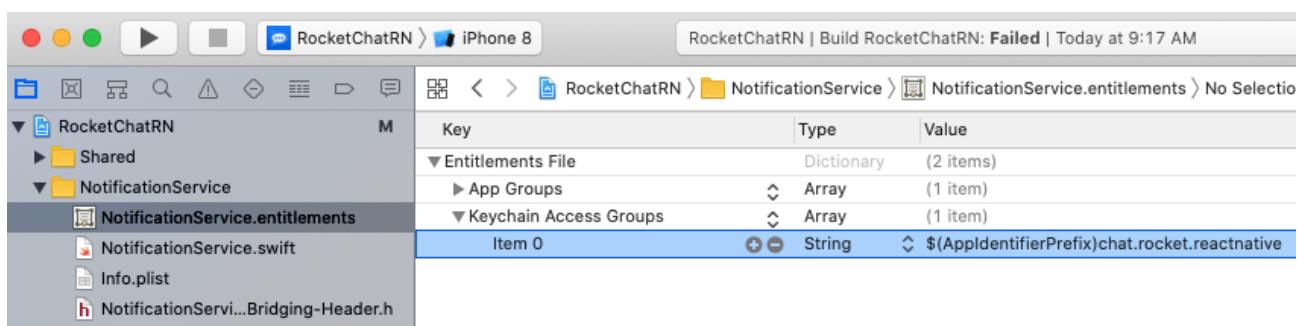
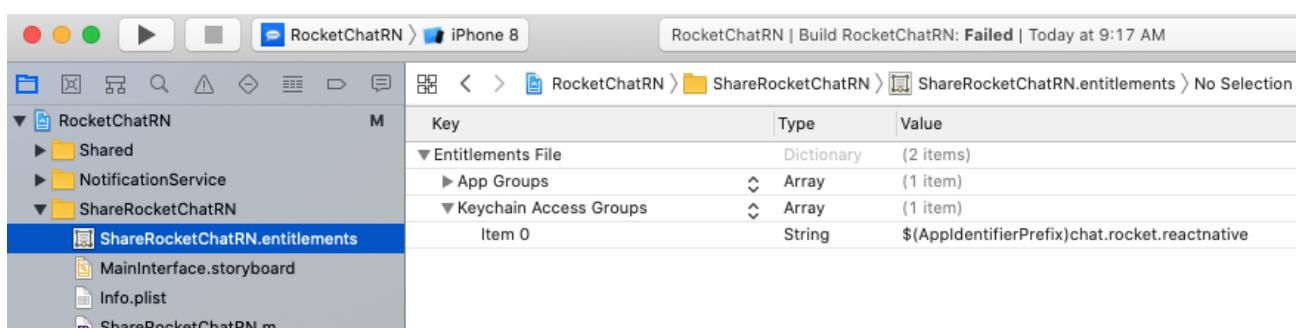
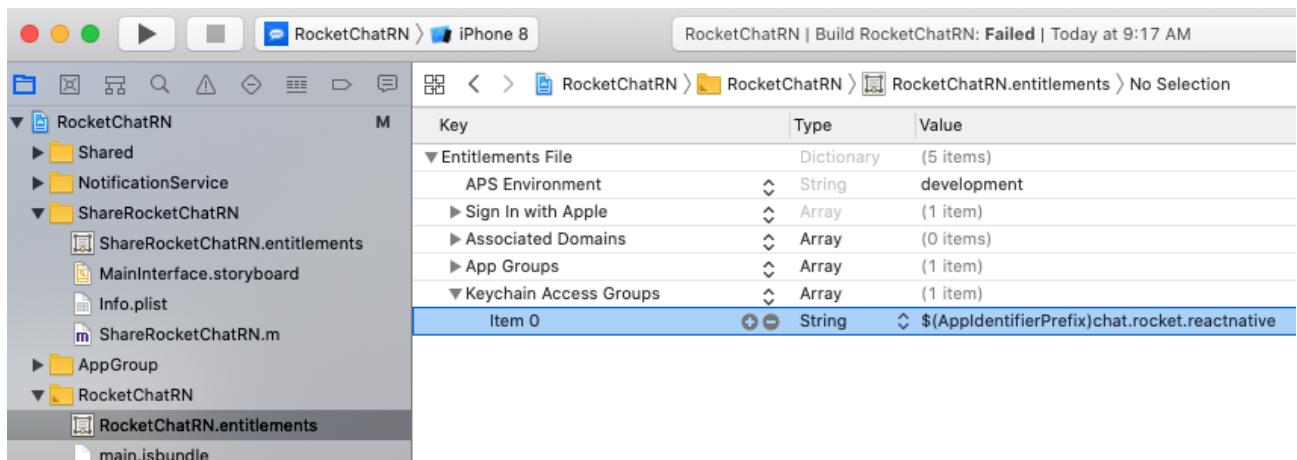
The screenshot shows the Xcode project structure for 'RocketChatRN'. The 'Info.plist' file is selected in the project navigator. The right-hand panel displays the contents of 'Info.plist' as a key-value table for the 'NotificationService' target.

Key	Type	Value
▼ Information Property List	Dictionary	(12 items)
AppGroup	String	group.chat.rocket.whitelabel
Localization native development re...	String	\$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	\$(PRODUCT_NAME)
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	\$(PRODUCT_BUNDLE_PACKAGE_TYPE)
Bundle version string (short)	String	\$(MARKETING_VERSION)
Bundle version	String	1
KeychainGroup	String	\$(AppIdentifierPrefix)chat.rocket.reactnative
► NSExtension	Dictionary	(2 items)

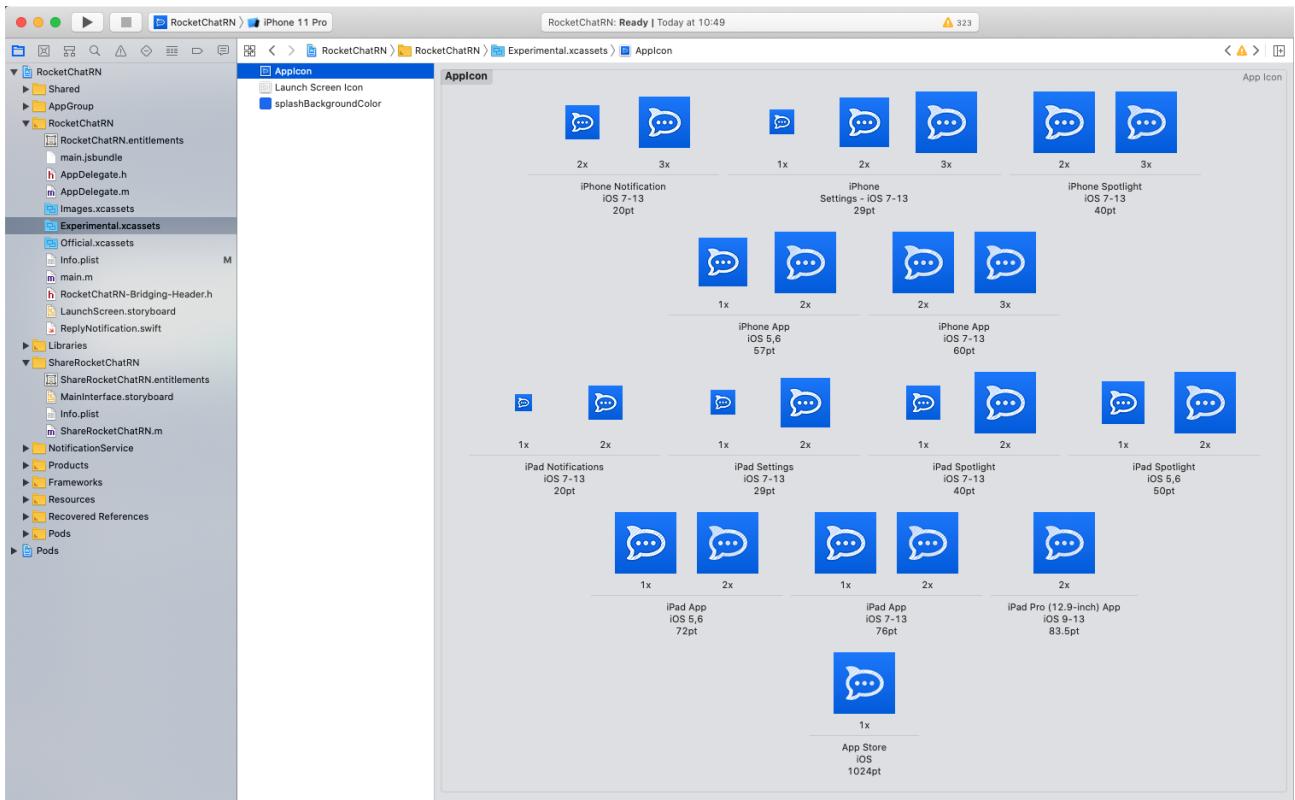
- Set the same keychain group on `RocketChatRN/Info.plist`, `ShareRocketChatRN/Info.plist` and `NotificationService/Info.plist`



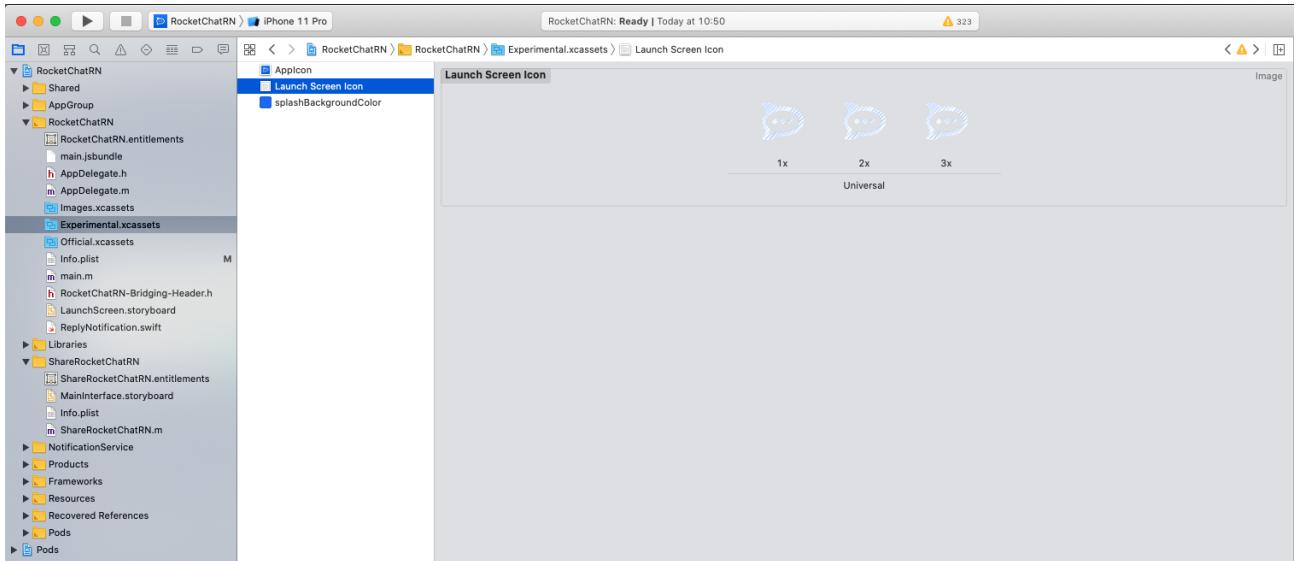
- It needs to be the same on all entitlements



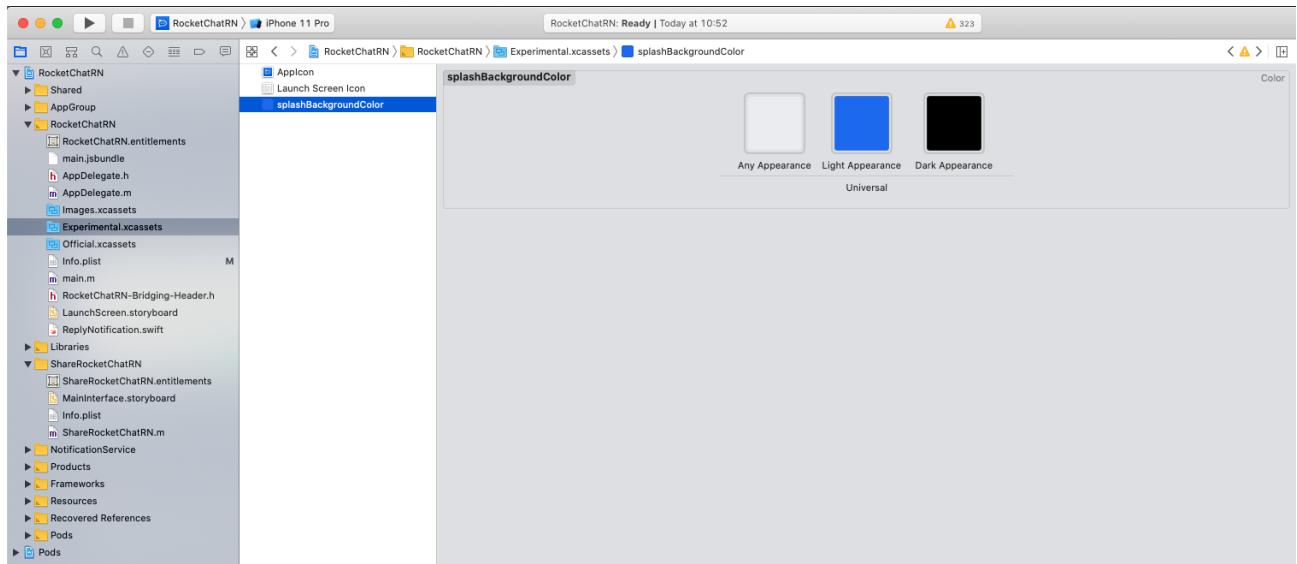
- Change the app icon on [Experimental.xcassets/App Icon](#)



- Change the app splash screen on Experimental.xcassets/Launch Screen Icon



- Change the splash background colors on Experimental.xcassets/splashBackgroundColor



- Set your Bugsnag API key on [RocketChatRN/Info.plist](#)

Key	Type	Value
▼ Information Property List		
BugsnagAPIKey	String	
AppGroup	String	group.chat.rocket.whitelabel
Localization native development re...	String	en
Bundle display name	String	Rocket.Chat Experimental
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	4.5.1
Bundle creator OS Type code	String	????
▼ URL types	Array	(1 item)

Generating iOS app on Firebase

- Visit the project overview on <https://console.firebaseio.google.com>
- Click on the gear icon and then “Project settings”

The screenshot shows the Firebase Project Overview page. At the top left is the Firebase logo and the word "Firebase". To the right is the project name "chat-rocket-whitelabel-test" with a dropdown arrow. Below the project name is a navigation bar with "Project Overview" and a gear icon. On the left, under the heading "Develop", there is a sidebar with icons and text for "Authentication", "Database", "Storage", "Hosting", and "Functions". A white callout box is open over the "Project settings" option in the main menu, listing "Project settings", "Users and permissions", and "Usage and billing".

- On “General” tab, click on “iOS” button under “Your apps” section

The screenshot shows the "Your apps" section in the Firebase console. It displays a message: "There are no apps in your project" and "Select a platform to get started". Below this are four circular icons for "iOS", "Android", "Web", and "Cloud Functions".

- Enter your bundle ID and then “Register app”

Add Firebase to your iOS app

1 Register app

iOS bundle ID ?

App nickname (optional) ?

App Store ID (optional) ?

Register app

2 Download config file

3 Add Firebase SDK

4 Add initialization code

5 Run your app to verify installation

- Download config file and move it as instructed

× Add Firebase to your iOS app



Register app

iOS bundle ID: chat.rocket.whitelabel



Download config file

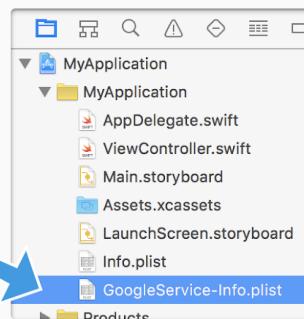
Instructions for Xcode below | [Unity](#) [C++](#)

[Download GoogleService-Info.plist](#)

Move the GoogleService-Info.plist file you just downloaded into the root of your Xcode project and add it to all targets.



GoogleService-Info.plist



[Previous](#)

[Next](#)



Add Firebase SDK

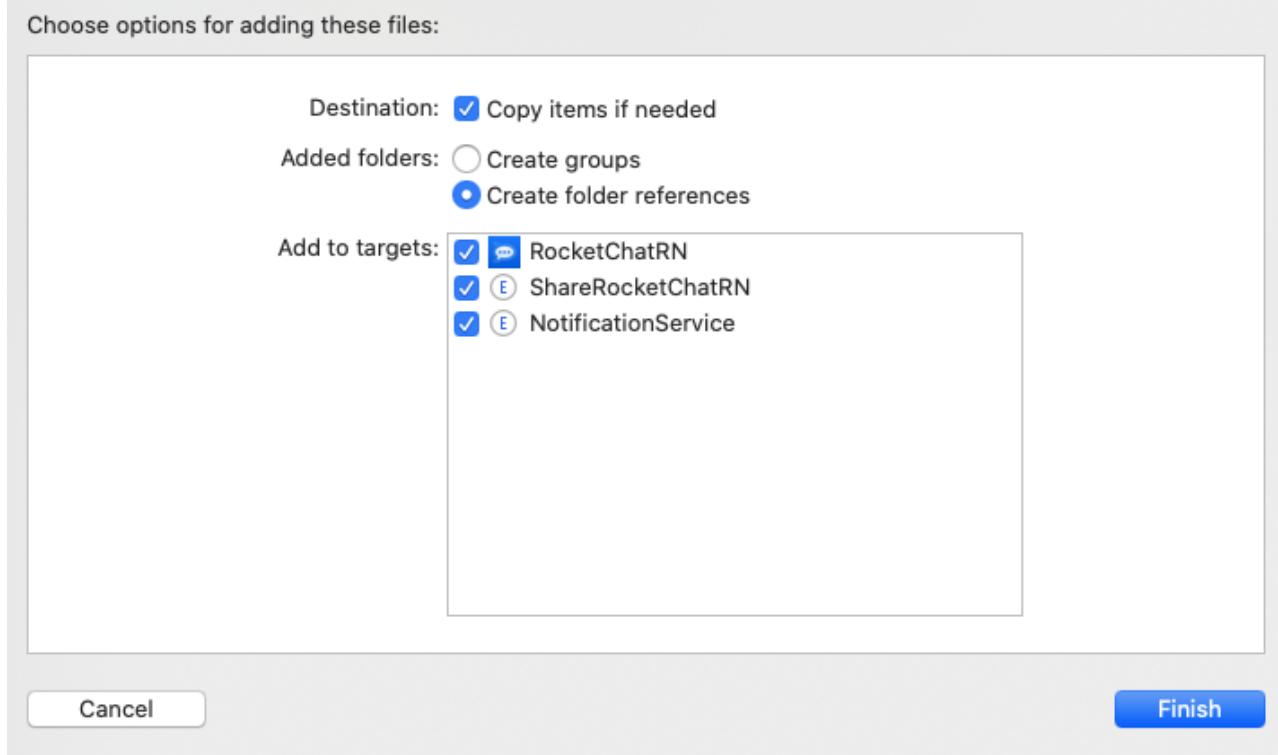


Add initialization code



Run your app to verify installation

- Add it to all targets



Running the app

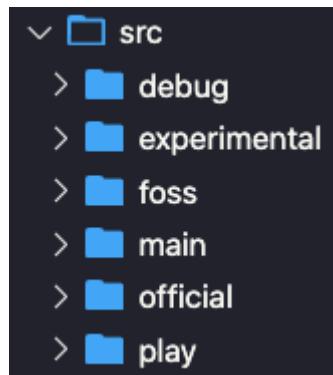
- Execute the following on project terminal
 - `yarn`
 - `npx pod-install`
 - `yarn ios`

Android

General setup

- Similarly to iOS, we have build flavours to generate our Official, Experimental and F-Droid versions of the app
 - `experimental` and `official` folders contain app icons and splash screens
 - `play` and `foss` folders contain necessary code to run the app with or without Google Play services, respectively
 - `foss` build doesn't contain push notifications implemented
 - `main` folder contains core implementations

- `debug` folder contains code to run the app in debug mode
- This doc is going to focus on building the Experimental app, so we're going to use `experimental`, `play`, `debug`, and `main` folders



- Set `APPLICATION_ID`, `VERSIONCODE` and `BugsnagAPIKey` on `./android/gradle.properties`
- Generate a **new image asset** for `ic_notification` and target `main`
- Generate a **new image asset** for `ic_launcher` and target `experimental`
- Change splash screen background and notification text color on `./android/app/src/experimental/res/values/colors.xml`

```

</> colors.xml <
      android > app > src > experimental > res > values > </> colors.xml
1   <?xml version="1.0" encoding="utf-8"?>
2   <resources>
3     <color name="primary_dark">#660B0B0B</color>
4     <item name="splashBackground" type="color">#1D74F5</item>
5     <item name="notification_text" type="color">#1D74F5</item>
6   </resources>

```

- Change splash screen logo on `./android/app/src/experimental/res/drawable-xxhdpi/splash.png`
- Change app name and share extension name on `./android/app/src/main/res/values/strings.xml`

strings.xml

```
1 <resources>
2   <string name="app_name">Rocket.Chat Experimental</string>
3   <string name="share_extension_name">Rocket.Chat Experimental</string>
4 </resources>
```

Generate upload key

- This step will generate the keystore that is going to verify your app on Google Play
 - You can use this guide as reference: <https://reactnative.dev/docs/signed-apk-android#generating-an-upload-key>
- Execute the following on terminal
 - cd android/app
 - keytool -genkeypair -v -keystore my-upload-key.keystore -alias my-key-alias
 - alias -keyalg RSA -keysize 2048 -validity 10000
- Credentials will be prompted

```
→ keytool -genkeypair -v -keystore my-upload-key.keystore -alias my-key-alias -keyalg RSA -keysize 2048 -validity 10000
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]:
What is the name of your organizational unit?
[Unknown]:
What is the name of your organization?
[Unknown]:
What is the name of your City or Locality?
[Unknown]:
What is the name of your State or Province?
[Unknown]:
What is the two-letter country code for this unit?
[Unknown]:
Is CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?
[no]:
What is your first and last name?
[Unknown]:
What is the name of your organizational unit?
[Unknown]:
What is the name of your organization?
[Unknown]:
What is the name of your City or Locality?
[Unknown]:
What is the name of your State or Province?
[Unknown]:
What is the two-letter country code for this unit?
[Unknown]:
Is CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?
[no]: yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
  for: CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
Enter key password for <my-key-alias>
  (RETURN if same as keystore password):
Re-enter new password:
[Storing my-upload-key.keystore]

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool -importkeystore -src keystore my-upload-key.keystore -destkeystore my-upload-key.keystore -deststoretype pkcs12".
```

- Set KEYSTORE_PASSWORD and KEY_PASSWORD on ./android/gradle.properties with the passwords you were prompted

Generating Android app on Firebase

- Visit the project overview on <https://console.firebaseio.google.com>
- Click on the gear icon and then “Project settings”

The screenshot shows the Firebase Project Overview page for a project named "chat-rocket-whitelabel-test". On the left, there's a sidebar with icons for Authentication, Database, Storage, Hosting, and Functions. The "Project Overview" tab is selected. On the right, a "Develop" section is visible. A "Project settings" dropdown menu is open, listing "Project settings", "Users and permissions", and "Usage and billing".

- On “General” tab, click on “Add app” button under “Your apps” section and then “Android”

The screenshot shows the "Your apps" section in the Firebase console. It lists an "iOS app" named "chat.rocket.whitelabel". To the right of the app name, there are links to "Download the latest config file" (with a download icon) and "GoogleService-Info.plist" (with a download icon). Below the app name, there is a note: "This file contains configuration details such as keys and identifiers, for the services you just enabled." At the bottom of the screen, there is a link for "App ID" with a question mark icon.

Add Firebase to your app

X

Select a platform to get started



Android

App Store ID [?](#)

- Enter your bundle ID and then “Register app”

× Add Firebase to your Android app

1 Register app

Android package name [?](#)

chat.rocket.whitelabel

App nickname (optional) [?](#)

My Android App

Debug signing certificate SHA-1 (optional) [?](#)

00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:(

Required for Dynamic Links, Invites, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

Register app

- Download config file and move it as instructed

× Add Firebase to your Android app

1 Register app

Android package name: chat.rocket.whitelabel

2 Download config file

Instructions for Android Studio below | [Unity](#) [C++](#)

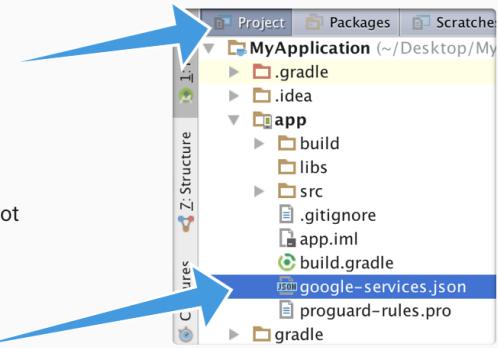
[!\[\]\(b2ab433931bef9c76490f5f7b0063233_img.jpg\) Download google-services.json](#)

Switch to the **Project** view in Android Studio to see your project root directory.

Move the google-services.json file you just downloaded into your Android app module root directory.



google-services.json

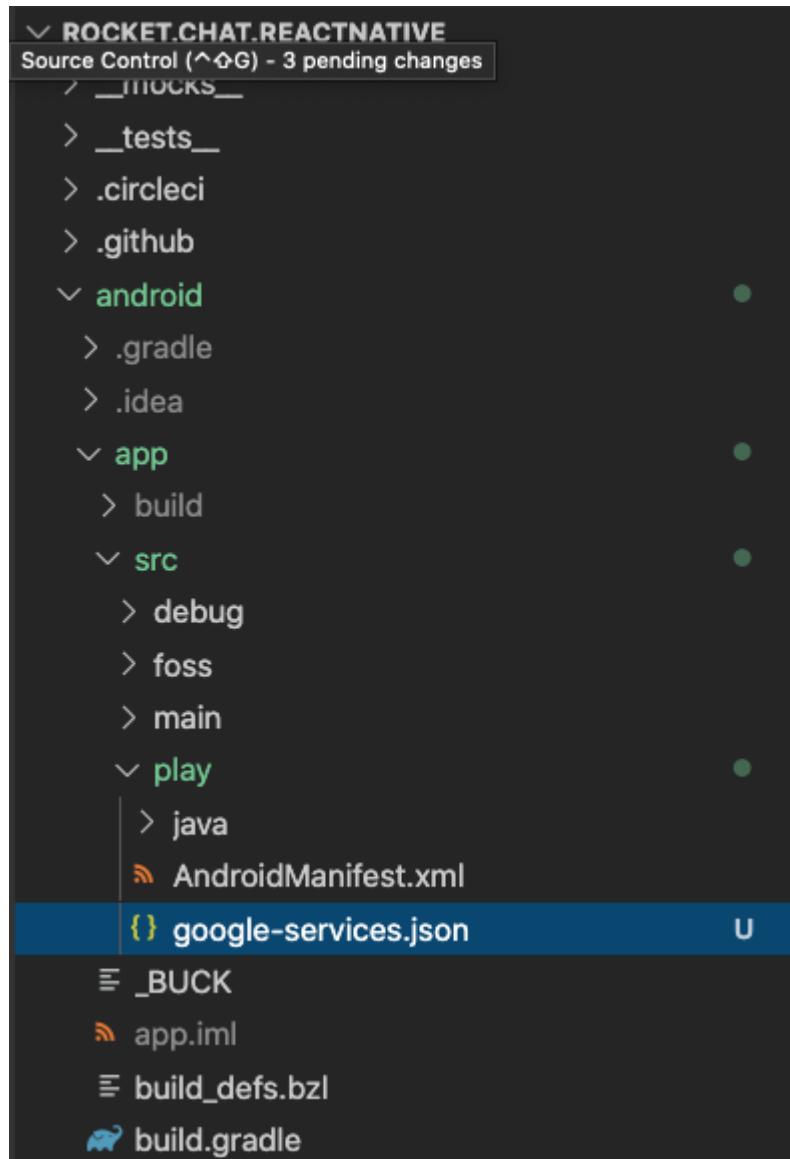


[Previous](#)

[Next](#)

3 Add Firebase SDK

4 Run your app to verify installation



Running the app

- Execute the following on project terminal
 - `yarn`
 - `yarn android-whitelabel <YOURAPPID>`
 - For example, the app created on this document would use

```
yarn android-whitelabel chat.rocket.whitelabel
```
- Note: this script uses `experimentalPlayDebug` build flavor. When you build your app on release mode, use `experimentalPlayRelease`
 - Refer to <https://developer.android.com/studio/build/build-variants> for more info about how it works

Push notification

Configuring gateway

- Go to your Rocket.Chat admin page > Push
- Disable “Gateway” and press “Save changes”

The screenshot shows the 'Push' configuration page in the Rocket.Chat admin interface. At the top right is a blue 'Save changes' button. Below it, under the heading 'Batch size to be processed every tick', is a text input field containing '10'. A yellow warning bar below the input field states: '⚠ Changing this value requires restarting Rocket.Chat.' Further down, there is a toggle switch labeled 'Enable Gateway' with its current state set to off (grey). Another yellow warning bar next to the switch says: '⚠ Changing this value requires restarting Rocket.Chat.' At the bottom of the section is a text input field labeled 'Gateway' containing 'https://gateway.rocket.chat'. A note below the field says: 'Multiple lines can be used to specify multiple gateways.'

- Also disable “Production”, if you’re trying in debug mode
- Expand “Credentials and Keys” section

Configuring Android

- Go to Cloud Messaging on Firebase settings
- Copy “Server Key” token from Firebase into “GCM API Key”
- Copy “Sender ID” into “GCM Project Number”

Settings

?

General Cloud Messaging Integrations Service accounts Data privacy Users and permissions

Project credentials

Add server key

Key	Token
Server key	AAA eVJ Qon [REDACTED]
Legacy server key <small>?</small>	Alz [REDACTED]
Sender ID <small>?</small>	[REDACTED]
8C [REDACTED]	

Configuring iOS

- Make sure you've done "Creating Push Notifications certificates" first
- In your terminal, go to the folder which contains your push files (CSR, .cer, .p12).

Generating PEM files (Development)

- Execute
 - `openssl x509 -in aps_development.cer -inform der -out DevPushCert.pem`
 - `openssl pkcs12 -nocerts -out DevPushKey.pem -in yourP12File.p12`
- You **must** set a password for your PEM file

Generating PEM files (Production)

- Execute
 - `openssl x509 -in aps.cer -inform der -out PushCert.pem`
 - `openssl pkcs12 -nocerts -out PushKey.pem -in yourP12File.p12`
- You **must** set a password for your PEM file

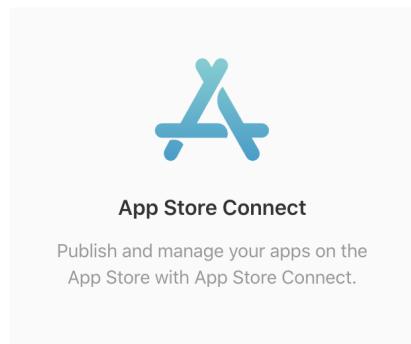
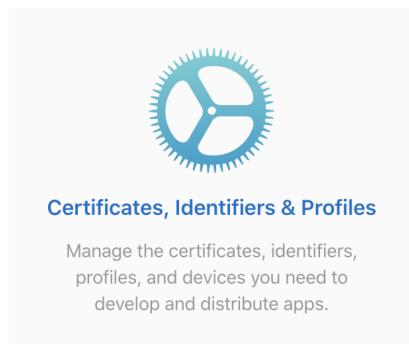
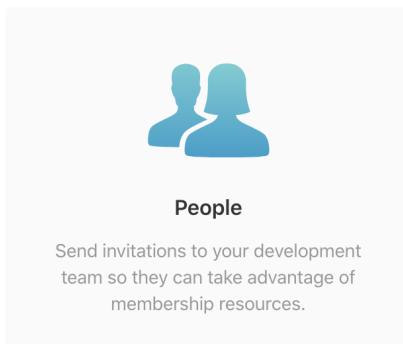
Copying PEM files to Rocket.Chat

- Copy the contents of your development PEM files and password into APN Dev Key, APN Dev Cert and APN Dev Passphrase
 - Copy the contents of your production PEM files and password into APN Key, APN Cert and APN Passphrase
 - You can use `cat` on terminal to get the content of your PEM files
 - `cat PushKey.pem`
 - Save and restart your server
 - Log into the server as the same user on your mobile device and close it (it won't receive push notification, if it's open)
 - Open Push settings on admin from desktop and click "Send a test push to my user"
-

Developer Apple

Login to Apple Developer

- Visit <https://developer.apple.com/account>
- Enter your credentials
- Click on Certificates, Identifiers & Profiles



Creating an App Identifier

- Visit <https://developer.apple.com/account/resources/identifiers/list>
- Click to add Identifier
- Select App IDs and Continue

< All Identifiers

Register a New Identifier

Continue

App IDs

Register an App ID to enable your app to access available services and identify your app in a provisioning profile. You can enable app services when you create an App ID or modify these settings later.

Services IDs

For each website that uses Sign in with Apple, register a services identifier (Services ID), configure your domain and return URL, and create an associated private key.

Pass Type IDs

Register a pass type identifier (Pass Type ID) for each kind of pass you create (i.e. gift cards). Registering your Pass Type IDs lets you generate Apple-issued certificates which are used to digitally sign and send updates to your passes, and allow your passes to be recognized by Wallet.

Website Push IDs

Register a Website Push Identifier (Website Push ID). Registering your Website Push IDs lets you generate Apple-issued certificates which are used to digitally sign and send push notifications from your website to macOS.

- Add description and Bundle ID

Platform

iOS, macOS, tvOS, watchOS

Description

Whitelabel example

You cannot use special characters such as @, &, *, ;, "

App ID Prefix

S6UPZG7ZR3 (Team ID)

Bundle ID

Explicit

Wildcard

chat.rocket.whitelabel|

We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).

- On Capabilities, select App Groups and Push notifications
- Click “Continue” and then “Register”

Creating an App Identifier for our Share Extension

- Share Extension is a version of the app that opens when you share data from another app to Rocket.Chat. For example, share a photo from the gallery.
- Visit <https://developer.apple.com/account/resources/identifiers/list>
- Click to add Identifier
- Select App IDs and Continue

[◀ All Identifiers](#)

Register a New Identifier

[Continue](#)

App IDs

Register an App ID to enable your app to access available services and identify your app in a provisioning profile. You can enable app services when you create an App ID or modify these settings later.

Services IDs

For each website that uses Sign in with Apple, register a services identifier (Services ID), configure your domain and return URL, and create an associated private key.

Pass Type IDs

Register a pass type identifier (Pass Type ID) for each kind of pass you create (i.e. gift cards). Registering your Pass Type IDs lets you generate Apple-issued certificates which are used to digitally sign and send updates to your passes, and allow your passes to be recognized by Wallet.

Website Push IDs

Register a Website Push Identifier (Website Push ID). Registering your Website Push IDs lets you generate Apple-issued certificates which are used to digitally sign and send push notifications from your website to macOS.

- Add description and Bundle ID

[◀ All Identifiers](#)

Register an App ID

[Back](#) [Continue](#)

Platform

iOS, macOS, tvOS, watchOS

App ID Prefix

S6UPZG7ZR3 (Team ID)

Description

Whitelabel share extension

Bundle ID Explicit Wildcard

chat.rocket.whitelabel.ShareExtension

You cannot use special characters such as @, &, *, :, "

We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).

Capabilities

ENABLED NAME

 Access WiFi Information [ⓘ](#)

 App Groups [ⓘ](#)

- This time, select only App Groups under Capabilities
- Click “Continue” and then “Register”

Create an App Identifier for our Notification Service

[All Identifiers](#)

Register an App ID

[Back](#) [Continue](#)

Platform
iOS, macOS, tvOS, watchOS

App ID Prefix
S6UPZG7ZR3 (Team ID)

Description
Whitelabel Notification

Bundle ID Explicit Wildcard
chat.rocket.whitelabel.NotificationService

You cannot use special characters such as @, &, *, :, ", -, .

We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).

Capabilities

ENABLED NAME

-  Access WiFi Information (i)
-  App Attest (i)
-  App Groups (i)

Creating an App Group

- Visit <https://developer.apple.com/account/resources/identifiers/list>
- Click to add Identifier
- Select App Groups and Continue

[All Identifiers](#)

Register a New Identifier

[Continue](#)

App IDs

Register an App ID to enable your app to access available services and identify your app in a provisioning profile. You can enable app services when you create an App ID or modify these settings later.

Services IDs

For each website that uses Sign in with Apple, register a services identifier (Services ID), configure your domain and return URL, and create an associated private key.

Pass Type IDs

Register a pass type identifier (Pass Type ID) for each kind of pass you create (i.e. gift cards). Registering your Pass Type IDs lets you generate Apple-issued certificates which are used to digitally sign and send updates to your passes, and allow your passes to be recognized by Wallet.

Website Push IDs

Register a Website Push Identifier (Website Push ID). Registering your Website Push IDs lets you generate Apple-issued certificates which are used to digitally sign and send push notifications from your website to macOS.

iCloud Containers

Registering your iCloud Container lets you use the iCloud Storage APIs to enable your apps to store data and documents in iCloud, keeping your apps up to date automatically.

App Groups

Registering your App Group allows access to group containers that are shared among multiple related apps, and allows certain additional interprocess communication between the apps.

- Enter a description and an Identifier

[« All Identifiers](#)

Register an App Group

[Back](#) [Continue](#)

Description

Whitelabel App Group

You cannot use special characters such as @, &, *, :, "

Identifier

group.chat.rocket.whitelabel

We recommend using a reverse-domain name style string (i.e., com.domainname.appname).

- Click “Continue” and then “Register”

Applying App Group

- Visit <https://developer.apple.com/account/resources/identifiers/list>
- Click on the first identifier you created
- On “App Groups”, click “Configure”
- Select the App Group you created and click “Continue”
- Click “Save”
- Repeat these steps for the second identifier you created for the Share Extension and NotificationService

Creating Push Notifications certificates

- Visit <https://developer.apple.com/account/resources/identifiers/list>
- Click on the first identifier you created
- On “Push Notifications”, click “Configure”

Apple Push Notification service SSL Certificates

To configure push notifications for this App ID, a Client SSL Certificate that allows your notification server to connect to the Apple Push Notification Service is required. Each App ID requires its own Client SSL Certificate. Manage and generate your certificates below.

Development SSL Certificate

Create an additional certificate to use for this App ID.

[Create Certificate](#)

Production SSL Certificate

Create an additional certificate to use for this App ID.

[Create Certificate](#)

[Done](#)

Development SSL Certificate

- On “Development SSL Certificate”, click “Create Certificate”
- Follow Apple’s tutorial to generate a Certificate Signing Request:
<https://help.apple.com/developer-account/#/devbfa00fef7>
- Select the certificate you created an click “Continue”

[« All Certificates](#)

Create a New Certificate

[Back](#) [Continue](#)

Certificate Type

Apple Push Notification service SSL (Sandbox)

Platform:

iOS

Upload a Certificate Signing Request

To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Mac.

[Learn more >](#)

[Choose File](#)

CertificateSigningRequest.certSigningRequest

- Download the certificate and install it on your machine (follow instructions on the screen)

[« All Certificates](#)

Download Your Certificate

[Revoke](#) [Download](#)

Certificate Details

Certificate Name	Certificate Type	Download your certificate to your Mac, then double click the .cer file to install in Keychain Access. Make sure to save a backup copy of your private and public keys somewhere secure.
chat.rocket.whitelabel	APNs Development iOS	

Expiration Date
2021/03/10

Created By
Diego Mello (diegomello@gmail.com)

- After installing it, “Keychain Access” should have opened automatically on your Mac
- Export the certificate to generate a .p12 file



- For simplicity, save it in the same folder of your CSR and .cer. You'll need it later.

Production SSL Certificate

- On “Production SSL Certificate”, click “Create Certificate”
- Follow Apple’s tutorial to generate a Certificate Signing Request:
<https://help.apple.com/developer-account/#/devbfa00fef7>
- Select the certificate you created and click “Continue”

< All Certificates

Create a New Certificate

[Back](#) [Continue](#)

Certificate Type

Apple Push Notification service SSL (Sandbox & Production)

Platform:

iOS

Upload a Certificate Signing Request

To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Mac.

[Learn more >](#)

[Choose File](#)

CertificateSigningRequest.certSigningRequest

- Download the certificate and install it on your machine (follow instructions on the screen)

< All Certificates

Download Your Certificate

[Revoke](#) [Download](#)

Certificate Details

Certificate Name
chat.rocket.whitelabel

Certificate Type
Apple Push Services

Download your certificate to your Mac, then double click the .cer file to install in Keychain Access. Make sure to save a backup copy of your private and public keys somewhere secure.

Expiration Date
2021/04/09

Created By
Diego Mello (diegolmello@gmail.com)

- After installing it, “Keychain Access” should have opened automatically on your Mac
- Export the certificate to generate a .p12 file



- For simplicity, save it in the same folder of your CSR and .cer. You'll need it later.

Supporting SSL for development on Rocket.Chat

If you are working with mobile apps, it is required that your server supports SSL.

Rocket.Chat is a "middle-tier application server", by itself it does not handle SSL. However, Rocket.Chat works well with several industrial grade, battle-tested, reverse proxy servers that you can configure to handle SSL.

You should find yourself in one of the two situations:

- The Rocket.Chat server is publicly accessible on the internet.
- The Rocket.Chat server is not accessible on the internet.

This doc has been broken down into two separate sections, walking you through either of the camps you might find yourself in.

The Rocket.Chat server is publicly accessible on the internet

If your server is publicly accessible, it is recommended that you use a service like [Let's Encrypt](#) to obtain your SSL certificates. A detailed guide for configuring your choice of SSL Reverse proxy servers is provided here: [Configuring SSL Reverse Proxy](#)

If you are using Ubuntu, this can be configured automatically with the help of Snaps. A guide for which is provided here: [Installing Rocket.Chat on Ubuntu with Snaps](#)

The Rocket.Chat server is not accessible on the internet

If your server is not accessible on the internet, you will need to provide self signed certificates to configure SSL on the server.

In this doc, we will be creating a self signed root certificate and using it to generate our SSL certificates. The steps written below have been adapted from [Self Signed Certificate with Custom Root CA](#)

Step 1: Create Root CA

Create Root Key

Attention: This is the key used to sign the certificate requests, anyone holding this can sign certificates on your behalf. So keep it in a safe place!

```
openssl genrsa -des3 -out Rocket.Chat-root.key 4096
```

If you want a non password protected key just remove the `-des3` option

Create and self sign the Root Certificate

```
openssl req -x509 -new -nodes -key Rocket.Chat-root.key -sha256 -days 1024 -
```

Here we used our root key to create the root certificate that needs to be distributed in all the computers that have to trust us.

NOTE: It is **not recommended** that you distribute this root certificate in production. A breach of the above-generated key will open every device that trusts your root certificate to ***potential security threats***.

Step 2: Create an SSL certificate

Create the certificate key

```
openssl genrsa -out mydomain.com.key 2048
```

Here, mydomain.com should be replaced with your IP address ([Bonjour](#) local domains work as well!)

Create the certificate signing request

Important: Please mind that while creating the certificate signing request is important to specify the `Common Name` providing the IP address or URL for the service, otherwise the certificate cannot be verified.

```
openssl req -new -key mydomain.com.key -out mydomain.com.csr
```

Generate the SSL certificate

Here, we are using the `mydomain.com` CSR along with the `Rocket.Chat-root` CA.

```
openssl x509 -req -in mydomain.com.csr -CA Rocket.Chat-root.crt -CAkey Rocket
```

Step 3: Configuring SSL for Rocket.Chat

The `mydomain.com.crt` and `mydomain.com.key` files generated above will be used as the certificate and the private key to configure SSL.

A detailed guide for configuring your choice of SSL Reverse proxy servers is provided here:
[Configuring SSL Reverse Proxy](#)

Step 4: Trusting Certificate Authority

All the devices that need to communicate with the server during development, need to trust the root certificate we generated in `Step 1 (Rocket.Chat-root.crt)`

- For Apple devices follow the instructions here: [HTTPS and Test Servers](#)
- For Android devices follow the link [Add & remove certificates](#), and scroll down to "Work with CA certificates (trusted credentials)".

Installation instructions for other operating systems can be easily found online.

On successful installation of the root certificate, the device should be able to access Rocket.Chat over SSL.

Troubleshooting

If your device is not able to connect over SSL, please make sure that the URL has `https://` explicitly typed out before it.

Analytics & Data Usage

Crashlytics and Bugsnag

Both iOS and Android applications are using Crashlytics and Bugsnag to send crash information. Both platforms collect crash information from mobile apps, transmit and store them securely on their servers for developer diagnosis. This transmitted data contains no users, channels, or groups information; it does not contain any message content. The collected content contains only anonymous application and system state information during the crash that can be helpful for diagnosis.

These reports can be disabled on both iOS and Android if the user goes to the Settings screen and disable crash reports.

Firebase Analytics

We use Firebase Analytics report events of usage in the app for some actions, such as sending messages, reacting, changing the theme of the app, etc. This is also anonymous information and does not contain any private information from the user. This is only being used with the purpose of understanding what features are being used more and how they're being used.

These reports can be disabled on both iOS and Android if the user goes to the Settings screen and disable crash reports.

UI and Theming

NB: Theming for Rocket.Chat is an incomplete feature. You can follow discussion on the approach under issue [#277](#)

- [How to create custom themes](#)
- [How to use UI color scheme](#)
- [How to use UI components](#)

UI Components

This article stub needs descriptive content and example usage for each component declared in Rocket.Chat UI to improve ease of development and prevent furthering multiple components being contributed for the same purpose with conflicting styles and usage. [Can you help?](#)

Example content:

Side Nav

Describe usage for template development contributions

Provide example usage and html structure / class names of nested elements

- Content
 - Admin Link
 - Input Line
 - Button
- Account Box
 - Status
 - Account Link
- Rooms List
 - Header
 - Item
 - Item List

Custom Themes

Theming for Rocket.Chat is an incomplete feature and we encourage developers to contribute to [this issue](#). We'd love to hear from anyone working on themes in the [#skins-and-theming](#) group channel.

To customize the Rocket.Chat UI you can either modify the `rocketchat-theme` or `rocketchat-ui` packages directly, but if you're keeping in sync with active development it would be easier to avoid conflicts by creating your own theme package.

Creating a Theme

You can add theme customisations to Rocket.Chat by just creating a Meteor package with your code, then adding it to the packages file.

Private themes would need to be maintained on your own fork of Rocket.Chat, but public themes could be published as a Meteor package outside the Rocket.Chat repo.

The minimum contents for a theme package would be a `package.js` file containing the description, e.g:

```
1 Package.describe({
2     name: 'author:mytheme',
3     version: '0.0.1',
4     summary: 'My theme customisations.',
5     git: 'https://github.com/author/my-rocketchat-theme'
6 });
```

Then include dependent packages and your custom theme files. e.g:

```
1 Package.onUse(function(api) {
2     api.versionsFrom('1.2');
3     api.use([
```

```
4         'templating',
5         'rocketchat:lib',
6         'rocketchat:theme'
7     ]);
8     api.use('templating', 'client');
```

Adding Stylesheets

The `rocketchat-theme` package has methods for including `Less` asset files in the build. Less files (and the a `server.coffee` or `.js` file to load them) must first be included in the `package.js` manifest (within the `Package.onUse` function), e.g.:

```
1     api.addAssets([
2         'assets/theme.less'
3     ], 'server');
4     api.addFiles([
5         'server.coffee'
6     ], 'server');
```

Then in `server.coffee` ...

```
RocketChat.theme.addPackageAsset -> Assets.getText 'assets/theme.less'
```

That will read in any styles and variables from your custom less file and compile it with the rest of the css.

Adding and Modifying Templates

A suggested approach for including custom templates and helpers is to use the `'aldeed:template-extension'` package (include it in your main packages file). In your

package manifest, declare use of the `template-extension` package, then add your template files to Meteor `api.addFiles(['myfiles'], 'client')`.

Here's an example replacing the unauthorized page template:

In `package.js`

```
api.addFiles(['views/notAuthorized.html', 'client.coffee'], 'client');
```

In `views/notAuthorized.html`

```
1 <template name="myNotAuthorized">
2   <h2>My Custom Not Authorized Page</h2>
3 </template>
```

In `client.coffee`

```
Template.myNotAuthorized.replaces 'notAuthorized'
```

See the [docs for that package](#) for more info on inheriting and overwriting templates and helpers.

UI Colors

NB: The UI is in active development and component refactoring will change (and hopefully improve) the use of color. The immediate goal is to consolidate use of color and styles, before moving forward to a consistent style guide for Rocket.Chat.

Most* use of color in Rocket.Chat can be customised by changing color settings under Administration > Layout > Colors. We encourage developers to use the defined variables in their contributions (instead of hard-coding colors), to allow site owners to change the color scheme with consistent results.

See the [theme variables file](#) for current available color settings. All color settings are available to Less files as variables, as long as the Less files were compiled using the

`addPackageAsset` method of `rocketchat-theme`.

Color Scheme

The Rocket.Chat color scheme consists of three groups of color settings, Alpha, Major and Minor Colors.

Some further variations of these colors are created in Less and not exposed to settings.

The naming of color settings/variables is not related to any specific component, the names reflect the visual hierarchy that will (hopefully) make it obvious and easy to carry forward consistent color usage in newly contributed components and theme development.

Alpha Colors

Semi-transparent black or white, used in components to shade/tint the background color, e.g. to indicate a selected or disabled state. The use of alpha colors allows site owners to easily change color scheme without defining every variation for every state of a component.

- transparent-dark
- transparent-darker

- transparent-light
- transparent-lighter

Alpha Colors Example

transparent-darker	15% black	✓			
transparent-dark	3% black	✗			
transparent-light	60% white	✓			
transparent-lighter	25% white	✓			
	primary-background-color		dark bg	light bg	example custom bg color

Alpha colors example colors

Major Colors

The primary palette of the app. Contributions and modifications to components should make use of these colors.

- content-background-color **#FFFFFF**
- primary-background-color **#04436A**
- primary-font-color **#444444**
- primary-action-color **#1d74f5**
- secondary-background-color **#F4F4F4**
- secondary-font-color **#A0A0A0**
- secondary-action-color **#DDDDDD**
- component-color **#f2f3f5**
- success-color **#4dff4d**
- pending-color **#FCB316**
- error-color **#BC2031**
- selection-color **#02ACEC**
- attention-color **#9C27B0**

Minor Colors

A set of minor colors for specific use cases will inherit from the major colors by default but can be used by admins who want more granular control over the color scheme.

- `tertiary-background-color` *defaults to component-color*
- `tertiary-font-color` *defaults to transparent-light*
- `link-font-color` *defaults to primary-action-color*
- `info-font-color` *defaults to secondary-font-color*
- `custom-scrollbar-color` *defaults to transparent-dark*
- `status-online` *defaults to success-color*
- `status-away` *defaults to pending-color*
- `status-busy` *defaults to error-color*
- `status-offline` *defaults to transparent-darker*

Dark UI

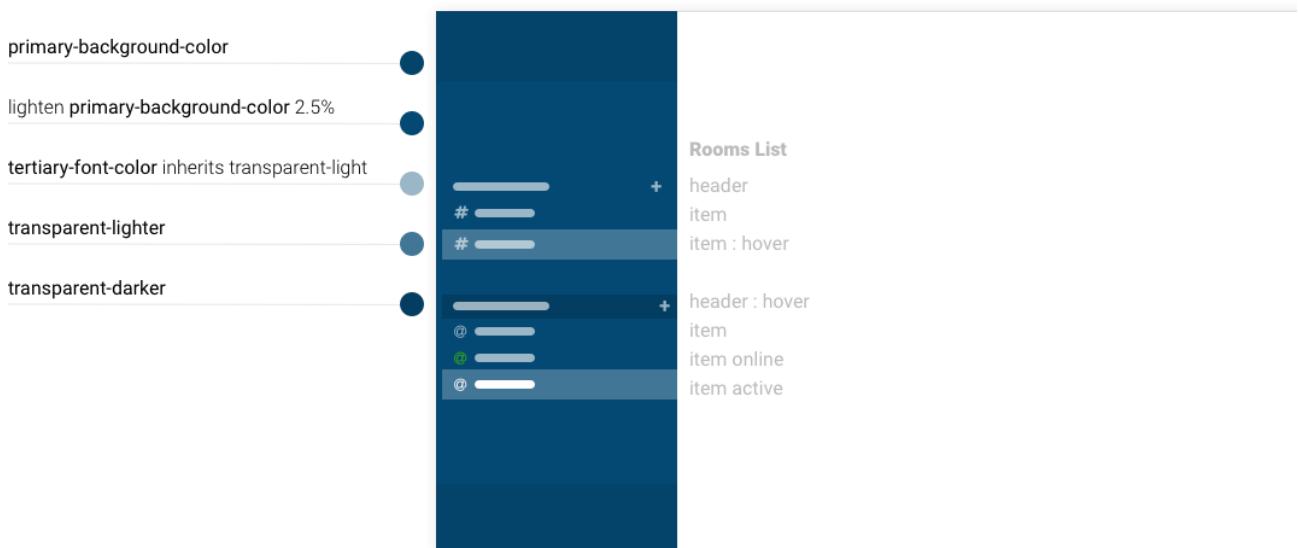
The computed colors allow owners to choose a **dark UI** with appropriate contrast. e.g. in a light UI, the disabled state might darken an element, but on a dark UI it should be lightened. The mixins achieve this by mixing the color with a contrast of the background color instead of using darken/lighten. See [this example](#) of form input states that dynamically contrast to both dark and light backgrounds.

Developers are encouraged to use mixins and computed colors in contributions instead of hard-coding variations, to maintain consistent balance and contrasts of colors regardless how the scheme settings may be changed by owners.

Default Colors

These examples show the implementation of the default color scheme with the main components of the Rocket.Chat UI.

Side Nav



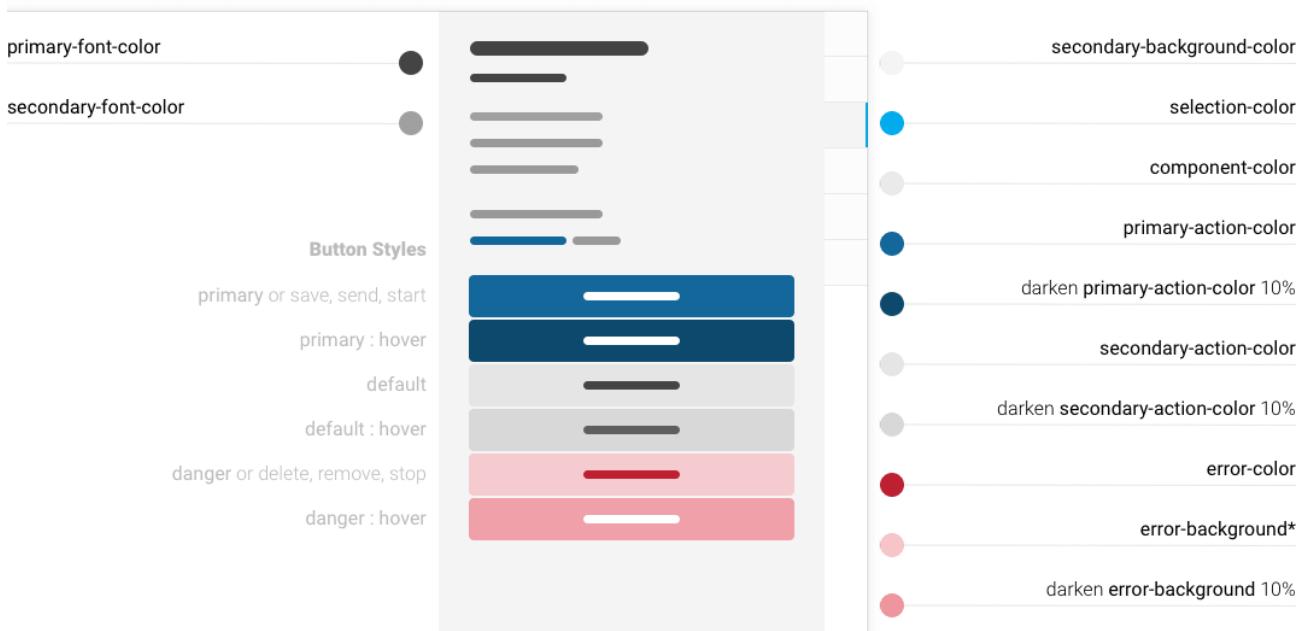
Side nav example colors

Account Box



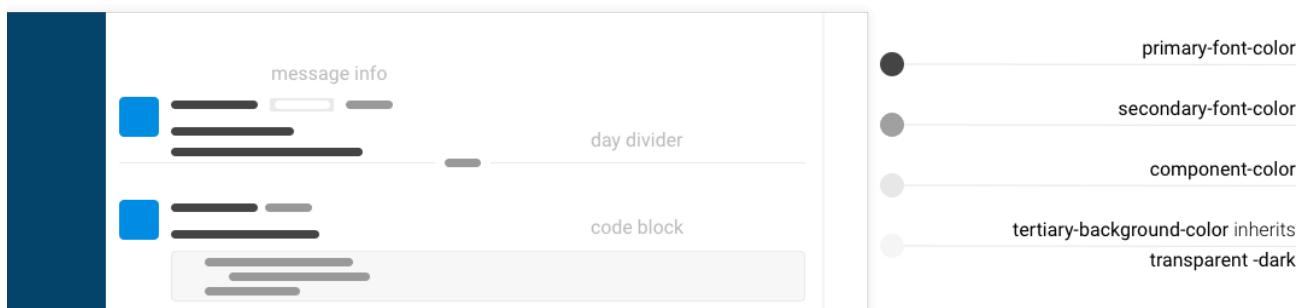
Account Box example colors

Flex Nav



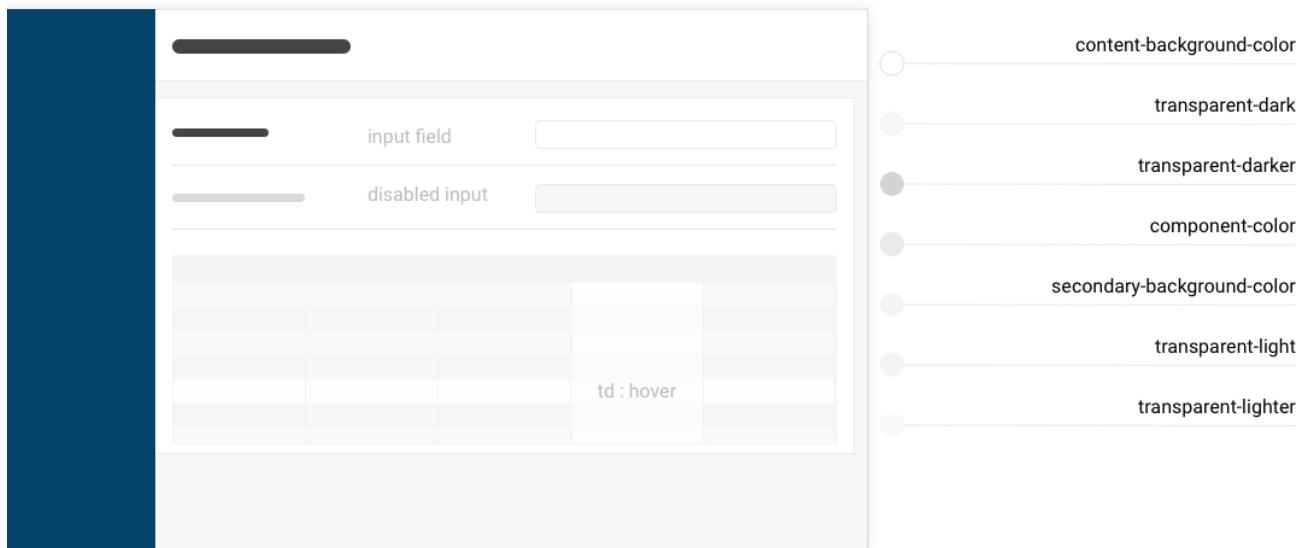
Flex nav example colors

Message Box



Message box example colors

Settings Page



Settings page example colors

Deeplink

General information

Our mobile and desktop clients have implemented deeplinking that will allow them to handle links meant for Rocket.Chat.

This is done using a special url that the applications are registered to handle.

There are two prefixes available:

- `https://go.rocket.chat`
- `rocketchat://`

We call `https://go.rocket.chat` links go links. Links here will work across web, mobile and desktop clients.

The alternative will only be handled by mobile and desktop clients.

In all examples below `https://go.rocket.chat/` is interchangeable with `rocketchat://`

Authentication

These links can be used for adding a server to your client in one click. This makes it easier for deployment. You then have a universal link that will add to the clients.

You can also include credentials to make it add the server and auto log them in.

Authentication links start with: `https://go.rocket.chat/auth`

Params:

- host: The host of the server;
- token (optional): The token of the user to be authenticated;

- `userId` (optional): The id of the user to be authenticated;

Examples:

Link directly to server:

```
https://go.rocket.chat/auth?host=open.rocket.chat
```

Link directly to server and authenticate:

```
https://go.rocket.chat/auth?host=foo.bar.com&token=123abc&userId=1234abcd
```

Channel / Group / DM

You can also link directly to a room.

These links start with: `https://go.rocket.chat/room`

Params:

- `host`: The host of the server;
- `rid`: The rid of the room to be opened;
- `path` (optional): The path URL to be opened on the web;

Example:

```
https://go.rocket.chat/room?host=open.rocket.chat&rid=GENERAL&path=channel/g
```

Testing

Requirements

- Google Chrome Browser
-

Getting Started

Start Meteor

Run meteor with the command below:

```
TEST_MODE=true meteor
```

To run the tests, the server **must** be started with the environment variable `TEST_MODE=true`. This will set all animations' duration to 0 and create an extra admin user with the login values:

```
1 _id: "rocketchat.internal.admin.test"
2 name: "RocketChat Internal Admin Test"
3 username: "rocketchat.internal.admin.test"
4 emails: "rocketchat.internal.admin.test@rocket.chat"
5 password: "rocketchat.internal.admin.test"
```

Run Tests

On another terminal window, run the test with the command bellow:

```
meteor npm run test
```

Deprecation

The methods and endpoints of the **Realtime API** and **Rest API**, respectively, that have been deprecated are listed below.

Realtime API

Method	Release deprecated	Release removed
cleanChannelHistory	0.64.0	0.67.0

REST API

Endpoint	Release deprecated	Release removed
/user.roles	0.63.0	0.66.0
/channels.cleanHistory	0.64.0	0.67.0
/permissions	0.66.0	0.69.0
/permissions.list	0.73.0	1.11.0
/emoji.custom	1.0.0	1.12.0
/v1/info	1.0.0	1.12.0

Embedded Layout

If you are embedding Rocket.Chat in your site, you can change Rocket.Chat's layout by adding `?layout=embedded` after your server URL.

Example <https://open.rocket.chat/channel/general?layout=embedded>

Adding this to the URL will change the layout of Rocket.Chat to a more "simplistic" view, hiding the left side bar with the channel lists and account management buttons. Additionally, you can enable the top navbar in the embedded layout by going to "Administration -> Layout -> Interface -> Show top navbar in embedded layout".

Setting a URL to embedded mode will "lock" that view in a single channel (since the user will not have access to the channel lists).

Embedded layout works wonders when using with [Iframe integration](#) since you can already login the user using iframe and only show a desired channel embedded to a page.

You can test this feature by going to <https://open.rocket.chat> entering a channel and adding `?layout=embedded` to the URL.

Note: If you want to stop users from accessing other channels, embedded layout alone will not work, since the user could change the URL of the embedded view via browser tools. For that you should edit the user permissions so they can't see other channels

Iframe Integration

Use your own login page to login users in Rocket.Chat

If you want to authenticate users using your own login page in place of the Rocket.Chat's login page via the iframe integration, go to `Administration > Accounts > Iframe` and enable it:

- [How to use iframe integration for authentication](#)
-

Use Rocket.Chat in your site/app inside an iframe

If you need listen to events, go to `Administration > General > Iframe Integration` and enable send (events) or receive (commands), depending on your needs:

- [How to use iframe integration events](#)
- [How to use iframe integration commands](#)
- [Using embedded layout mode](#)

What is iframe auth?

With `iframe` auth you can use your own authentication page/API to log in users on Rocket.Chat.

When enabled Rocket.Chat first do an `XMLHttpRequest` to the `iFrame API` URL trying to see if the user is already logged in at the third party website. If that doesn't succeed then Rocket.Chat will present the `Iframe URL` within an `iframe`, so the user logs in on the third party website which means he is authenticated on Rocket.Chat as well.

We have developed an example app written in NodeJS in order to help you understanding this authentication flow. Please take a look at [iFrame Auth Example](#)

Configuring

API URL and API Method

Configure how Rocket.Chat will call the third party system to either login or to verify if the user is already logged in, by setting `API URL` and `API Method` fields.

`API URL` refers to endpoint on the third-party system that will check if the user is already logged in to that system. The `API Method` is used to select the submission method Rocket.Chat will use to submit information to the `API URL` (for instance using `POST`).

If the user has already logged into the third-party system, the `API URL` should communicate to Rocket.Chat and return a JSON object containing either a `token` or `loginToken` property, otherwise (if the user is not already logged in) the `API URL` should return an empty body with status `401`.

The choice of which property `API URL` will return depends on how the third-party system decides to interface back with Rocket.Chat, as described in one of the two ways below:

Using Rocket.Chat API

If you have the user's password stored (or it is the same between your third party system and Rocket.Chat), you can use [Rocket.Chat's REST APIs](#) to log in the user, this way you will get an `authToken` back from Rocket.Chat that should be returned as `loginToken` by your endpoint.

At this point, if the user does not have a Rocket.Chat account yet, you can either use Rocket.Chat API to [create an user](#) using a admin account or [register him](#).

After you log the user in, you should return a payload like the following:

```
1  {
2    "loginToken": "already-saved-or-returned-login-token"
3 }
```

Managing MongoDB directly

In the case you have access to Rocket.Chat's database, you can connect there directly and manage the user record by yourself. This might be useful if you have MongoDB on your stack already and don't want to learn Rocket.Chat's API.

To do so the endpoint should connect on Rocket.Chat's MongoDB database and make sure the `generated-token` is saved on `users` collection on the corresponding user record. The `generated-token` should be saved on the field path `servicesiframe.token`. This is how the user record should look like:

```
1  {
2    "_id": "MZiFvWAf96876875u",
3    "createdAt": new Date(1432252673528),
4    "services": {
5      "iframe": {
6        "token": "generated-token"
7      }
8    },
9    "emails": [
10      {
11        "address": "useremail@gmail.com",
12        "verified": true
13      }
14    ]
15  }
```

```
14 ],
15 "name": "John Doe",
16 "username": "john.doe",
17 "active": true,
18 "statusDefault": "online",
19 "roles": [
20   "user"
21 ],
22 "type": "user"
23 }
```

On this case, the response should be:

```
1 {
2   "token": "generated-token"
3 }
```

IFrame URL

The URL of the page you want to show as the login page of your Rocket.Chat instance (this page can be created in any programming language and/or web framework).

The login page will then communicate back to Rocket.Chat using `postMessage` API.

After user logs in, you have to authenticate him on Rocket.Chat side, pretty much the same as you did before on `API URL` endpoint, but now you should return a JavaScript code that will be rendered within the `iframe`, depending how you logged in the user:

- If have used Rocket.Chat's APIs to log in the user or already have user's token saved in your end, return:

```
1 <script>
2 window.parent.postMessage({
3   event: 'login-with-token',
4   loginToken: 'your-token'
5 }, 'http://your.rocket.chat.url');
6 </script>
```

-
- If you have saved user's token connecting directly to Rocket.Chat's database on the user's field `servicesiframe.token` :

```
1 <script>
2 window.parent.postMessage({
3   event: 'try-iframe-login'
4 }, 'http://your.rocket.chat.url');
5 </script>
```

Using OAuth configured on Rocket.Chat's end

If you have OAuth services configured on Rocket.Chat, you can trigger them from within your login page as well.

To implement this authentication, after triggering the OAuth authentication you will receive a `postMessage` back from Rocket.Chat with user's credentials response from OAuth service. You need to manage the user creation/authentication on Rocket.Chat's database by yourself, the same as described earlier.

Facebook

```
1 window.parent.postMessage({
2   event: 'call-facebook-login',
3   permissions: ['email']
4 }, 'http://your.rocket.chat.url');
```

The reply will be a `postMessage` back to your page with:

```
1 {
2   event: 'facebook-login-success',
3   response: {
4     // authResponse: Object
5     // accessToken: "a7s6d8a76s8d7..."
6     // expiresIn: "5172793"
```

```
7      // secret: "..."
8      // session_key: true
9      // sig: "..."
10     // userID: "675676576"
11     // status: "connected"
12   }
13 }
```

Or an error

```
1  {
2    event: 'facebook-login-error',
3    error: error,
4    response: response
5 }
```

Google

```
1 window.parent.postMessage({
2   event: 'call-google-login',
3   // scopes:
4   // webClientId:
5 }, 'http://your.rocket.chat.url');
```

The reply will be a postMessage back to your page with:

```
1  {
2    event: 'google-login-success',
3    response: {
4      // "email": "rodrigoknascimento@gmail.com",
5      // "userId": "1082039180239",
6      // "displayName": "Rodrigo Nascimento",
7      // "gender": "male",
8      // "imageUrl": "https://lh5.googleusercontent.com/-shUpniJA480/AAAAAAA",
9      // "givenName": "Rodrigo",
10     // "familyName": "Nascimento",
11     // "ageRangeMin": 21,
12     // "oAuthToken": "123198273kajhsdh1892h"
```

```
13      }
14  }
```

Or an error

```
1  {
2    event: 'google-login-error',
3    error: error
4 }
```

Twitter

```
1 window.parent.postMessage({
2   event: 'call-twitter-login'
3 }, 'http://your.rocket.chat.url');
```

The reply will be a postMessage back to your page with:

```
1  {
2    event: 'twitter-login-success',
3    response: {
4      // "userName": "orodrigok",
5      // "userId": 293123,
6      // "secret": "asdua09sud",
7      // "token": "2jh3k1j2h3"
8    }
9  }
```

Or an error

```
1  {
2    event: 'twitter-login-error',
3    error: error
4 }
```

How to login in Rocket.Chat with default account system while in development

When you activate the IFrame auth you will not be able to access Rocket.Chat's default login page, however if still need/want to use your Rocket.Chat's credentials to log in, you can do that by opening the browser's Developer Console and executing the following code:

```
Meteor.loginWithPassword('username-or-email', 'your-password');
```

Testing the iFrame Authentication

This article will describe how to test the iFrame authentication, using a test tool developed by Rocket.Chat team. This test scenario is based in a localhost environment.

Step 1

Download and start the *iframe-auth-example* test service following the steps mentioned [here](#)

The test service code can be used as an example on how to setup the iFrame auth calls to Rocket.Chat. Code is available [here](#).

Step 2

Configure Rocket.Chat server (*Administration > Accounts > iFrame*) as follows:

Iframe



Enabled



Iframe URL



`http://localhost:3030/login`

API URL



`http://localhost:3030/sso`

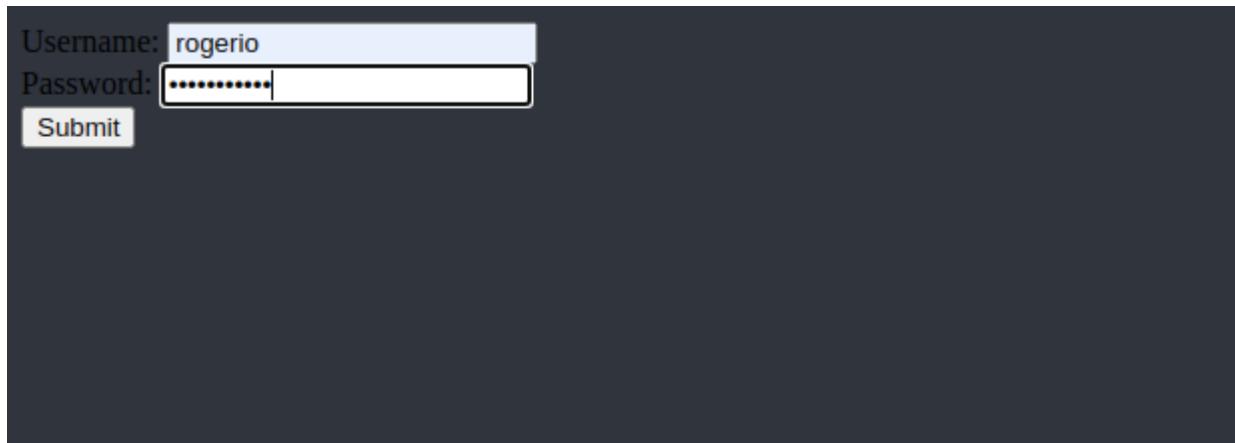
Api Method

POST

[Reset Section Settings](#)

Step 3

Test the iFrame service by calling Rocket.Chat login URL (in this example,
[http://localhost:3000\](http://localhost:3000/))



In this moment, the *iframe-auth-example* service runs and the iframe authentication calls are executed.

By default, the test service code expects the login to be done with the following credentials (harcoded)

```
username: new-user  
password: new-users-passw0rd
```

Change the `currentUsername` (line 105) to `true` so you can login again with the same user. By changing the code you may use any user which already exists in Rocket.Chat

Further reference can be found [here](#).

Iframe Events

Events sent

Here is a list of events triggered from Rocket.Chat when the iframe integration is enabled:

Event name	Description
notification	Fired when a user receives a notification
unread-changed-by-subscription	Fired each time a user's subscription record changes (i.e.: unread counts, etc)
unread-changed	Fired when the pages title changes
room-opened	Fired when a room is opened
new-message	Fired every time the opened room receives a new message
click-user-card-message	Fired when the user clicks on a username link
click-mention-link	Fired when the user clicks on a mention link
click-message-link	Fired when the user clicks on a posted link
click-action-link	Fired when the user clicks on an action link button (i.e.: "click to join" to video conferences)

Below there is a sample code that listens to events fired from Rocket.Chat opened on an `iframe`:

```
1 window.addEventListener('message', function(e) {  
2     console.log(e.data.eventName); // event name  
3     console.log(e.data.data); // event data  
4});
```

Iframe integration: Sending commands

Available commands

Command	Params	Description
go	- path string	Change url
login-with-token	- token string	Allow login with token
call-custom-oauth-login	- service string	Allow login via oauth methods
set-user-status	- status string	Set the status of the user
logout		Log the user out of their current session

Example

```
1 document.querySelector('iframe').contentWindow.postMessage({  
2   externalCommand: 'go',  
3   path: '/admin/General'  
4 }, '*')
```

Adding a Rocket.Chat Chat Room to your Web App

Requirements

You start with the creation of the Rocket.Chat server and then move on to the integration. A demo app is used to show how to do it concretely.

To get started you need at least one room and a running Rocket.Chat server.

In addition to adding a chat room to your web app, this article demonstrates :

- How Rocket.Chat can identify a user through the iframe and therefore handle a login token
- How Rocket.Chat can use an endpoint in your application to identify/create a user
- Use cases with commands and events provided by the RC code once the room has been installed.

Create a Rocket.Chat server

There are many ways to create a Rocket.Chat server. A very quick way to do this is via snaps. To start a [server via snaps](#) on Ubuntu, type in your terminal:

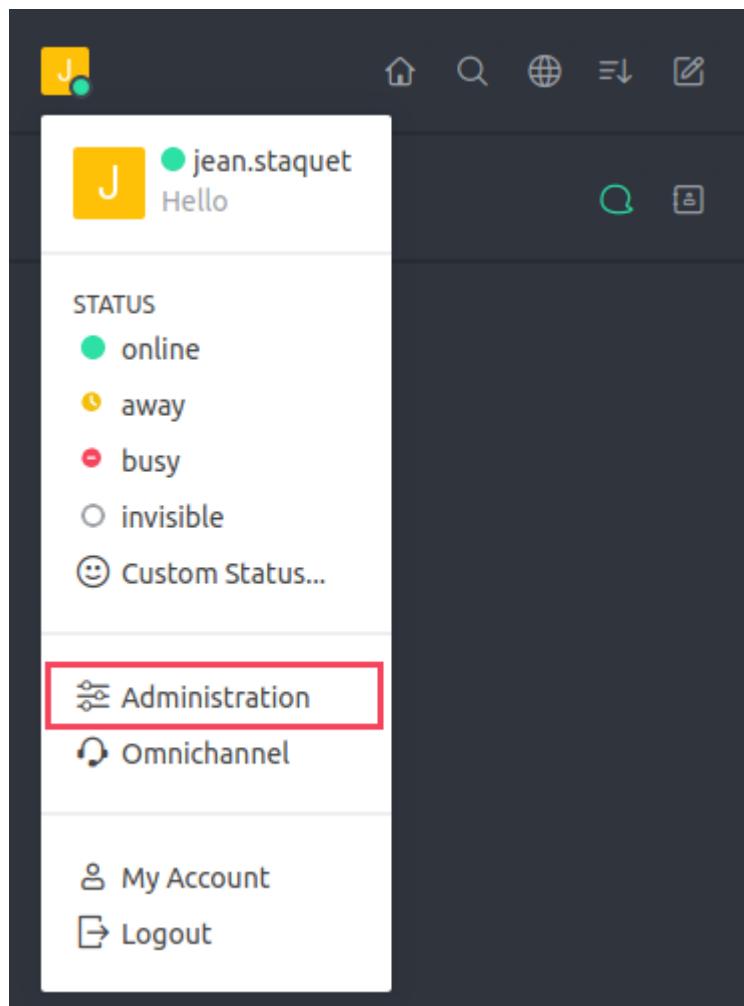
```
sudo snap install rocketchat-server
```

Then go to <http://localhost:3000> and add the information & settings to launch the Rocket.Chat server.

Part I: Activate the Iframe integration

To get started, you need to enable the iframe integration setting in your Rocket.Chat server. To do so:

1. Click **Administration**



2. Go to **Settings** and click **Account**:

Administration **Development** ×

Info

↳ Integrations

- OAuth Apps
- Mailer
- Custom User Status
- Permissions

Settings

Search Q

Accounts

- Analytics
- Assets
- Atlassian Crowd
- Blockstack
- Bots
- CAS
- Custom Emoji Filesystem
- Custom Sounds Filesystem
- Discussion
- E2E Encryption
- Email
- Enterprise

Deployment

- Version 3.15.0-develop
- Deployment ID eNCGCXHjai4Cnydjb
- Apps Engine Version 1.25.0
- Node Version v12.22.1
- Database Migration 223 (May 9, 2021 11:05 AM)
- MongoDB 4.2.8 /wiredTiger (oplog Enabled)
- Commit Details HEAD: (2d6b1197a)
Branch: AddingLabels
- PID 3688

License

Development

- Features
 - OmniChannel
 - Auditing
 - Canned responses
 - Engagement Dashboard

Usage

Users

	Total	202
Online	2	
Busy	0	
Away	0	
Offline	200	

Types and Distribution

	Connected	2
Activated Users	196	
Activated Guests	0	
Deactivated Users	6	
Rocket.Chat App Users	0	

Rooms

	Total Rooms	103
Total Channels	33	
Total Private Groups	45	
Total Direct Message Rooms	25	
Total Discussions	9	
Total OmniChannel Rooms	0	

Messages

	Total Messages	490
Total Threads	3	
Total Messages in Channels	412	
Total Messages in Private Groups	54	
Total Messages in Direct Messages	24	
Total Messages in OmniChannel	0	

Uploads

	Total Uploads	1
Total Uploads Size	13.56 kB	

[See on Engagement Dashboard](#)

[Apply Offline License](#)

3. In account scroll to the iframe menu and click on the button to enable iframe integration.

This will allow you to integrate iframes from your Rocket.chat server on other websites/web apps

Administration **Development** ×

Accounts

Iframe

Enabled

Iframe URL

API URL

Api Method POST

[Reset Section to Default](#)

Failed Login Attempts

Login Logs

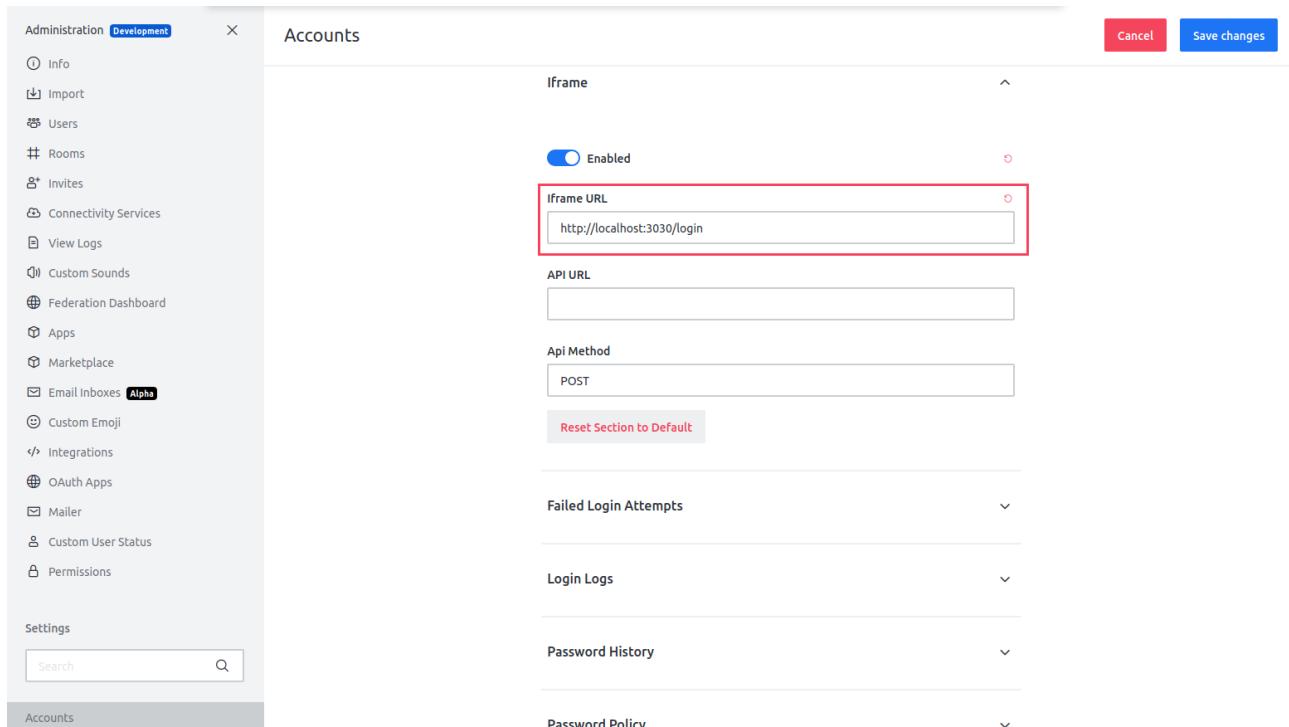
Password History

[Discard Changes](#)

[Cancel](#) [Save changes](#)

Now that iframe integration is enabled in your Rocket.Chat server, we can connect the server to your application.

4. Go to the iframe URL and type in the endpoint API to add an application. Here we type `http://localhost:3030/login`. Here you need to put the URL of the application where you want to integrate the current Rocket.Chat application that you have. We used `localhost:3030/login` as we are running an application on this port for a demo later in the guide.



The screenshot shows the Rocket.Chat Administration interface with the 'Development' tab selected. On the left, there's a sidebar with various settings like Info, Import, Users, Rooms, Invites, Connectivity Services, View Logs, Custom Sounds, Federation Dashboard, Apps, Marketplace, Email Inboxes (Alpha), Custom Emoji, Integrations, OAuth Apps, Mailer, Custom User Status, and Permissions. Below that is a 'Settings' section with a search bar. The 'Accounts' tab is currently active. The main content area has a 'Iframe' tab selected. Under the 'Iframe' tab, there's a toggle switch labeled 'Enabled' which is turned on. Below it is a 'Iframe URL' input field containing 'http://localhost:3030/login', which is also highlighted with a red box. There are other fields for 'API URL' and 'Api Method' (set to 'POST'), and a 'Reset Section to Default' button. Below the 'Iframe' tab, there are sections for 'Failed Login Attempts', 'Login Logs', 'Password History', and 'Password Policy', each with a collapse/expand arrow.

5. The second URL we put in is the API URL. It is used so that Rocket.Chat can verify that a user is logged into our application.

When a user from your website/web app goes to a web page containing the Rockat.chat Room iframe, the Rocket.Chat server will use this API endpoint to verify:

1. The user is identified on your web app
2. They use the credentials provided to attempt to login into the Rocket.chat server

The screenshot shows the 'Accounts' settings page in the Rocket.Chat administration interface. The 'Iframe' section is active, featuring an 'Enabled' toggle switch (set to 'Enabled'), an 'Iframe URL' input field containing 'http://localhost:3030/login', an 'API URL' input field containing 'http://localhost:3030/sso' (which is highlighted with a red box), an 'Api Method' input field containing 'POST', and a 'Reset Section to Default' button.

6. Next, you need to enable the Restrict access inside any Iframe restriction to allow only your application's URL to use the Iframe.

! Disabling this feature is a security hole in your Rocket.Chat server and will allow anyone to load the iframe from your server into their app.

To do so:

1. Go to **Settings >> General** and disable **Restrict access inside any Iframe**

The screenshot shows the 'General' settings page in the Rocket.Chat administration interface. The 'Restrict access inside any Iframe' checkbox is highlighted with a red box and has a tooltip stating: 'This setting enable/disable restrictions to load the RC inside any iframe'. Other visible settings include 'Document Domain', 'Language', 'Allow Invalid Self-Signed Certs', 'Options to X-Frame-Options' (with a note: 'Options to X-Frame-Options. You can see all the options here.'), and 'Enable Favorite Rooms'.

7. Hit Save Changes

Part II: iframe authentication

In order to use the iframe, the user must have an account and be identified. In this scenario, there are two possibilities:

1. The user already has a Rocket.Chat account
2. The user does not have a Rocket.Chat account

It is explained down below how to manage these two cases and make sure that the Rocket.Chat server can identify the user of the iframe.

You can test the iframe authentication using an example provided by the Rocket.Chat [here](#).

We will explain the use of the iframe authentication with this demo code, but you are free to do it directly in your application in parallel.

Steps to use and install the demo code:

1. Clone the code: <https://github.com/RocketChat/iframe-auth-example>

```
git clone https://github.com/RocketChat/iframe-auth-example.git
```

2. Go to the directory of iframe-auth-examples code by typing on your console and open your IDE there:

```
cd iframe-auth-examples/
```

3. Install the demo packages, type in your console:

```
npm install
```



Do not launch the demo app directly yet because we will modify it a bit!

Part III Modifying the demo code:

Here we will see how to manage to add the iframe to our code. We'll show you how to do this by modifying the demo.

1. Delete from line 129-136 inclusive and replace with:

```
res.redirect("/home")
```

We do this because we will redirect the user to our home page containing the iframe once he pressed the login button.

2. On line 89 add:

```
app.get('/home', function (req, res) {  
  
  res.set('Content-Type', 'text/html');  
  
  fs.createReadStream('/home.html').pipe(res);  
  
});
```

3. Add a page in ./ called home.html containing this code:

```
<!DOCTYPE html>  
  
<html lang="en">  
  
<head>  
  
<meta charset="UTF-8">
```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body style="text-align: center;">

<button style="margin-bottom: auto;" onclick={x()}>GO TO GENERAL</button>

<iframe src="http://localhost:3000/channel/general/?layout=embedded"
title="myframe"></iframe>

</body>

<script>

function x() {

    document.querySelector("iframe").contentWindow.postMessage(
        {
            externalCommand: "go",
            path: "/channel/general/?layout-embedded"
        },
        "http://localhost:3000"
    );
}

}
```

```
</script>
```

```
</html>
```

So now that a new user is created, they will be redirected to our home.html page.

-  In the Rocket.Chat iframe configuration, we have left POST in the API method. This needs to be tuned with the request methods in our backend for the link we put in the API URL.

It is demo backend for handling the Rocket.Chat server's /sso POST request. The /sso code in our backend will run once the user has been on the page containing the iframe, in this case, it is home.html.

4. For the demo we will change line 25 to :

```
var notLoggedIn = false;
```

as the user will be logged in as they arrive at this page.

5. And the line 40:

```
var currentUsername = true;
```

Because the user will already be logged in. As a reminder, this only serves the demo code.

So in this part of the code with /sso, we're going to verify that the user is logged into Rocket.Chat. The code from line 68 onwards shows us that to verify this, we make a post request to the Rocket.Chat server API with the user's credentials.

-  The login credentials must match those previously entered by the user. For the demo we will put them back as we created them.

6. Line 69 and 70, we will put:

```
username: 'new-user',  
password: 'new-users-passw0rd'
```

They were hardcoded for the demo.

After these modifications, the code is ready for the demonstration.

Type in your console:

```
npm run
```

This will launch the demo app.

Let's dive a little bit into the code.

The demo application is currently running on channel 3030. There is the index.js file, that's the backend of our demo. And we also have login.html which is the front-end of the demo. Go to the index.js file and look for the get request code on line 84. We see that this code renders the login.html file. In the index.js file, let's take a closer look at the content of the POST /login and POST /sso endpoints which are very important for the iframe integration.

In the code, we see that, after receiving the credentials via the frontend form, we need to pass them through the auth logic of the demo. Here there is no code because it is in a demo, but in your web app, you will certainly have code to manage the auth of your application.

Once passed into the demo/web app auth logic, the credentials are used to do two things:

- Either they are used to connect to the Rocket.Chat app via the Rocket.Chat server endpoint /login (you can copy and paste the demo code into your own app)
- If Rocket.Chat has not found a user, it creates one with the identifiers received by your application

The home.html file has two important points:

1. The button and the function of the button: We will explain this later in the fast-loading part when we change channels.
2. The iframe: When the user arrives on the page the iframe will load the src of the iframe will call the URL to Rocket.Chat server. After that the Rocket.Chat code will make a request to the sso endpoint we defined earlier. So to:

<http://localhost:3030/sso>

The code does this to check if the user is logged in or not. This will be handled by the demo backend.

Once successfully identified Rocket.Chat's code will store a token in its DB to say that the user has logged in (as usual) and it will also store the user's credentials in the user's browser local storage.

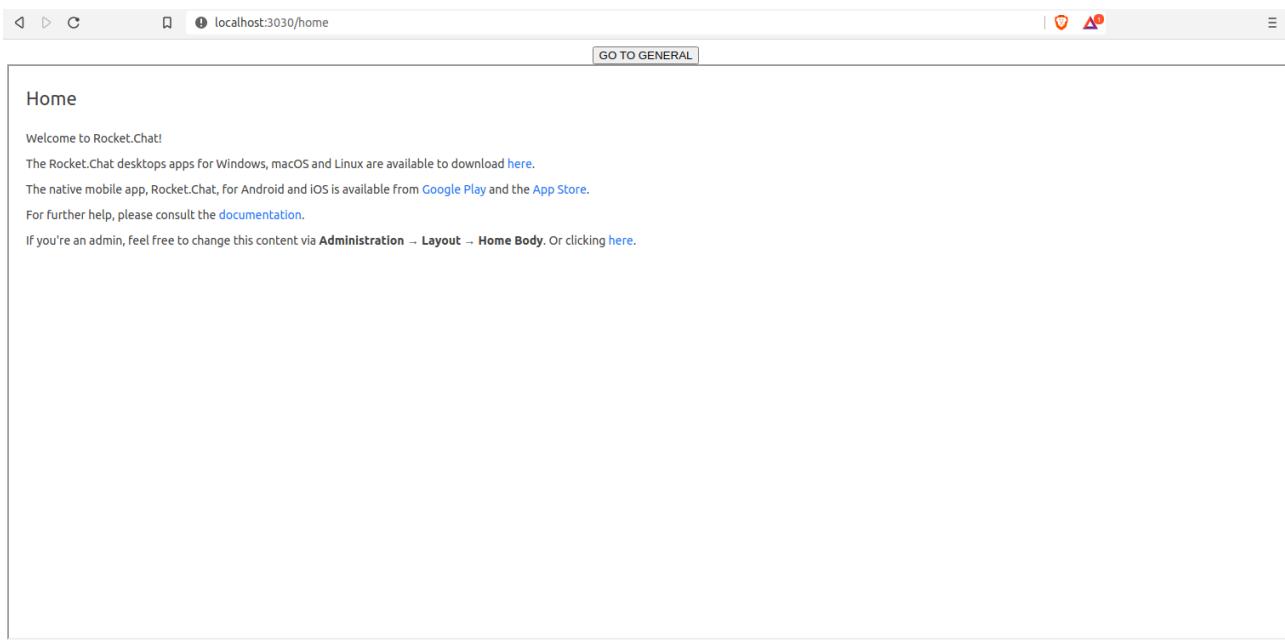
So now go to localhost:3030/login and enter any credentials and press Submit:

-  You have to go to a browser in private mode so that the iframe doesn't take your identifiers that are in your local storage, otherwise you will connect with your own account instead of the one that has just been created by our demo app.

Username:

Password:

After doing all this, you should have the home view in your web app, as shown below:



That's a great first step. In the next part, we'll see how to enable fast-loading to switch between rooms.

Part 4: Enabling fast-loading for the embedded RC room

- i Rocket.Chat is a client side rendered application, so when the user arrives on Rocket.Chat, the whole application loads in their browser, avoiding loading waits when the user changes room for example.

Rocket.Chat is a client-side rendered application, so when the user arrives on Rocket.Chat, the whole application loads in their browser, avoiding loading waits when the user changes

room for example.

The problem is that this is not the case in the iframe. So if the user switches rooms in the iframe, there will be latency. We want to remove this latency and make the iframe load completely when the user goes to the page.

Rocket.Chat provides commands for iframe integration. In these commands, there is `postMessage()`. Using this command in your client-side code, the iframe contentWindow will client-side route to a chat room thus making the switching instant without any loading.

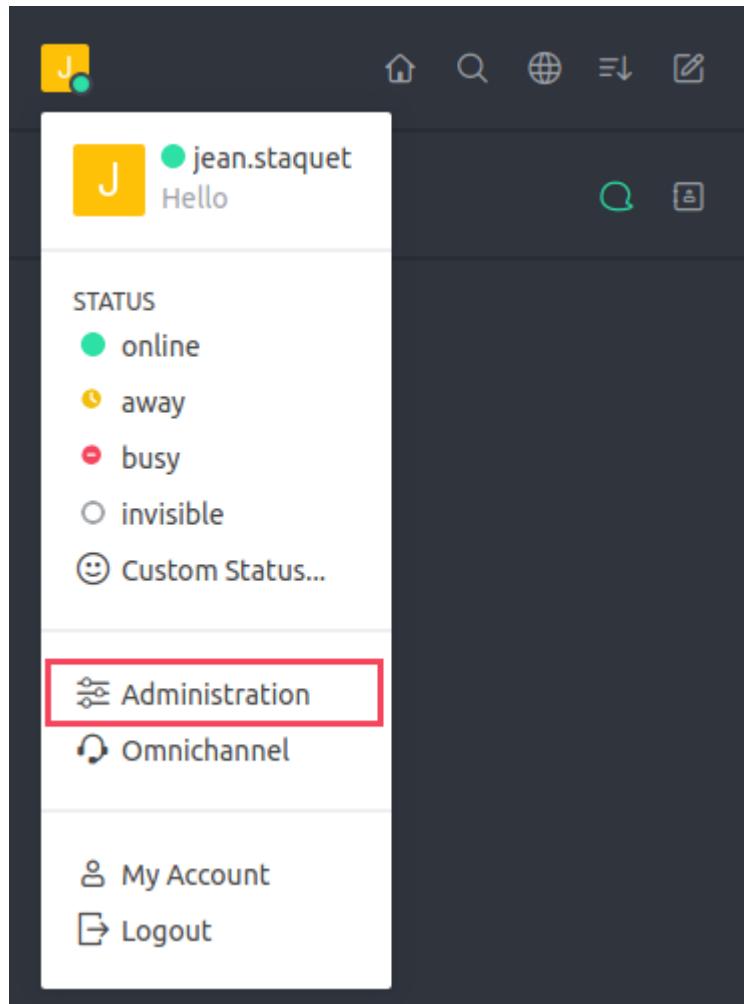


Iframe integration: Sending commands

<https://developer.rocket.chat/guides/developer/iframe-integration/iframe-integration-sending-commands>

In order to use these commands, which we will demonstrate next, you need to change a setting in your Rocket.Chat server. To change it:

1. Go to **Administration**:



2. Go to **Settings > General > iframe Integration:**

In iframe integration, check **Enable Recieve**.

In the **Receive Origin** field, you should put the URL of your web app that will receive the commands. As we are working with the demo, we have put localhost:3030:

Administration **Development** × General Cancel Save changes

Atlassian Crowd
Blockstack
Bots
CAS
Custom Emoji Filesystem
Custom Sounds Filesystem
Discussion
E2E Encryption
Email
Enterprise
Federation
File Upload
General
IRC Federation
Layout
LDAP
LiveStream & Broadcasting
Logs
Message
Meta
Mobile
OAuth

Enable Send
Send events to parent window

Send Target Origin
*
Origin with protocol prefix, which commands are sent to e.g. <https://localhost>, or * to allow sending to anywhere.

Enable Receive
Allow parent window to send commands to Rocket.Chat.

Receive Origins
http://localhost:3030
Origins with protocol prefix, separated by commas, which are allowed to receive commands e.g. <https://localhost>, <http://localhost>, or * to allow receiving from anywhere.

Reset Section to Default

NPS ▾

Once saved, this will allow the iframe's parent window to send commands to the Rocket.Chat server.

Then go back to the code, in home.html.

You can review the function we have hooked. Click **Go To General** button:

```
document.querySelector("iframe").contentWindow.postMessage(  
  {  
    externalCommand: "go",  
    path: "/channel/general/?layout=embedded"  
  },  
  "http://localhost:3000 <Your server URL>"  
);
```

This command will go directly to the general room of the Rocket.chat server.

So now, once permission has been enabled in your server, it allows you to change rooms without loading.

And there you have it, you are now ready to add a chat room to your web app.

However, Rocket.Chat provides more than just the ability to have an embedded room. Indeed, using the application code and server permissions, one can use certain commands (like the go to room we used) and events triggered by the embedded application.

For more details about events and commands go to:



Iframe integration: Sending commands

<https://developer.rocket.chat/guides/developer/iframe-integration/iframe-integration-sending-commands>



Iframe Events

<https://developer.rocket.chat/guides/developer/iframe-integration/iframe-events>

We will see some use cases of how to use them in the next points

Part 5: Uses cases

Use case 1 - Commands: Change your user status

We will demo another example of the commands:

Go to home.html and add the x() function to the tag script:

```
const x = () => {
```

```
document.getElementsByTagName("iframe")[0].contentWindow.postMessage({  
  externalCommand: "set-user-status",  
  status: "away"  
},  
  "http://localhost:3000"  
)  
}
```

Then replace the <button> line in home.html with:

```
<button style="margin-bottom: auto;" onclick={x()}>Set away status</button>
```

Now once you are in the embedded room, you can click on the button and it will change your status.

Use case 2 : Events: Adding code for Rocket.Chat iframe events

Rocket.Chat provides events in the iframe. These events trigger different actions by the user. You can get the full list here:



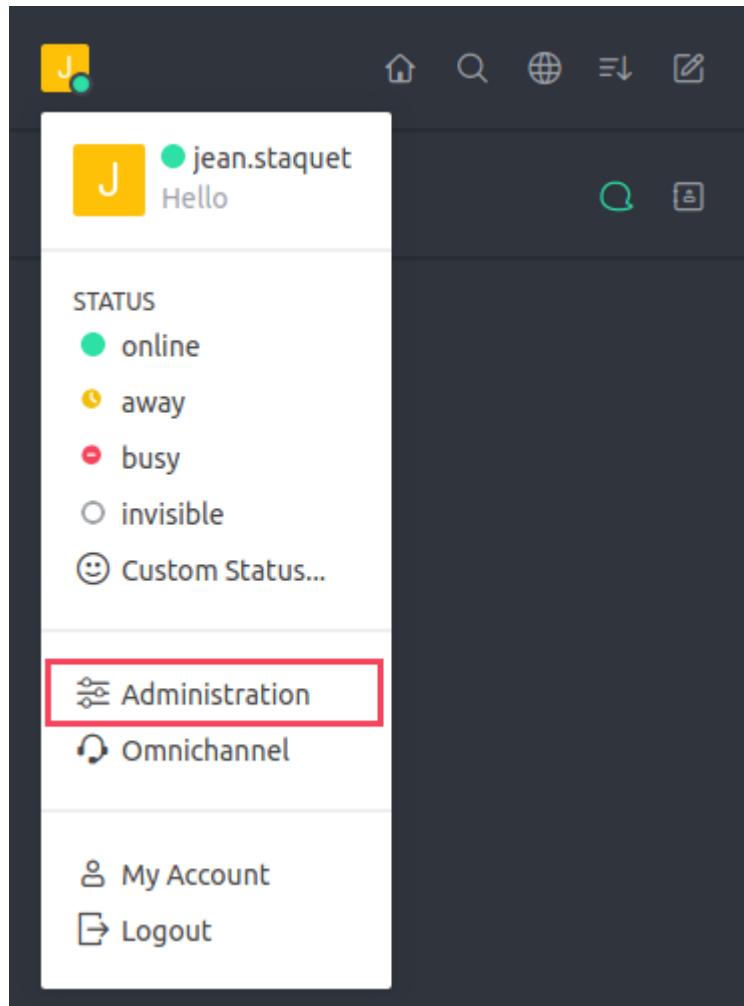
Iframe Events

<https://developer.rocket.chat/guides/developer/iframe-integration/iframe-events>

However, we'll do a use case with an event so you can see how it works:

First, to enable the sending of iframe events, you need to enable the feature in your Rocket.Chat app:

1. Go to **Administration**



2. Go to **Settings > General > Iframe Integration:**

3. Toggle switch **Enable Send** on.

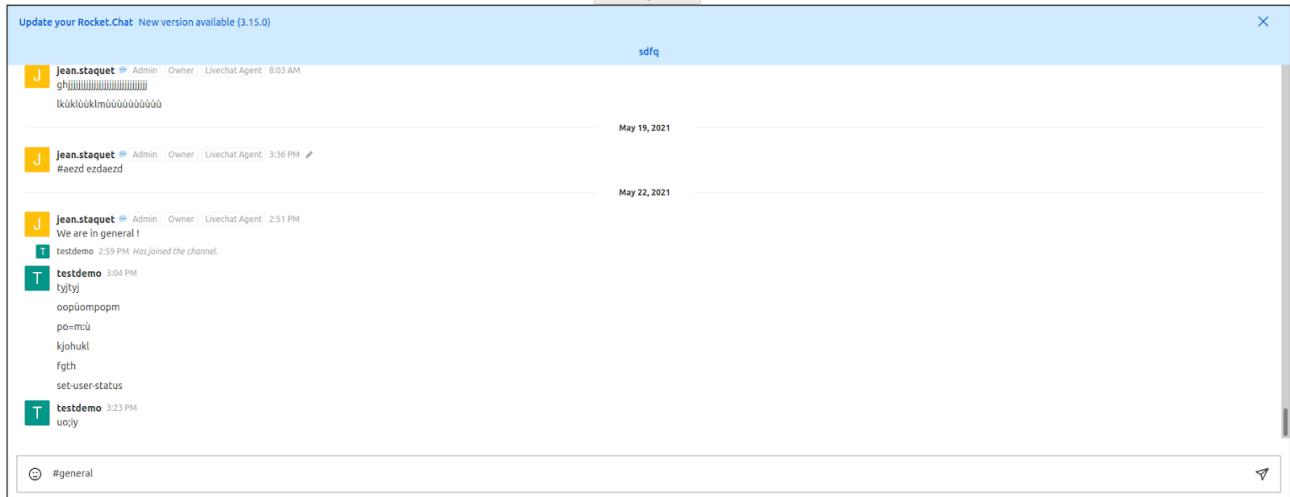
A screenshot of the Rocket.Chat administration interface under the "General" tab. On the left is a sidebar with various settings categories like Accounts, Analytics, Assets, etc. The "General" category is currently selected and highlighted in grey. The main content area is titled "General" and contains a "Development" tab at the top right. Below the tabs is a search bar. The "Iframe Integration" section is expanded, showing two toggle switches: "Enable Send" (which is turned on) and "Enable Receive". There is also a "Send Target Origin" input field containing an asterisk (*) and a note about protocol prefixes. The "Receive Origins" input field contains "http://localhost:3030" and a note about receiving commands from specific origins. At the bottom right are "Cancel" and "Save changes" buttons.

This is to allow your Rocket.Chat server to send events to the parent window of the iframe. In the **Send Target Origin** field, you can put a * or the URL of your application. Click **Save Changes**

3. In home.html in the tag script, add:

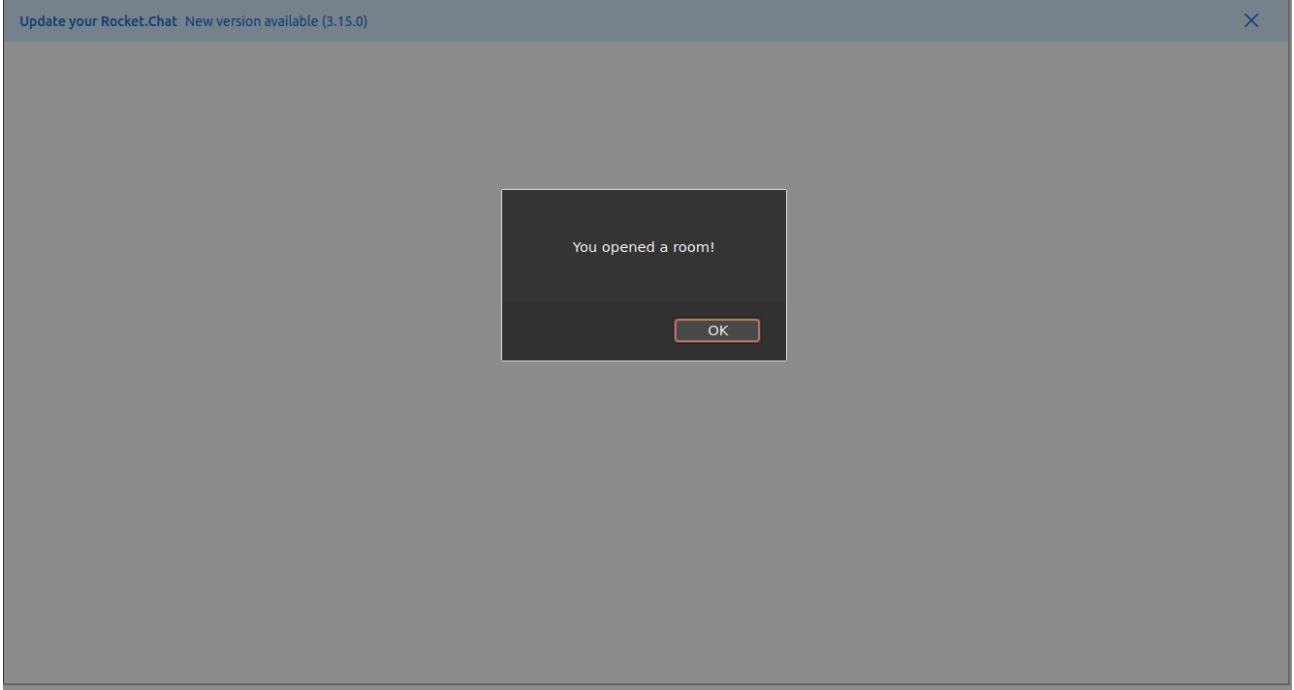
```
window.addEventListener('message', function(e) {  
    if(e.data.eventName === 'room-opened') {  
        alert('You opened a room!')  
    }  
};
```

Go in your embedded room, and type #<ARoomName> in the message field:



Hit Enter.

Now, if you click on the link and thus open a new room, an alert should appear on your screen:



Use Case 3 - Enable / Disable embedded layout

Currently, we see that the application is in embedded mode. That is, we only see the room we are in and the left sidebar and room header are not present.

However, we can make the user see your whole Rocket.Chat application in your web app.

To do this you need to change a line in the iframe tag to enable or disable the embedded layout:

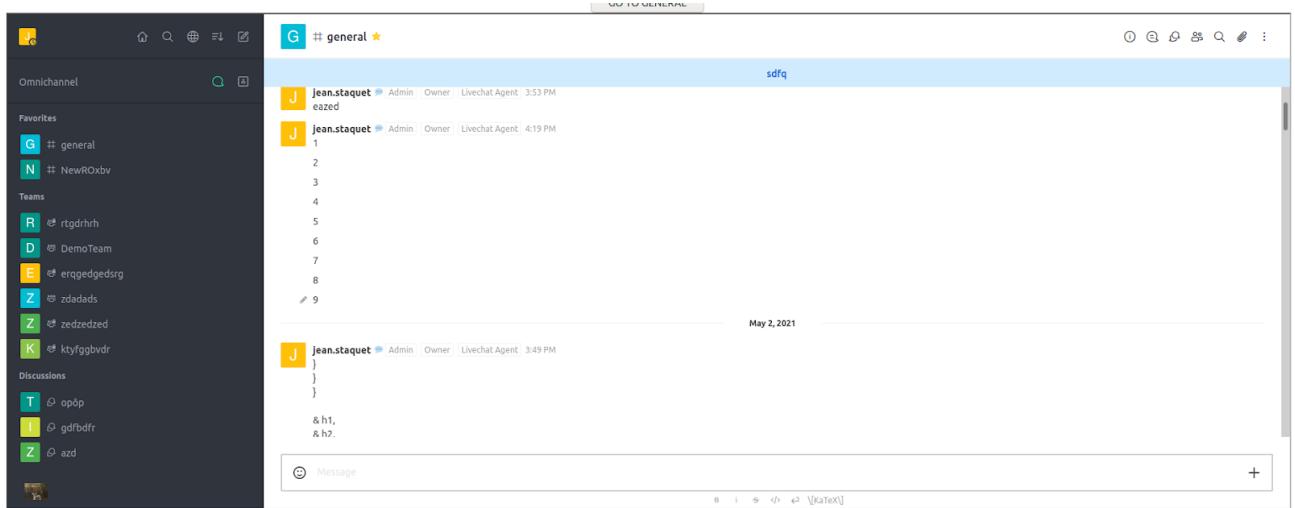
Enable embedded layout:

```
<iframe style="width: 100%; height: 80vh;"  
src="http://localhost:3000/channel/ChannelTest1/?layout=embedded"  
title="myframe"></iframe>
```

Disable embedded layout:

```
<iframe style="width: 100%; height: 80vh;"  
src="http://localhost:3000/channel/ChannelTest1" title="myframe"></iframe>
```

The result after deactivating it will be like this :



Two Factor Authentication

Rocket.Chat uses Two Factor to authorize important actions. There is a list of possible sources for the two-factor code:

- **Authenticator App:** like Google Authenticator or Authy (need to be configured by the user);
- **Email:** users will receive the code via email (configured by default and enabled for those users with verified emails);
- **Password***: it's not a two factor by itself, but a fallback for the cases where the user has no other option to configure;

* The password fallback is disabled for the login process, so the login will not require the password twice when the user has no other two factor method configured.

Any **DDP Method** or **REST call** may have the two-factor requirement; for that reason, we suggest creating a wrapper for your calls to handle the errors described here and executing the request again, passing the required info as we will describe here as well.

Remember Me

By default, after a two-factor validation, the client used (a hash of user-agent + IP address) will be trusted for 5 minutes. It's configurable via the admin panel.

Some methods may disable this feature forcing the API to always require the two-factor for that method/endpoint. The method to disable the two-factor by email and the login are examples.

Compatibility

We are using the error `totp-required` for compatibility purposes, it doesn't mean that the error is related to TOTP only, so we pass more details to identify the action required.

Personal Access Tokens

Personal Access Tokens are tokens created by the users (when enabled by the server) commonly used to give access to other applications, bots, etc. Those tokens do not expire, and they have the option to **bypass** the Two-Factor (required by default), allowing users to use their integrations without restrictions when needed.

Now it's the two-factor required to create personal access tokens.

The bypass should be used carefully because it gives super powers to who gain access to the token

Realtime API

Visit the [Two Factor Authentication](#) page at Realtime API guides for more information.

REST API

Visit the [Two Factor Authentication](#) page at REST API guides for more information.

Apps Dev Guides

→ WhatsApp Integration

/guides/developer/apps-dev-guides/whatsapp-integration

WhatsApp Integration

Getting started with Template messages

The WhatsApp app allows you to send template messages to a target user.

In order to send a template message, you have 2 options

1. Via the endpoint

- REST API Documentation for this endpoint can be found [here](#)

2. Via the Slash Command and UI

- You can send the template message in a more user-friendly way using the GUI form.
- To trigger the GUI, open a WhatsApp contact room and run the following command

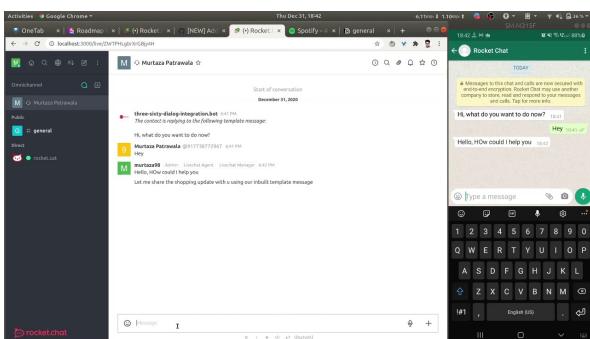
```
/whatsapp send-template
```

- This will open up a modal, where you can select

- The `Template` you wish to send,
- The `Language` you wish to send the template in,
- And also substitute `parameters` within the message, if present.

Note, If parameters (i.e eg) are present withing the template message, please make sure that you substitute all of them before sending the message. Failure to do so will result in an error.

A video demonstration of the above feature is available here:



Getting status of messages sent

To receive real-time status updates for outbound messages, you can choose to set a *Message Status Endpoint URL*. The app will send a request to this URL each time your message status changes to one of the following: queued, failed, sent, delivered, read, or if some error occurs.

You can set the *Message Status Endpoint URL* in the Apps Settings.

Following is the payload format you will receive on the endpoint

Field Name	Type	Description
msId	String or undefined	Message Id.
message	String	Text message
mobileNo	String	Phone number of WhatsApp user to which you had sent the message
connectedWhatsAppNo	String	Your connected WhatsApp Number from which you had sent the message to user
MessageStatus	String	Status of message - sent, delivered, read, failed, deleted
ErrorDetails	Optional Type: String or WhatsApp error format	Errors from template messages will be shown here
type	String	Template-Message or Regular-Message
timestamp	String	Timestamp of request

Troubleshooting

1. babel-runtime:

If you are having the following error:

```
1 (STDERR) Error: The babel-runtime npm package could not be found in your n
2 (STDERR) directory. Please run the following command to install it:
3 (STDERR)
4 (STDERR) meteor npm install --save babel-runtime
5 (STDERR)
6 (...)

7 => Exited with code: 1
8 => Your application is crashing. Waiting for file change.
```

Just install the mentioned package with the following command:

```
meteor npm install --save babel-runtime
```

1. bcrypt:

If you see the following warning in the `meteor` logs:

```
1 (STDERR) Note: you are using a pure-JavaScript implementation of bcrypt.
2 (STDERR) While this implementation will work correctly, it is known to be
3 (STDERR) approximately three times slower than the native implementation.
4 (STDERR) In order to use the native implementation instead, run
5 (STDERR)
6 (STDERR) meteor npm install --save bcrypt
7 (STDERR)
```

Don't panic =) It means that the `bcrypt` library is not installed on your system and `meteor` will use a javascript alternative that is about three times slower.

If you want to install the library to make it faster, use the following command:

```
meteor npm install --save bcrypt
```

If the version of the `python` interpreter on your system is **greater than** v2.5.0 or **less than** 3.0.0, it should work fine, but, if you see a message like this:

```
1 gyp ERR! configure error
2 gyp ERR! stack Error: Python executable "/usr/local/bin/python3" is v3.5.2
3 gyp ERR! stack You can pass the --python switch to point to Python >= v2.5
```

After you are sure that you have a `python` interpreter that matches the above requirements, use the following command to fix the error:

```
meteor npm config set python python2.7
```

Build it again:

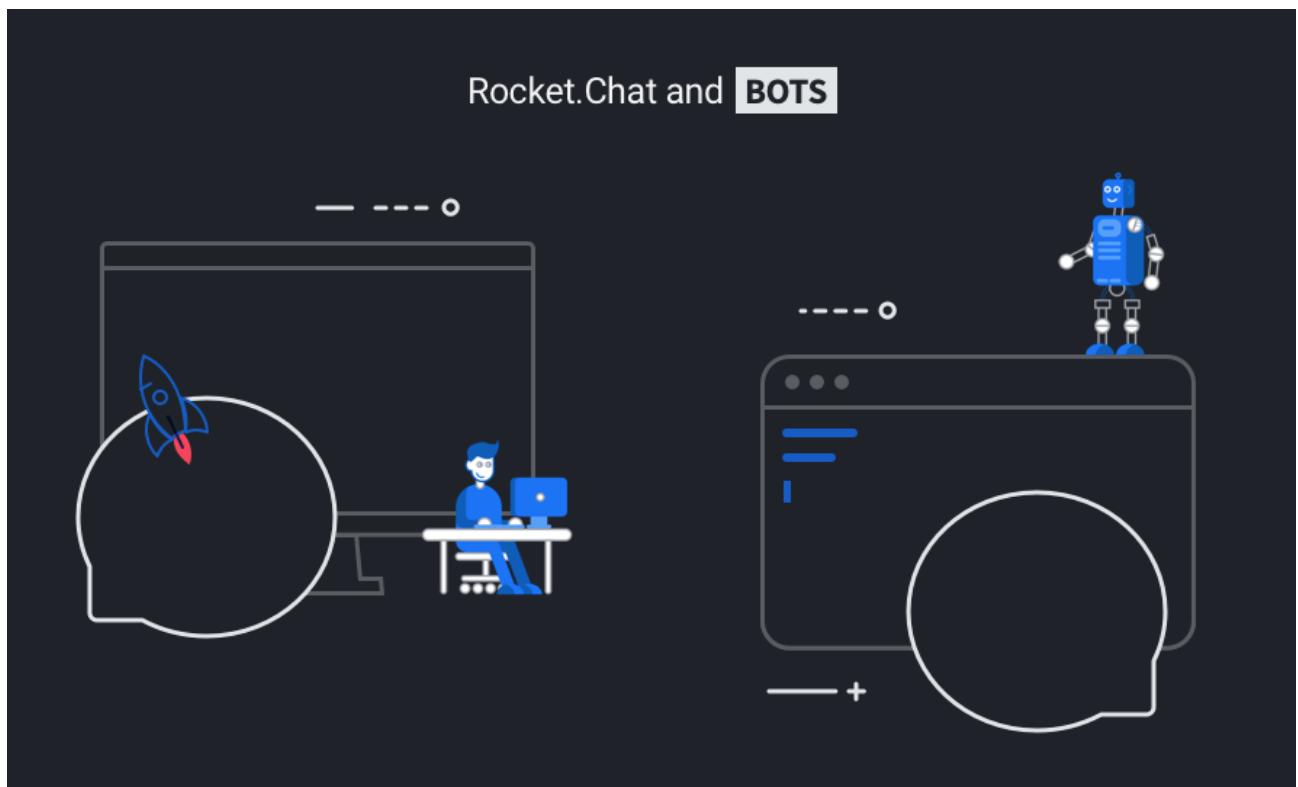
```
meteor npm install --save bcrypt
```

If everything works, you should see a message like this:

```
1 > node-gyp rebuild
2
3 CXX(target) Release/obj.target/bcrypt_lib/src/blowfish.o
4 CXX(target) Release/obj.target/bcrypt_lib/src/bcrypt.o
5 CXX(target) Release/obj.target/bcrypt_lib/src/bcrypt_node.o
6 SOLINK_MODULE(target) Release/bcrypt_lib.node
7 clang: warning: libstdc++ is deprecated; move to libc++ with a minimum dep
8 Rocket.Chat@0.46.0-develop /Users/douglas/work/github/Rocket.Chat
```

```
9   └── bcrypt@0.8.7
10     ├── bindings@1.2.1
11     └── nan@2.3.5
```

Bots Guides



Rocket.Chat Bots Banner

What are bots in Rocket.Chat?

Bots (or Chatbots) are pieces of software or software systems that provide automated messaging to a chat platform. They typically add value by integrating large external services such as CRM and big data sources. Bots provide a friendly conversational interface that users on chat platforms are already familiar with, which also brings simplicity to complex systems. Besides, bots add automation, natural language understanding, and machine learning possibilities.

In Rocket.Chat, a bot is a special user account with the `bot` role and a specific set of permissions.

How do bots send and receive messages?

Bots in Rocket.Chat cannot send messages to users on their own. Instead, they subscribe to so-called collections of messages that every user has. The subscription creates a stream that is updated every time messages are sent either directly to bots or any room they are joined in.

For more details, see the [Bots Architecture section](#).

How are bots hosted?

Like most chatbot platforms, Rocket.Chat supports bots that are hosted outside of your Rocket.Chat instance, running on an external server.

For more details, see the [Bots Architecture section](#).

Are there different types of Bots?

All bots share the common properties described above. However, historical evolution driven by user and business demand has segregated bots into two major distinct categories:

- **chat-ops bots** Typically deployed in team-chat or group-chat environments to facilitate the operation of external systems (such as a development shop's CI pipelines, gitlab/github builds, etc). These bots will report on status of workflows and empower chat users to trigger or control operational workflows on these external systems. Chat-ops bots link *chat* to *operations*.
- **omnichannel bots** Typically backed by big data, CRMs, as well as machine learning engines to facilitate or supplement customer facing workflows and functions. These bots are deployed across channels (works in live chat, as readily as Whatsapp, as readily as Apple Business Chat; and works in harmony with human service agents) to carry out their tasks.

Getting Started

These are the basic steps for using bots with Rocket.Chat:

1. A bot user is [created by an admin](#) on the server;
 2. The bot is running as [a separate process](#) using your chosen framework or platform;
 3. The bot environment is pre-configured with [environment variables](#);
 4. The bot's behavior is defined via scripts according to the requirements of your framework. For example, check the details on how to [run a bBot bot](#).
-

Next steps

Get yourself familiar with [Bots Architecture](#).

Quick Links

- [Configure the Bot Environment](#)
 - [Create a Bot](#)
 - [Running a Botkit Bot](#)
 - [Running a Hubot Bot](#)
 - [Running a Botpress Bot](#)
 - [Running a Rasa Bot](#)
 - [Running a bBot Bot](#)
-

References

Configuration and implementation details for components of the [Bots Architecture](#) can be found in their project READMEs:

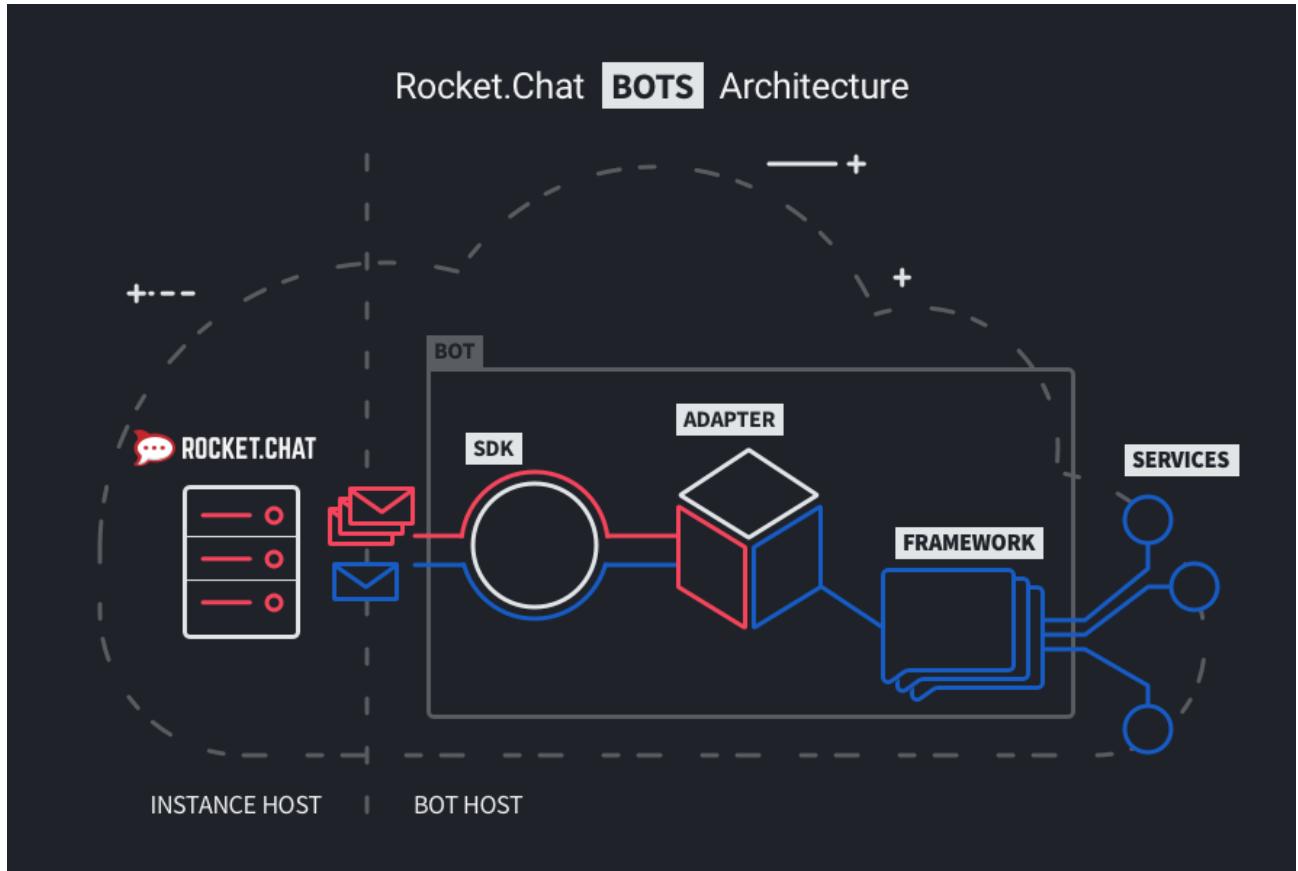
- [Rocket.Chat JS SDK](#)

- [bBot Rocket.Chat boilerplate](#)
 - [Hubot Rocket.Chat boilerplate](#)
 - [Hubot Rocket.Chat adapter](#)
-

Contribute

To contribute to features under development see our [Bots Project issues](#).

Bots Architecture



Bots Architecture Diagram

The above diagram illustrates the general concepts of how a Rocket.Chat server communicates with a chatbot framework or platform. These concepts are applicable to **external** bots.

Bots architecture consists of two main parts: Rocket.Chat host and **bot host**. These two parts are connected with each other via **Rocket.Chat SDK** methods. Bot host runs externally to the Rocket.Chat host and consists of the bot infrastructure and, typically, external **services**. The bot infrastructure, in turn, consists of **SDK**, **adapter**, and **framework**.

A bot, being subscribed to room's messages (all or specific ones) handles these messages and responds back to the user in an interactive manner. The conversational model can be designed and managed by tooling and frameworks on the bot host.

Today, many bots are created using bot frameworks or platforms based on popular programming languages, such as NodeJS and Python.

How Bots Architecture works

Bot creators design the bot's logic with the help of their favorite framework. This logic defines behavioral interaction with users, giving the bot ability to recognize the context of the conversation. In this way, the bot can focus on what users want to achieve and even take a proactive approach in finding the right solution and creating better user experience.

With the ability to listen to and respond to messages, the bot can be integrated into the Rocket.Chat host so that it can engage with users in a conversational manner. To perform this connection, bot creators use adapters that can translate the logic defined via the framework's powers to the format that the Rocket.Chat host understands. Adapters use Rocket.Chat SDK methods to communicate with the Rocket.Chat host.

Users and Bots

Bots in Rocket.Chat require user accounts with the `bot` role. Other than having a specific set of permissions, bot accounts are treated like regular user accounts within the Rocket.Chat host. However, to prevent possible user confusion, messages from online bots are marked with a default "Bot" label.

Bot Admin

Admins (and only admins) can create the bot account and configure role permissions. This includes setting the credentials a bot uses to connect as well as which user fields the bots have access to for privacy and security. Rocket.Chat team works on improving management views for bots and enhancing the future potential of user-activated bots.

Bots Host

A bot host is a machine where bots run. It can be a bare-metal server, a network cluster, a virtualized environment or cloud-based.

A bot host can manage and run many different bots as well as many instances of any bot.

The bot host typically manages connections to [external services](#) as well.

Bot platforms and frameworks

Rocket.Chat supports multiple bot frameworks and platforms, including: [bBot](#), [Hubot](#), [Botkit](#), [Botpress](#).

The hosting for bots depends on the platform. For example, Hubot can be launched on any Node.js environment, but you have to set it up yourself (using Heroku or Glitch services). Other bot platforms like Botkit provide their own services for hosting and provisioning bots. They just need a configuration to connect to your Rocket.Chat instance.

Message Streams

Every user in Rocket.Chat has a collection for their messages that can be emitted by the server's streamer. Bots subscribe to the collection of messages for their own user account, creating a stream that is updated every time a message is sent either directly to them or any room they are joined in.

Rocket.Chat SDK

The SDK is a low-level software module that offers an interface for external consumers to subscribe to message streams, send messages, and call methods on the Rocket.Chat server via WebSockets / DDP.

Bot framework adapter are written using the SDK.

Framework Adapter

An adapter (also referred as **connector** or **middleware**) is a software component that is used to link the framework with the Rocket.Chat host.

An adapter allows bot creators to design and build bot interactions according to the framework's rules and practices without dealing with the low-level programming details and methods of the communication with the Rocket.Chat host.

Adapters use Rocket.Chat SDK to communicate with the Rocket.Chat host.

Bot Framework

Bot frameworks (such as Hubot, Botkit, Rasa or Botpress) provide unique approaches to create conversational interfaces and allow developers to focus on the design and management of complex bot logic.

Most of frameworks can be run on your own bot host. There are also frameworks that operate as SaaS (Software as a Service), so they work in the cloud.

Most of the popular frameworks have adapters created by the Rocket.Chat's community.

Bot Scaling

In production, hundreds or thousands of users can be carrying out conversations with a bot at the same time. The ability to scale a bot from one conversation to many concurrent ones is essential.

Bot frameworks/platforms support varying degrees of scalability. To ensure that growing number of users will not cause architectural and performance problems, you should choose your bot's framework wisely. For example, considerng potential scalability issues even before you start implementing your bot's logic

External Services

The value of most production bots relies on their ability to access and manipulate external data and systems (services). External services, like databases, scheduling systems, CRM, NLP can be used to provide data or conversational processing.

The framework simplifies connection of a bot with these services and ensures ease of access to them through the bot host.

Configure bot environments

Regardless of bot type (e.g. Hubot, Botkit, Rasa, etc.), the Rocket.Chat SDK requires you to create a configuration with some variables for your environment. This configuration can be stored in a `.env` file, for local development, for example. In production, they would need to be set on server startup.

The following is a list of all the environmental variables you can use in your project. Variables marked with an asterisk (`*`) are mandatory.

Environment Variable	Description
<code>ROCKETCHAT_URL</code> *	URL of the Rocket.Chat instance to connect to. Can be specified as <code>host:port</code> , <code>http://host:port</code> or <code>https://host:port</code> .
<code>ROCKETCHAT_USE_SSL</code>	Force bot to connect with SSL. If unset, it will try and detect from URL protocol.
<code>ROCKETCHAT_AUTH</code>	Authorization method for a bot. Default: <code>password</code> . Set to <code>ldap</code> to enable LDAP login for bot users.
<code>ROCKETCHAT_USER</code> *	The bot's username (account name users will summon the bot with). Must be registered on your Rocket.Chat server and granted <code>bot</code> role.
<code>ROCKETCHAT_PASSWORD</code> *	The bot user's password.
<code>ROCKETCHAT_ROOM</code>	Stream callbacks receive messages from the listed channel name/s. Default: <code>GENERAL</code> . Accepts a comma separated list. Allows the bot to listen and respond to messages <i>from all newly created private groups</i> where the bot's user has been added as a member. Should be empty if <code>LISTEN_ON_ALL_PUBLIC=true</code>

Responding to channels and DMS

RESPOND_TO_LIVECHAT	Stream callbacks receive messages from Livechat (true/false). Default: <code>false</code> .
RESPOND_TO_DM	Stream callbacks receive DMs with bot (true/false). Default: <code>false</code> .
RESPOND_TO_EDITED	Stream callbacks receive edited messages (true/false). Default: <code>false</code> .
SDK Development	
ROOM_CACHE_SIZE	Size of cache (LRU) for room (ID or name) lookups.
ROOM_CACHE_MAX_AGE	Max age of cache for room lookups.
DM_ROOM_CACHE_SIZE	Size of cache for Direct Message room lookups.
DM_ROOM_CACHE_MAX_AGE	Max age of cache for DM lookups.
INTEGRATION_ID	ID applied to message object to integration source. Default: <code>js.SDK</code>
ADMIN_USERNAME	Admin user name for API calls used in SDK tests.
ADMIN_PASS	Admin user password for API calls used in SDK tests.
Hubot Specific	
HUBOT_NAME	Name of the bot. Hubot listeners can respond to this.
HUBOT_ALIAS	Another name to respond to. If unset, the adapter sets the <code>ROCKETCHAT_USER</code> as an alias, to ensure bots respond when addressed using their username.
EXTERNAL_SCRIPTS	Hubot scripts to require as NPM modules. Used only in some Docker instances.
HUBOT_LOG_LEVEL	<code>debug</code> , <code>info</code> , <code>warning</code> or <code>error</code> . Default: <code>info</code> .

Common set of variables

The common set for the majority of bots is to listen and respond to direct messages and all new public channels and private groups:

- RESPOND_TO_DM=true
- ROCKETCHAT_ROOM=' '

Be aware you *must* add the bot's user as a member of the new private group(s) before it will respond.

Create and run a Bot

This document outlines the steps required to create your first bot with Rocket.Chat.

Prerequisites

- [Running Rocket.Chat server](#)
 - User with admin privileges
-

Steps

1. Create a bot user

In order to talk to your chatbot there must be a user account pre-configured on the Rocket.Chat server that the bot can log in to.

To create the account you need to have admin privileges:

1. In the topbar, click the three dots (Options) and then click **Administration**
2. Select **Users** from the left sidebar
3. Click  (Add User) button in the right sidebar
4. In the profile window that appears, fill in *Name*, *Username*, *Email* and *Password* fields
5. Enable *Verified* toggle under the *Email* field
6. Disable *Require password change* toggle under the *Password* field
7. Select `bot` from the `Add Role` dropdown menu and click `Add Role` button to the right
8. Disable *Join default channels* and *Send welcome email* checkboxes
9. Click `Save`

Once saved, the bot will be configured with the username and password set on step 4. You can use `ROCKETCHAT_USER` and `ROCKETCHAT_PASSWORD` [environmental variables](#) to log in to Rocket.Chat server with this username and password pair.

NOTE: to avoid creating multiple accounts for bot emails, you can use Gmail +address alias. For example: `youremail+botnam@gmail.com`. See this issue for more.

2. Code your bot

To make the process of coding a bot easier and faster, you may want to check our existing guides below to quickly deploy a basic bot instance. As an advanced alternative, you can create your bot from the scratch using your favorite framework.

- [Running a Botkit Bot](#)
- [Running a Hubot Bot](#)
- [Running a Botpress Bot](#)
- [Running a Rasa Bot](#)
- [~~Running a bBot Bot~~](#)

Regardless of the option you choose, to make your bot work you will need credentials of the bot user you created in the previous step.

3. Talk to your bot

If the bot is configured to listen to direct messages (`RESPOND_TO_DM=true`), and messages are prepended with `BOT_NAME` or a preconfigured `BOT_ALIAS`, the bot will *usually* respond to all messages addressed directly to the bot user (depending on the particular bot framework).

Run a Hubot bot

Quick start guide

The fastest way to start with Hubot bot is to use [Hubot boilerplate](#). The boilerplate is a simple Node.js package that requires Hubot and the Rocket.Chat adapter. The bot can then be executed using a [bin file](#) in production, or via the package scripts locally using

```
npm run local or yarn local .
```

Navigate to the folder where you want to work with the bot and proceed with the following steps:

1. clone the boilerplate repo and install dependencies

```
1 git clone https://github.com/RocketChat/hubot-rocketchat-boilerplate  
2 cd hubot-rocketchat-boilerplate  
3 npm install
```

2. create a `.env` file with the following content

```
1 export ROCKETCHAT_URL=myserver.com  
2 export ROCKETCHAT_USER=mybotuser  
3 export ROCKETCHAT_PASSWORD=mypassword  
4 export ROCKETCHAT_ROOM=general  
5 export ROCKETCHAT_USESSL=true
```

Adjust the content to fit your server and user credentials. Make sure `mybotuser` has a `BOT` role on the server. For more information on how to create a bot user please refer to [this page](#).

3. run the bot

```
1 source .env
2 bin/hubot
```

After executing the last command, hubot will try to connect to the Rocket.Chat instance and then will listen to messages in `general` room:

```
1 bin/hubot
2 audited 390 packages in 1.929s
3 found 1 low severity vulnerability
4 run `npm audit fix` to fix them, or `npm audit` for details
5 [Fri Sep 27 2019 23:24:19 GMT+0300 (Eastern European Summer Time)] INFO [s
6 [Fri Sep 27 2019 23:24:19 GMT+0300 (Eastern European Summer Time)] INFO [s
7 [Fri Sep 27 2019 23:24:19 GMT+0300 (Eastern European Summer Time)] INFO [s
8 [Fri Sep 27 2019 23:24:19 GMT+0300 (Eastern European Summer Time)] INFO [c
9 password: 'mypassword',
10 ldap: false,
11 host: 'myserver.com',
12 useSsl: true,
13 timeout: 20000,
14 rooms: [ 'general' ],
15 allPublic: false,
16 dm: true,
17 livechat: false,
18 edited: false,
19 integrationId: 'js.SDK',
20 roomCacheMaxSize: 10,
21 roomCacheMaxAge: 300000,
22 dmCacheMaxSize: 10,
23 dmCacheMaxAge: 100000 }
24 [Fri Sep 27 2019 23:24:19 GMT+0300 (Eastern European Summer Time)] INFO [c
25 [Fri Sep 27 2019 23:24:19 GMT+0300 (Eastern European Summer Time)] INFO [l
26 [Fri Sep 27 2019 23:24:20 GMT+0300 (Eastern European Summer Time)] INFO [s
27 [Fri Sep 27 2019 23:24:20 GMT+0300 (Eastern European Summer Time)] INFO [s
28 [Fri Sep 27 2019 23:24:20 GMT+0300 (Eastern European Summer Time)] INFO [r
29 [Fri Sep 27 2019 23:24:20 GMT+0300 (Eastern European Summer Time)] INFO [j
```

4. talk to your bot

On the server, login as a regular user (not the BOT user), go to `general` room, and try the following commands:

```
mybotuser what time is it
```

D docsUser 11:32 PM
docsBot what time is it

D docsBot Bot 11:32 PM
docsUser It's 23:32 and 6 seconds

Bot's response to what time is it

This message will also appear in the terminal window:

```
1 [Fri Sep 27 2019 23:32:06 GMT+0300 (Eastern European Summer Time)] INFO [r
2 [Fri Sep 27 2019 23:32:06 GMT+0300 (Eastern European Summer Time)] INFO [r
3 [Fri Sep 27 2019 23:32:06 GMT+0300 (Eastern European Summer Time)] INFO Fi
4 [Fri Sep 27 2019 23:32:06 GMT+0300 (Eastern European Summer Time)] INFO [s
5 [Fri Sep 27 2019 23:32:06 GMT+0300 (Eastern European Summer Time)] INFO [r
```

Another command:

```
mybotuser rc version
```

D docsUser 11:40 PM
docsBot rc version

D docsBot Bot 11:40 PM
You're on Rocket.Chat 1.2.0-rc.0, using Hubot 3.3.2.

Adapter version 2.0.0-development, using version 0.2.9 of the SDK.

Bot's response to rc-version

This message will also appear in the terminal window:

```
1 [Fri Sep 27 2019 23:40:20 GMT+0300 (Eastern European Summer Time)] INFO [r
```

```
2 i Sep 27 2019 23:40:20 GMT+0300 (Eastern European Summer Time)] INFO [rece
3 [Fri Sep 27 2019 23:40:20 GMT+0300 (Eastern European Summer Time)] INFO Fi
4 [Fri Sep 27 2019 23:40:21 GMT+0300 (Eastern European Summer Time)] INFO [s
5 [Fri Sep 27 2019 23:40:21 GMT+0300 (Eastern European Summer Time)] INFO [s
6 [Fri Sep 27 2019 23:40:21 GMT+0300 (Eastern European Summer Time)] INFO [r
```

Run a Hubot bot on production

With the example above, you can run the bot in production using the following command:

```
bin/hubot -a rocketchat
```

This command will install dependencies and run the bot with Rocket.Chat's Hubot adapter.

Advanced instructions

Configuration

When running locally, `dotenv` is used to load configurations from the `./.env` file. That makes it easier to set environment variables.

Please see a full list of environmental variables [here](#).

Please pay attention that if `ROCKETCHAT_URL` is using `https://`, you **MUST** setup websocket pass-through on your reverse proxy (for example, NGINX) with a valid certificate (not self-signed). Directly accessing Rocket.Chat without a reverse proxy via `https://` is not possible.

Running in production

There are executables for different environments that all run the Hubot binary:

- `bin/hubot` - for Linux/Unix
- `bin/hubot.cmd` - for Windows
- `Procfile` - for Heroku

Before running, make sure your production environment has the required environment variables for the adapter, url, user, name and pass. As an alternative, you can add them as command-line parameters, like `-a rocketchat`.

Environmental variables should be populated on the server before the launch (see [configuration](#)). The launcher will also install npm dependencies on every run, in case it is booting in a fresh container (this isn't required when working locally).

See more information on deployment configurations [here](#).

Adding scripts

Scripts can be added to the `./scripts` folder, or by installing node packages and listing their names in the `external-scripts.json` array. There is an example of each in this repo but neither is required.

Checking example scripts

The boilerplate comes pre-packed with two scripts as demos for manual tests. In each of the following script, you can talk to a bot in a public channel by prefixing the bot's username or in direct messages without the bot's username.

- `what time is it` or `what's the time` - Tells you the current time
- `rc version` - Gives you version info about Rocket.Chat and Hubot (two messages)

Deployment using Docker

1. clone the hubot repo and go into the clonned directory

```
1 git clone git@github.com:RocketChat/hubot-rocketchat.git
2 cd hubot-rocketchat
```

2. start the Docker container

```
1 docker run -it -e ROCKETCHAT_URL=your rocketchat instance>:<port> \
2   -e ROCKETCHAT_ROOM='' \
3   -e LISTEN_ON_ALL_PUBLIC=true \
4   -e RESPOND_TO_DM=true \
5   -e ROCKETCHAT_USER=mybotuser \
6   -e ROCKETCHAT_PASSWORD=mybotpassword \
7   -e HUBOT_NAME=bot \
8   -e EXTERNAL_SCRIPTS=hubot-help,hubot-diagnostics \
9   rocketchat/hubot-rocketchat
```

On the container launch, hubot will try to connect to the Rocket.Chat instance and then will listen to messages in all public rooms:

```
1 [Sun Sep 29 2019 16:49:54 GMT+0000 (UTC)] INFO Starting Rocketchat adapter
2 [Sun Sep 29 2019 16:49:54 GMT+0000 (UTC)] INFO Once connected to rooms I w
3 [Sun Sep 29 2019 16:49:54 GMT+0000 (UTC)] INFO I will also respond to my R
4 [Sun Sep 29 2019 16:49:54 GMT+0000 (UTC)] WARNING No services ROCKETCHAT_R
5 [Sun Sep 29 2019 16:49:54 GMT+0000 (UTC)] INFO Connecting To: <rocketchat_
6 [Sun Sep 29 2019 16:49:55 GMT+0000 (UTC)] INFO Successfully connected!
7 [Sun Sep 29 2019 16:49:55 GMT+0000 (UTC)] INFO
8 [Sun Sep 29 2019 16:49:55 GMT+0000 (UTC)] INFO Logging In
9 [Sun Sep 29 2019 16:49:55 GMT+0000 (UTC)] INFO Successfully Logged In
10 [Sun Sep 29 2019 16:49:55 GMT+0000 (UTC)] INFO rid: []
11 [Sun Sep 29 2019 16:49:55 GMT+0000 (UTC)] INFO All rooms joined.
12 [Sun Sep 29 2019 16:49:55 GMT+0000 (UTC)] INFO Preparing Meteor Subscripti
13 [Sun Sep 29 2019 16:49:55 GMT+0000 (UTC)] INFO Subscribing to Room: __my_m
14 [Sun Sep 29 2019 16:49:55 GMT+0000 (UTC)] INFO Successfully subscribed to
15 [Sun Sep 29 2019 16:49:55 GMT+0000 (UTC)] INFO Setting up reactive message
```

3. talk to your bot

On the server, login as a regular user (not the BOT user), go to `general` or any other room, and try the following command:

```
mybotuser time
```

You can also talk to the bot using its alias:

```
rocketbot time
```

Both commands result in the same output:

```
D docsUser 8:11 PM
D docsBot time

D docsBot 8:11 PM
Server time is: Sun Sep 29 2019 17:11:20 GMT+0000 (UTC)

D docsUser 8:11 PM
rocketbot time

D docsBot 8:11 PM
Server time is: Sun Sep 29 2019 17:11:26 GMT+0000 (UTC)
```

Important notes

- The first time you run the docker container, the image needs to be pulled from the public docker registry, which takes some time. Subsequent runs are super fast.
- If you are not running Linux (i.e. if you are on a Mac or PC), you cannot use \$PWD to mount the volumes. Instead, [read the *If you are using Boot2Docker...* note](#) to determine the absolute path where you must place the git-cloned directory.

Adding scripts

To include your own scripts, you can use the following Docker command:

```
1 docker run -it -e ROCKETCHAT_URL=your rocketchat instance>:<port> \
2   -e ROCKETCHAT_ROOM='' \
3   -e LISTEN_ON_ALL_PUBLIC=true \
4   -e RESPOND_TO_DM=true \
5   -e ROCKETCHAT_USER=mybotuser \
6   -e ROCKETCHAT_PASSWORD=mybotpassword \
7   -e HUBOT_NAME=bot \
8   -e EXTERNAL_SCRIPTS=hubot-help,hubot-diagnostics \
```

```
9      -v $PWD/scripts:/home/hubot/scripts \
10     rocketchat/hubot-rocketchat
```

Deployment using Yeoman

[Yeoman generator](#) allows you to generate a Hubot bot using the Rocket.Chat adapter. Yeoman uses the latest versions of Hubot and Rocket.Chat adapter module, which enable asynchronous processing in ES6 Javascript, and introduce various bug fixes.

Development for different versions of adapter

v2.x.x

Standard

In a Hubot instance once `hubot-rocketchat` is added by npm or yarn, you can replace the package with a development version directly:

- `cd node_modules` from the bot's project root
- `rm -rf hubot-rocketchat` to delete the published version
- `git clone git@github.com:RocketChat/hubot-rocketchat.git` to add dev version
- `cd hubot-rocketchat` move to dev path
- `npm install` install dependencies

Linked

Setting up a locally linked package is easier for continued development and/or using the same development version of the adapter in multiple bots.

- Change directory to your development adapter path
- `npm link` or `yarn link` to set the origin of the link
- Change directory to your bot's project root
- `npm link hubot-rocketchat` or `yarn link hubot-rocketchat` to create the link

v1.x.x and v0.x.x

Docker-compose

If you want to use docker-compose for this task, add this for v0.1.4 adapter (this must be inserted in your docker-compose.yml):

```
1 # hubot, the popular chatbot (add the bot user first and change the password)
2 hubot:
3   image: rocketchat/hubot-rocketchat:v0.1.4
4   environment:
5     - ROCKETCHAT_URL=your-rocket-chat-instance-ip:3000 (e.g. 192.168.2.240)
6     - ROCKETCHAT_ROOM=general
7     - RESPOND_TO_DM=true
8     - ROCKETCHAT_USER=username-of-your-bot
9     - ROCKETCHAT_PASSWORD=yourpass
10    - BOT_NAME=bot
11    - GOOGLE_API_KEY=yourgoogleapikey
12  # you can add more scripts as you'd like here, they need to be installable
13    - EXTERNAL_SCRIPTS=hubot-help,hubot-seen,hubot-links,hubot-diagnostics
14  links:
15    - rocketchat:rocketchat
16  # this is used to expose the hubot port for notifications on the host on port 8080
17  ports:
18    - 3001:8080
```

If you want your bot to listen to all public rooms and all private rooms it is joined to, leave the env `ROCKETCHAT_ROOM` empty like in the example above and set the env `LISTEN_ON_ALL_PUBLIC=true`.

Please pay attention to some external scripts that are in the example above, as some of them need your Google-API-Key in the docker compose file.

Version information and compatibility

The examples above use [Hubot](#) v3 and [Rocketchat.Chat adapter](#) v2, using the new [Rocket.Chat JS SDK][[sdk](#)] for Rocket.Chat instances 0.60.0 onward.

v2 contains major breaking changes. Starting with this version:

- CoffeeScript BOTs will no longer be supported, all new bot scripts should be written in NodeJS 8 compatible JavaScript
- Only [Hubot v3](#) will be supported
- [Rocket.Chat Server version 0.63.0](#) or higher is supported
- Yeoman generator support is discontinued, you should start your project with our [hubot v3 boilerplate](#)
- [NodeJS 8](#) or later is required for operations
- The latest ES6 syntax in NodeJS 8, including the popular async-await, is fully supported

Hubot versions

Versions of `hubot-rocketchat` prior to v2 are incompatible with Hubot v3

This bot is written in ES6 and intended to run on Node v8+. To run a bot on older versions of Node would require compilation with Babel to use the full ES6 feature set.

Adapter versions

Older versions of the adapter (before v0.1.4) are incompatible with more recent versions of Rocket.Chat (v0.35+). Please report an issue, if you find specific version mismatch.

NOTE: releases between 0.35.0 and 0.37.1 are not recommended for hubot operations.

Rocket.Chat adapter v1 uses CoffeeScript, which makes extention of classes in JavaScript ES6 troublesome.

v1 versions of the adapter are only compatible with 0.37.1 and higher of Rocket.Chat Server. However, you should be able to continue using v1 adapter and CoffeeScript bots with the most up-to-date version of the server.

If you are writing CoffeeScript bots, you need Hubot v2.x and v1.x.x or v0.x.x of the adapter. Please see [instructions for v1.x.x and v0.x.x of the adapter](#).

Additional information

For more examples on building and scripting your bot, please see [Hubot documentation](#).

FAQ

Q: I am not trying to stage a denial of service attack, why would I ever want to write a bot?

A: There are many positive and productive use cases for bots. Imagine a customer service support chat. As soon as a customer enters the support channel, a bot immediately identifies the customer and then:

- fetches recent sales information from the sales dept server
- fetches personal information from the customer data base
- fetches latest notes made by her/his salesperson from the CRM system
- scans the customer's facebook and twitter posts
- obtains details of the last support ticket for this customer

Putting it altogether and then private message the service rep with the information.

Another use-case is a load test bot, imagine a bot that accepts the command:

```
rocketbot loadtest europe 25, asia 50, usa 100, canada 10
```

This command specifies a distribution of test bot instances, to be created across globally located data centers.

Once received, the bot:

- parses the distribution
- concurrently ssh to remote Kubernetes controllers and spawns the specified number of test bot instances to start the load test

Q: The architecture of hubot-rocketchat looks interesting, can you tell me more about it? A: Sure, it is based on hubot-meteorchat. hubot-meteorchat is the hubot integration project for Meteor based chats and real-time messaging systems. Its driver based architecture simplifies creation and customization of adapter for new systems. For example, the hubot-rocketchat integration is just hubot-meteorchat + Rocket.Chat driver.

Learn more about hubot-meteorchat and other available drivers [at this link](#).

Contributions and discussions

Please see [our documentation on contributing](#), then [visit the issues](#) to share your needs or ideas.

If you want to contribute to the development or improvement of this adapter, pull requests are welcome.

Feel free to join us in the [#hubot](#) channel to discuss hubot, and any scripts you might be working on.

Run a Botkit Bot

Botkit is the leading developer tool for building chat bots, apps and custom integrations for major messaging platforms.

Botkit offers everything you need to design, build and operate an app:

- Easy-to-extend starter kits;
- Fully-featured SDK with support for all major platforms;
- **Tons of plugins and middlewares**

In addition, Botkit works with all the NLP services (like Microsoft LUIS and IBM Watson), it can use any type of database, and runs on almost any hosting platform.

Quick start guide

Botkit is a Node.js module and works with Node and npm.

The fastest way to start with Botkit is using [botkit-starter-rocketchat](#):

1. clone the starter repo and install dependencies

```
1 git clone https://github.com/RocketChat/botkit-starter-rocketchat
2 cd botkit-starter-rocketchat/
3 npm install
```

2. create a `.env` file with the following content

```
1 ROCKETCHAT_URL=<ROCKETCHAT HOST>
2 ROCKETCHAT_USER=<BOTKIT USER NAME>
3 ROCKETCHAT_PASSWORD=<BOTKIT USER PASS>
4 ROCKETCHAT_USE_SSL=<SSL USAGE>
5 ROCKETCHAT_ROOM=<ROCKETCHAT CHANNEL>
```

```
6 MENTION_ROOMS=<CHANNEL THAT THE BOT WILL ONLY ANSWER WHEN MENTIONED>
7 RESPOND_TO_DM=<ENABLE BOT TO RESPOND TO DIRECT MESSAGES>
8 RESPOND_TO_LIVECHAT=<ENABLE BOT TO RESPOND TO LIVE CHAT>
9 RESPOND_TO_EDITED=<ENABLE BOT TO RESPOND TO EDITED MESSAGES>
```

3. run the bot

```
npm start
```

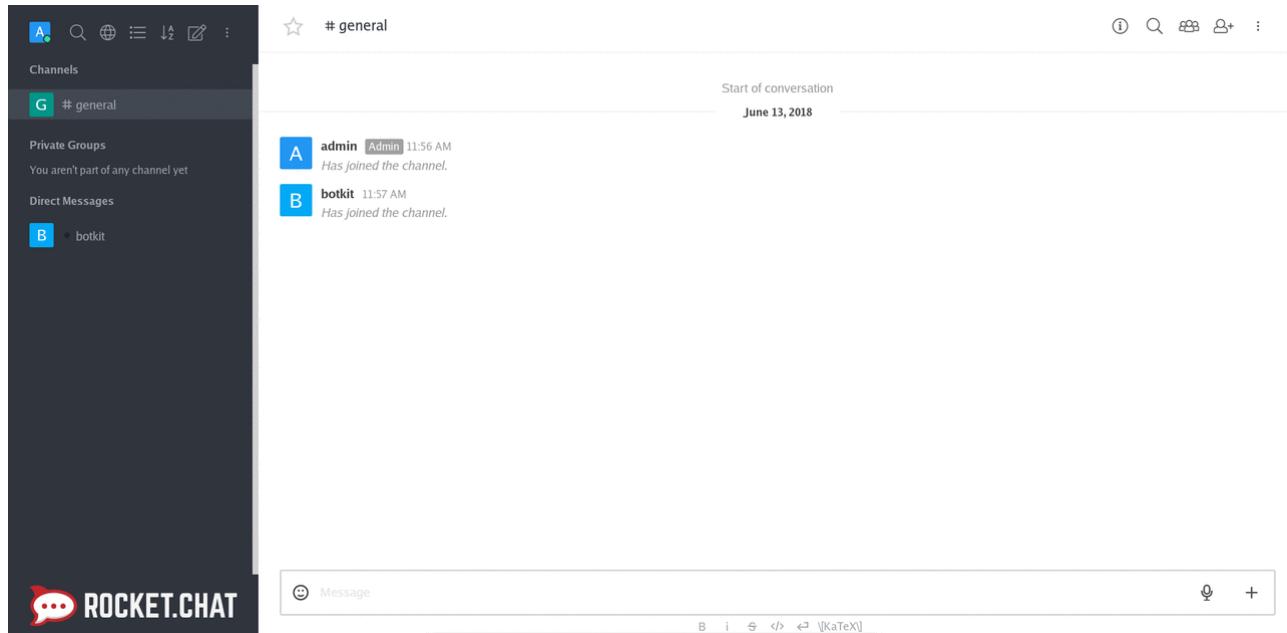
After executing the command, botkit will try to connect to the Rocket.Chat instance and then will listen to messages in `general` room:

```
1 Initializing Botkit v0.6.16
2 info: ** No persistent storage method specified! Data may be lost when pro
3 debug: Setting up a handler for spawned
4 debug: Setting up a handler for heard_trigger
5 debug: Setting up a handler for command_triggered
6 debug: Setting up a handler for remote_command_end
7 debug: Setting up a handler for direct_message
8 debug: Setting up a handler for live_chat
9 debug: Setting up a handler for channel
10 debug: Setting up a handler for private_channel
11 [connect] Connecting { username: 'mybotuser',
12 password: 'mypassword',
13 ldap: false,
14 host: 'myserver.com',
15 useSsl: true,
16 timeout: 20000,
17 rooms: [ 'GENERAL' ],
18 allPublic: false,
19 dm: true,
20 livechat: true,
21 edited: true,
22 integrationId: 'js.SDK',
23 roomCacheMaxSize: 10,
24 roomCacheMaxAge: 300000,
25 dmCacheMaxSize: 10,
26 dmCacheMaxAge: 100000 }
27 debug: Setting up a handler for message_received
28 debug: Setting up a handler for sticker_received
29 debug: Setting up a handler for image_received
30 debug: Setting up a handler for audio_received
```

```
31 debug: Setting up a handler for heard_trigger
32 debug: Setting up a handler for conversationStarted
33 debug: Setting up a handler for message_received
34 ~~~~~
35 NOTE: Botkit Studio functionality has not been enabled
36 To enable, pass in a studio_token parameter with a token from https://stud
37 Botkit Studio statistics are no longer supported. Update your project to t
38 [connect] Connected
39 [login] Logging in mybotuser
40 [getRoomIdByNameOrId] Calling (caching): GENERAL
41 [getRoomIdByNameOrId] Success: "GENERAL"
42 [joinRoom] Calling (async): ["GENERAL"]
43 [joinRoom] Success
44 [subscribe] Preparing subscription: stream-room-messages: __my_messages__
45 [subscribe] Stream ready: 4
46 [reactive] Listening for change events in collection stream-room-messages
```

4. talk to your bot

On the server, login as a regular user (not the BOT user), go to `general` room, and try to talk to your bot:



Botkit bot is talking

Run a Rocket.Chat SDK Bot

This bot is a simple example of how you can use [Rocket.Chat JS SDK](#) methods directly. It is extremely basic and does not handle errors, different message types, server resets and other production situations.

Quick start guide

Rocket.Chat Bot is a Node.js package and works with Node and npm.

Navigate to the folder where you want to work with the bot and proceed with the following steps:

1. install Rocket.Chat SDK package

```
1 npm init -y
2 npm install @rocket.chat/sdk
```

2. create bot files

To proceed with the simplest setup, you need to create two files: the first one will be responsible for the working logic of the bot. Another one will contain a map with the responses.

1. create a `server.js` file with the following content:

```
1 const { driver } = require('@rocket.chat/sdk');
2 const respmap = require('./reply');
3
4 // Environment Setup
5 const HOST = '<ROCKETCHAT HOST>';
6 const USER = '<BOT USER NAME>';
7 const PASS = '<BOT USER PASS>';
8 const BOTNAME = '<ROCKET CHAT BOT ALIAS>';
```

```

9  const SSL = '<SSL USAGE>';
10 const ROOMS = ['<ROCKETCHAT CHANNEL>'];
11 var myUserId;
12
13 // Bot configuration
14 const runbot = async () => {
15     const conn = await driver.connect({ host: HOST, useSsl: SSL })
16     myUserId = await driver.login({ username: USER, password: PASS });
17     const roomsJoined = await driver.joinRooms( ROOMS );
18     console.log('joined rooms');
19
20     const subscribed = await driver.subscribeToMessages();
21     console.log('subscribed');
22
23     const msgloop = await driver.reactToMessages( processMessages );
24     console.log('connected and waiting for messages');
25
26     const sent = await driver.sendToRoom( BOTNAME + ' is listening ...', R
27     console.log('Greeting message sent');
28 }
29
30 // Process messages
31 const processMessages = async(err, message, messageOptions) => {
32 if (!err) {
33     if (message.u._id === myUserId) return;
34     const roomname = await driver.getRoomName(message.rid);
35
36     console.log('got message ' + message.msg)
37     var response;
38     if (message.msg in respmap) {
39         response = respmap[message.msg];
40     }
41     const sentmsg = await driver.sendToRoomId(response, message.rid)
42     }
43 }
44
45 runbot()

```

NOTE: Adjust the `Environment Setup` section content to fit your server and user credentials.

Make sure `<BOT USER NAME>` has a `BOT` role on the server. For more information on how to create a bot user please refer to [this page](#).

1. create a `reply.js` file with the following content:

```
1 const respmap = {  
2     "hi" : "hey",  
3     "u da bot" : "no, YOU da bot",  
4     "no u da bot" : "Come'on - YOU DA BOT!!",  
5     "I give up" : "ok. silly human :rolleyes:"  
6 };  
7  
8 module.exports = respmap;
```

3. run the bot

```
node server.js
```

After executing the last command, Rocket.Chat SDK bot will try to connect to the Rocket.Chat instance and then will listen to messages in `general` room:

```
1 $ node server.js  
2 [connect] Connecting { username: 'username',  
3   password: 'pass',  
4   ldap: false,  
5   host: 'rocket.chat.host',  
6   useSsl: true,  
7   timeout: 20000,  
8   rooms: [],  
9   allPublic: false,  
10  dm: false,  
11  livechat: false,  
12  edited: false,  
13  integrationId: 'js.SDK',  
14  roomCacheMaxSize: 10,  
15  roomCacheMaxAge: 300000,  
16  dmCacheMaxSize: 10,  
17  dmCacheMaxAge: 100000 }  
18 [connect] Connected  
19 [login] Logging in botUser  
20 [getRoomIdByNameOrId] Calling (caching): general  
21 [getRoomIdByNameOrId] Success: "GENERAL"  
22 [joinRoom] Calling (async): ["GENERAL"]  
23 [joinRoom] Success  
24 joined rooms
```

```
25 [subscribe] Preparing subscription: stream-room-messages: __my_messages__
26 [subscribe] Stream ready: 4
27 subscribed
28 [reactive] Listening for change events in collection stream-room-messages
29 connected and waiting for messages
30 [getRoomIdByNameOrId] Calling (cached): general
31 [getRoomIdByNameOrId] Success: "GENERAL"
32 [sendMessage] Calling (async): [{"msg": "Rocket.Chat BOT is listening ..."}]
33 [sendMessage] Success: {"msg": "Rocket.Chat BOT is listening ...", "bot": {"id": "i..."}}
34 Greeting message sent
35 [received] Message in room GENERAL
```

4. talk to your bot

On the server, login as a regular user (not the BOT user), go to `general` room, and talk to your newly created bot:

Channels

November 13, 2019

general

Private Groups

testchannel

Direct Messages

docsBot

D docsBot Bot 12:47 AM Rocket Chat BOT is listening ...

D docsUser Admin 12:49 AM hi

D docsBot Bot 12:49 AM hey

D docsUser Admin 12:49 AM u da bot

D docsBot Bot 12:49 AM no, YOU da bot

D docsUser Admin 12:49 AM no u da bot

D docsBot Bot 12:49 AM Come'on - YOU DA BOT!!

D docsUser Admin 12:49 AM I give up

D docsBot Bot 12:49 AM ok. silly human :rolleyes:

Rocket.Chat SDK bot responses to user messages

Run a Botpress Bot

Botpress is an on-prem, open-source chatbot building platform for businesses.

Quick start guide

Botpress is a Node.js package and works with Node and npm.

The fastest way to start with Botpress is using [botpress-kick-starter](#) repository that is already integrated with Rocket.Chat via [botpress-channel connector](#).

Navigate to the folder where you want to work with the bot and proceed with the following steps:

1. install botpress package

```
1 npm init -y  
2 npm install botpress@10.40.0
```

1. create a bot

```
1 mkdir bot  
2 cd bot  
3 ../node_modules/.bin/botpress init
```

`botpress init` command launches a wizard that will help you create the initial configuration. You should see something as follows:

```
1 $ ../node_modules/.bin/botpress init  
2  
3 -----
```

```
4 Hey there !, thanks for using Botpress!
5 We'll walk you through the creation of your new bot.
6 For more information or help, please visit https://botpress.io/docs
7 -----
8 name: (bot) botpress-bot
9 botpress version: (10.40.0)
10 description: Test bot for docs
11 author: Anton Zolotukhin
12
13 =====
14 Template: Basic (default)
15 Author: Botpress, Inc.
16 Description: A basic bot for your website
17 Channels: web
18 =====
19
20 -> Copying .welcome
21 -> Copying botfile.js
22 -> Copying config/channel-web.json
23 -> Copying config/nlu.json
24 -> Copying generated/content/builtin_text.json
25 -> Copying generated/flows/main.flow.json
26 -> Copying generated/flows/main.ui.json
27 -> Copying generated/intents/forget.json
28 -> Copying generated/intents/forget.utterances.txt
29 -> Copying generated/intents/hello.json
30 -> Copying generated/intents/hello.utterances.txt
31 -> Copying generated/intents/none.json
32 -> Copying generated/intents/none.utterances.txt
33 -> Copying index.js
34 -> Copying package.json
35 -> Copying README.md
36 -> Copying src/actions.js
37 -> Copying src/content/.empty
38 -> Copying src/custom.js
39 -> Copying src/index.js
40 -> Copying src/renderer.js
41 -> Copying .gitignore
42
43 ! Your bot was initialized successfully!
44
45 Next steps:
46 1) Install bot dependencies by running npm install (or yarn install)
47 2) Start the bot by running npm start (or yarn start)
48
49 Enjoy Botpress!
```

1. install botpress connector and bot dependencies

```
1 npm install
2 npm install botpress-channel-rocketchat
```

2. configure the bot

Open `config` folder and create `channel-rocketchat.json` file with the following content:

```
1 {
2     "ROCKETCHAT_USER": <BOTPRESS USER NAME>,
3     "ROCKETCHAT_PASSWORD": <BOTPRESS USER PASS>,
4     "ROCKETCHAT_URL": <ROCKETCHAT HOST>,
5     "ROCKETCHAT_USE_SSL": <SSL USAGE>,
6     "ROCKETCHAT_ROOM": <ROCKETCHAT CHANNEL>,
7     "scope": ""
8 }
```

Adjust the content to fit your server and user credentials. Make sure `mybotuser` has a `BOT` role on the server. For more information on how to create a bot user please refer to [this page](#).

3. run the bot

```
npm start
```

After executing the command, botpress will try to do the following actions:

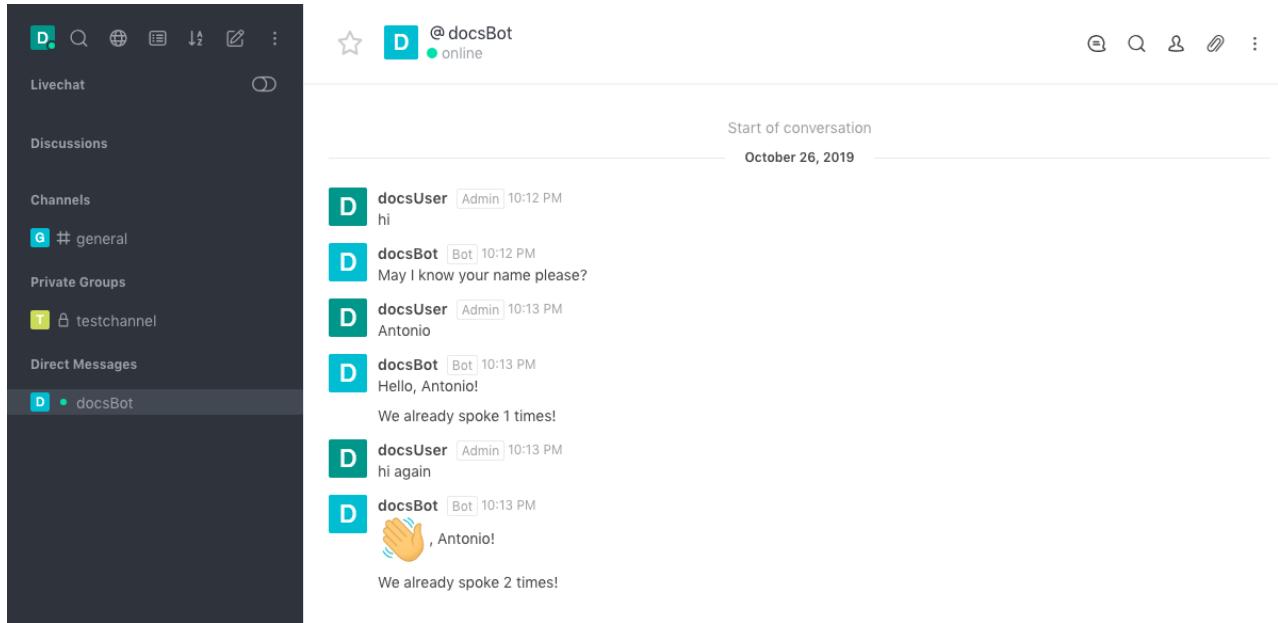
- launch botpress instance;
- launch dedicated Admin Dashboard;
- launch dedicated Webchat window;
- connect to the Rocket.Chat instance;
- listen to messages in `general` room.

```
1 $ npm start
2
3 > botpress-bot@1.0.0 start /botpress-bot/bot
4 > node index.js
5
6 19:27:44 - debug: [DB Janitor] Added table "logs"
7 19:27:44 - info: Starting botpress version 10.40.0
8 19:27:44 - info: [DB Janitor] Started
9 19:27:44 - info: [Ghost Content Manager] (transparent) Initialized
10 19:27:44 - debug: [Ghost Content Manager] (transparent) Added root folder
11 19:27:44 - debug: [Ghost Content Manager] (transparent) Added root folder
12 19:27:44 - info: [Skills] Initiated
13 19:27:44 - info: Loaded @botpress/audience, version 10.40.0
14 19:27:44 - verbose: [Renderers] Enabled for webchat.
15 19:27:44 - info: Loaded @botpress/channel-web, version 10.40.0
16 19:27:44 - info: Loaded @botpress/hitl, version 10.40.0
17 19:27:44 - debug: [Ghost Content Manager] (transparent) Added root folder
18 19:27:44 - debug: [Ghost Content Manager] (transparent) Added root folder
19 19:27:44 - info: Loaded @botpress/nlu, version 10.40.0
20 19:27:44 - info: Loaded @botpress/skill-choice, version 10.40.0
21 19:27:44 - warn: DEPRECATION NOTICE - bp.umm is deprecated and will be rem
22 19:27:44 - warn: DEPRECATION NOTICE - bp.umm is deprecated and will be rem
23 19:27:44 - verbose: [Renderers] Enabled for rocketchat.
24 19:27:44 - info: Loaded botpress-channel-rocketchat, version 0.0.14
25 19:27:44 - info: Loaded 6 modules
26 19:27:44 - info: [Skills] Loaded 1 skills
27 19:27:44 - debug: [Ghost Content Manager] (transparent) Added root folder
28 19:27:44 - debug: Loading middleware: rendering.instrumentation
29 19:27:44 - debug: Loading middleware: hitl.captureInMessages
30 19:27:44 - debug: Loading middleware: nlu.incoming
31 19:27:44 - debug: Loading middleware: hear
32 19:27:44 - debug: Loading middleware: hitl.captureOutMessages
33 19:27:44 - debug: Loading middleware: webchat.sendMessages
34 19:27:44 - debug: Loading middleware: rocketchat.sendMessages
35 19:27:44 - debug: Loading middleware: fallback
36 19:27:44 - debug: Loading data for builtin_text from builtin_text.json
37 19:27:44 - info: Read 7 item(s) from builtin_text.json
38 19:27:44 - info: Bot launched. Visit: http://localhost:3000
39 [connect] Connecting { username: 'bot',
40 password: 'pass',
41 ldap: false,
42 host: 'bots.rocket.chat',
43 useSsl: true,
44 timeout: 20000,
45 rooms: [],
46 allPublic: false,
47 dm: false,
48 livechat: false,
49 edited: false,
```

```
50 integrationId: 'js.SDK',
51 roomCacheMaxSize: 10,
52 roomCacheMaxAge: 300000,
53 dmCacheMaxSize: 10,
54 dmCacheMaxAge: 100000 }
55 19:27:44 - debug: Loading data for builtin_image from builtin_image.json
56 19:27:44 - debug: Loading data for builtin_single-choice from builtin_single-choice.json
57 19:27:44 - debug: Loading data for builtin_card from builtin_card.json
58 19:27:44 - debug: Loading data for builtin_action-button from builtin_action-button.json
59 19:27:44 - debug: Loading data for builtin_carousel from builtin_carousel.json
60 19:27:44 - debug: Loading data for builtin_raw from builtin_raw.json
61 19:27:45 - debug: [NLU::Native] Model is up to date
62 19:27:45 - info: -----
63 19:27:45 - info: Webchat available at http://localhost:3000/s/chat
64 19:27:45 - info: -----
65 [connect] Connected
66 [login] Logging in DocsBot
67 [getRoomIdByNameOrId] Calling (caching): general
68 [getRoomIdByNameOrId] Success: "GENERAL"
69 [joinRoom] Calling (async): ["GENERAL"]
70 [joinRoom] Success
71 [subscribe] Preparing subscription: stream-room-messages: __my_messages__
72 [subscribe] Stream ready: 4
73 LISTEN TRIGGERED
74 [reactive] Listening for change events in collection stream-room-messages
```

4. talk to your bot

On the server, login as a regular user (not the BOT user), go to `general` room, and talk to your newly created bot:



Botpress bot responses to user messages

Run a Rasa Bot

Rasa is the leading open-source machine learning toolkit that lets developers expand bots beyond answering simple questions with minimal training data. At the core, Rasa bot has a machine learning model which trained on example conversations.

Rasa is developed with Python but for the most part, you don't need to know Python to design the basic conversational flows. You will need Python knowledge only when creating custom actions in Rasa which call external actions.

Quick start guide

The fastest way to start with Rasa is using [Rasa starter kit](#). The starter kit uses webhooks to communicate between Rasa and Rocket.Chat.

1. clone the starter repo

```
1 git clone https://github.com/RocketChat/rasa-kick-starter  
2 cd rasa-kick-starter/
```

2. configure the bot

- open the `rasa-kick-starter/bot_rasa` folder and update the `credentials.yml` file with Rasa bot's username, password, and Rocket.Chat URL:

```
1 rocketchat:  
2   user: "<RASA USER NAME>"  
3   password: "<RASA USER PASS>"  
4   server_url: "<ROCKETCHAT HOST>"
```

- train the bot's Machine Learning Model:

```
docker run -it -v $(pwd)/bot_rasa:/app rasa/rasa train
```

After the training, the machine learning model will be created inside the `bot_rasa/models` folder.

3. run the bot

```
docker-compose up bot_rasa
```

You should see the following output:

```
1 $ docker-compose up bot_rasa
2 Starting rasa-kick-starter_bot_rasa_1 ... done
3 Attaching to rasa-kick-starter_bot_rasa_1
4 bot_rasa_1           | 2019-10-31 21:23:24 INFO      root  - Starting Rasa
```

Open your browser and navigate to `http://localhost:5005`. You should see the response from the running Rasa bot:

```
Hello from Rasa: 1.5.0a1
```

If you have Rocket.Chat running on the same machine, the bot's URL is `http://bot_rasa:5005`. This guide uses a remote Rocket.Chat instance, so it is necessary to get a public URL for the Rasa bot to be able to link it properly. It is recommended to use [ngrok](#) for this purpose.

[Download ngrok](#), open terminal in the folder you downloaded ngrok to and execute the following command:

```
./ngrok http 5005
```

You should see the following output:

1 Session Status	online
2 Session Expires	7 hours, 59 minutes
3 Version	2.3.35
4 Region	United States (us)
5 Web Interface	http://127.0.0.1:4040
6 Forwarding	http://dde093e1.ngrok.io -> http://localhost
7 Forwarding	https://dde093e1.ngrok.io -> http://localhos

Public URL will be provided next to `Forwarding` column. In this case,
`http://dde093e1.ngrok.io`.

4. configure Rocket.Chat webhook

Go to `Administration > Integrations`. Click `New Integration` button in the top right corner and then select `Outgoing WebHook`.

Select `Message Sent` Event Trigger from the dropdown list. Complete the configuration with the following settings:

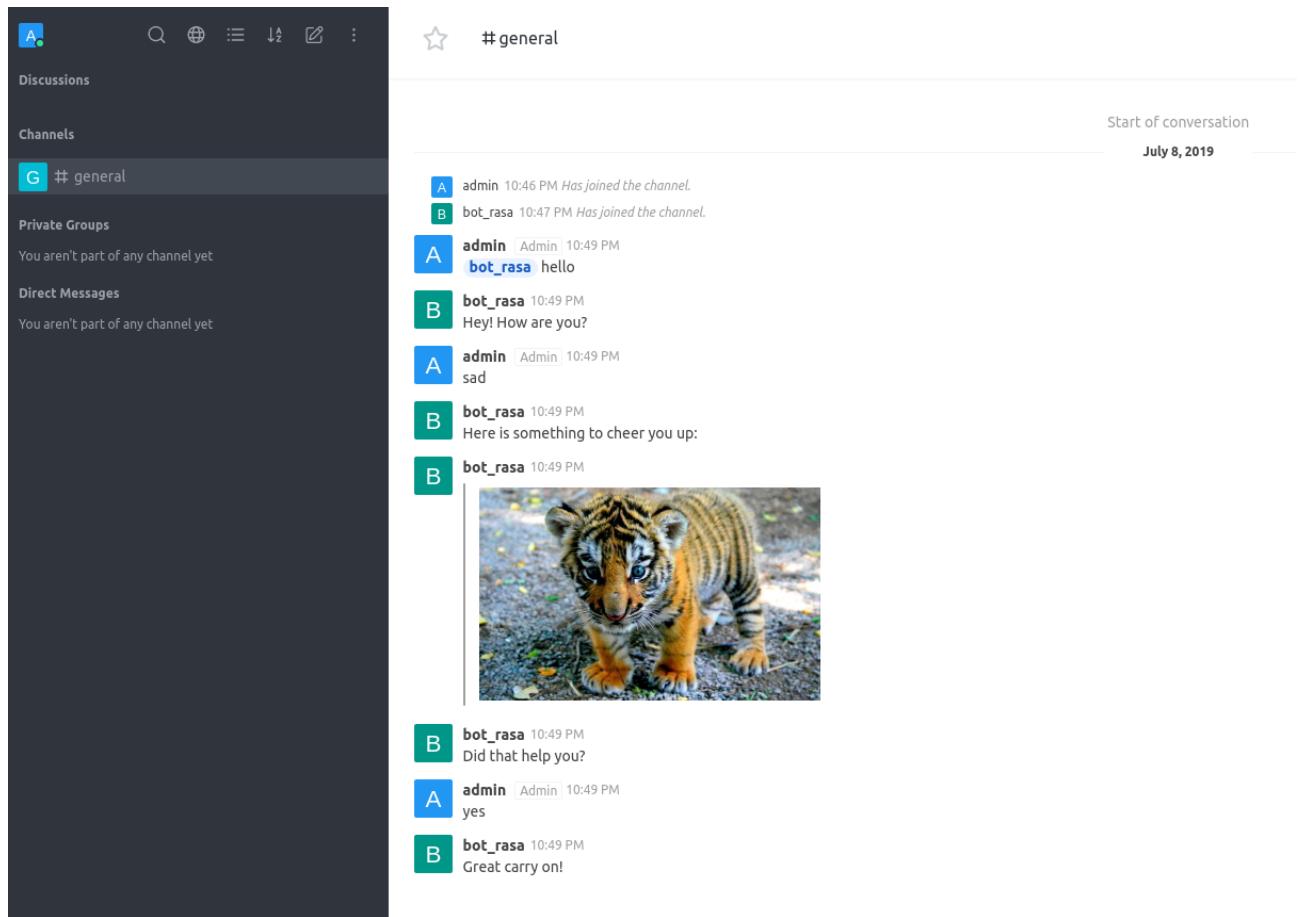
- Enabled: `True`
- Channel: `#general`
- URLs: `http://<ngrok_public_url>/webhooks/rocketchat/webhook`
- Post as: `<RASA USER NAME>`

NOTE: Make sure you replace the URL with a valid public URL obtained on the previous step using ngrok.

Click `Save changes` in the top right corner.

5. talk to your bot

On the server, login as a regular user (not the BOT user), go to `general` room, and try to talk to your bot by typing `@bot_rasa hello`:



Rasa bot is talking

Bot Glossary

What is Livechat inquiry?

A Livechat inquiry is an internal data structure used to control the Omnichannel queue. The conversation state - ready, queued or taken - is a state of the conversation in the queue and kind of data is stored in the inquiry collection. It won't be useful for the bot integration because the Omnichannel routing system will handle all stuff

What is Omnichannel?

Omnichannel is a product name. In older documentation, it is referred to as Livechat. In recent years, the former Livechat feature has been expanded to include Whatsapp and Facebook Messenger. In the near future, the Omnichannel product may include interoperation with other technologies such as Viber, Wechat, Blackberry, Apple Business Message, and RCS.

API

Schema Definition

Rooms

```
1  {
2      "_id": "22nytdn4QqKvSk2Av",           // Random.id()
3      "t": "p",                            // String          Room Type:
4      "ts": new Date(1432937851208),       // Date           Room Creation
5      "name": "general",                  // String          Room Name
6      "lm": new Date(1432937851208),       // Date           Last Message
7      "msgs": 2345,                      // Integer         Messages Count
8      "cl": true,                        // Boolean         If users can
9      "ro": false,                       // Boolean         Read Only
10     "usernames": [                     // Array(String) Room Users
11         "username1",                  // String          User Username
12         "username2",                  // String          User Username
13         "username3"                  // String          User Username
14     ],
15     "u": {                           // Object         Owner User
16         "_id": "CABrXSHpDqbZXGkYR",    // Random.id()   User Id
17         "username": "john"           // String          User Username
18     },
19     "customFields": {                // Object         User defined
20         "userDefinedField": "userValue", //               User defined
21         "userDefinedField2": true,      //               User defined
22         "userObject3": {             //               User defined
23             "a": "hello",            //               User defined
24             "b": "lalala"           //               User defined
25         }
26     }
27 }
```

Fields

Fields	Type	Description
_id	Random.id()	Room Id
t	String	Room Type
ts	Date	Room Creation Timestamp

name	String	Room Name (t:d -> empty)
lm	Date	Last Message Timestamp
msgs	Integer	Messages Counter
cl	Boolean	If users can leave room
ro	Boolean	Read Only
usernames	Array[String]	Room Users
usernames.0	String	User Username
u	Object	Owner User
u._id	Random.id()	User Id
u.username	String	User Username
customFields	Object	User custom fields (for t:c and t:p only)

Indexes

Fields	Options
name	unique, sparse
u._id	

Subscriptions

```

1  {
2    "_id": "22nytdn4QqKvSk2Av",           // Random.id()
3    "t": "p",                            // String      Subscription
4    "ts": new Date(1432937851208),       // Date        Subscription
5    "ls": new Date(1432937851208),       // Date        Last Seen T
6    "name": "my-cool-friends",           // String      Subscription

```

```

7   "rid": "25fkmHMDeFp57ZqYg",           // Random.id()      Room Id
8   "f": true,                           // Boolean          Favorited
9   "open": true,                        // Boolean          Is Room Ope
10  "alert": true,                       // Boolean          Room has un
11  "unread": 5,                         // Integer          Counter of
12  "u": {                                // Object           User
13    "_id": "CABrXSHpDqbZXGkYR",        // Random.id()      User Id
14    "username": "liam"                 // String           User Username
15  },
16  "v": {                                // Object           Visitor
17    "_id": "CABrXSHpDqbZXGkYR",        // Random.id()      Visitor Id
18    "username": "anonymous"           // String           Visitor Use
19  },
20  "customFields": {                     // Object           User define
21    "userDefinedField": "userValue",   //               User define
22    "userDefinedField2": true,         //               User define
23    "userObject3": {                  //               User define
24      "a": "hello",                //
25      "b": "lalala"                //
26    }
27  }
28 }

```

Fields

Fields	Type	Description
_id	Random.id()	Subscriptions Id
t	String	Subscription Type (copy from Room)
ts	Date	Subscription Creation Timestamp
ls	Date	Last Seen Timestamp
name	String	Subscription Name (t:d -> target username)
rid	Random.id()	Room Id
f	Boolean	Favorited
open	Boolean	Is Room Opened
alert	Boolean	Room has unread messages for this user

unread	Integer	Counter of unread messages with mentions or from direct messages
u	Object	User
u._id	Random.id()	User Id
u.username	String	User Username
v	Object	Visitor
v._id	Random.id()	Visitor Id
v.username	String	Visitor Username
customFields	Object	User custom fields

Notes about customFields:

- customFields inherits from room's customFields for channels (Room Type: c = channel) and groups (Room Type: p = group) and changes with room's customFields
- customFields inherits from user's customFields for Direct Messages (Room Type: d = direct) and changes with user's customFields. Note that users of Direct Messages room will have own customFields.

Indexes

Fields	Options
rid, u._id	unique
u._id, name, t	unique
open	
alert	
unread	

Messages

```
1  {
2    "_id": "22nytdn4QqKvSk2Av",           // Random.id()
3    "t": "p",                            // String      Message Type
4    "ts": new Date(1432937851208),       // Date        Message Create
5    "rid": "25fkmHMDeFp57ZqYg",          // Random.id() Room Id
6    "msg": "Yeah, doesn't seem to like it", // String      Message Body
7    "url": [                             // Array(String) Message URL
8      "http://google.com/"
9    ],
10   "expireAt": new Date(1432937951208), // Date        Message auto-expire
11   "mentions": [                        // Array(String) Mentioned Users
12     "username1"
13   ],
14   "u": {                             // Object      User
15     "_id": "CABrXSHpDqbZXGkYR",       // Random.id() User Id
16     "username": "john"                // String      User Username
17   },
18   "v": {                             // Object      Visitor
19     "_id": "CABrXSHpDqbZXGkYR",       // Random.id() Visitor Id
20     "username": "anonymous"          // String      Visitor User
21 }
22 }
```

Indexes

Fields	Options
rid, ts	
expireAt	expireAfterSeconds

Visitors

```
1  {
2    "_id": "22nytdn4QqKvSk2Av",           // Random.id()
3    "name": "John",                      // String      Name
4  }
```

```
4   "email": "a@d.com",           // String      Email
5   "phone": "555 876333443",     // String      Phone
6   "token": "token"             // String      Token
7 }
```

Indexes

User

The **IUser** interface represents the **User** object stored in the collection `users`

```
1  interface IUser {
2      _id: string;
3      createdAt: Date;
4      roles: string[];
5      type: string;
6      active: boolean;
7      username?: string;
8      name?: string;
9      services?: IUserServices;
10     emails?: IUserEmail[];
11     status?: string;
12     statusConnection?: string;
13     lastLogin?: Date;
14     avatarOrigin?: string;
15     utcOffset?: number;
16     language?: string;
17     statusDefault?: string;
18     oauth?: {
19         authorizedClients: string[];
20     };
21     _updatedAt?: Date;
22     statusLivechat?: string;
23     e2e?: {
24         private_key: string;
25         public_key: string;
26     };
27     requirePasswordChange?: boolean;
28     customFields?: {
29         [key: string]: any;
30     };
31     settings?: IUserSettings;
32 }
```

The following sub-interfaces are used by the **IUser** interface

```
1  interface ILoginToken {
2      hashedToken: string;
3      twoFactorAuthorizedUntil?: Date;
```

```
4     twoFactorAuthorizedHash?: string;
5   }
6
7   interface IMeteorLoginToken extends ILoginToken {
8     when: Date;
9   }
10
11  interface IPersonalAccessToken extends ILoginToken {
12    type: 'personalAccessToken';
13    createdAt: Date;
14    lastTokenPart: string;
15    name?: string;
16    bypassTwoFactor?: boolean;
17  }
18
19  interface IUserEmailVerificationToken {
20    token: string;
21    address: string;
22    when: Date;
23  }
24
25  interface IUserEmailCode {
26    code: string;
27    expire: Date;
28  }
29
30  type LoginToken = ILoginToken & IPersonalAccessToken;
31
32  interface IUserServices {
33    password?: {
34      bcrypt: string;
35    };
36    email?: {
37      verificationTokens?: IUserEmailVerificationToken[];
38    };
39    resume?: {
40      loginTokens?: LoginToken[];
41    };
42    google?: any;
43    facebook?: any;
44    github?: any;
45    totp?: {
46      enabled: boolean;
47      hashedBackup: string[];
48      secret: string;
49    };
50    email2fa?: {
51      enabled: boolean;
52      changedAt: Date;
53    };
54    emailCode: IUserEmailCode[];
55  }
56
```

```
55  }
56
57 interface IUserEmail {
58     address: string;
59     verified: boolean;
60 }
61
62 interface IUserSettings {
63     profile: any;
64     preferences: {
65         [key: string]: any;
66     };
67 }
```

Room

The room object represents any form of communication between two or more users. There are at least two fields on the room object.

- `_id` : The room / user id (depends on the room type)
 - `t` : The room type
-

Room Types

- `d` : Direct chat
- `c` : Chat
- `p` : Private chat
- `l` : Livechat

The information that comes with the room object changes according to its type.

Direct chat

A direct chat between two users. In this case the room object only has two fields:

- `_id` : The room id
- `t` : The room type (in this case `d`)

Example:

```
1  {
2      "_id": "room-id",
3      "t": "d"
4 }
```

You'll need to access the room information using data from [Get Subscriptions](#).

Chat

An open chatroom. A chat contains more information about the room as it follows:

- `_id` : The room id
- `t` : The room type (in this case `c`)
- `name` : The room name
- `u` : The room creator (it may return a null user)
- `topic` : (Optional) The room topic
- `muted` : (Optional) A collection of muted users by its username
- `jitsiTimeout` : (Optional) (?)

Example:

```
1  {
2    "_id": "room-id",
3    "t": "c",
4    "name": "room-name",
5    "u": { "_id": "user-id", "username": "username" },
6    "topic": "room-topic",
7    "muted": [ "username" ],
8    "jitsiTimeout": { "$date": 1480377601 }
9 }
```

Private chat

A private chatroom. This type of room resembles the open chat room with an addition.

- `_id` : The room id
- `t` : The room type (in this case `p`)
- `name` : The room name
- `u` : The room creator (it may return a null user)
- `topic` : (Optional) The room topic
- `muted` : (Optional) A collection of muted users by its username
- `jitsiTimeout` : (Optional) (?)
- `ro` : Flags if the room is read-only

Example:

```
1  {
2    "_id": "room-id",
3    "t": "p",
4    "name": "room-name",
5    "u": { "_id": null, "username": null },
6    "topic": "room-topic",
7    "ro": false // This room is not read-only
8 }
```

Subscriptions

Message

The message object is the very soul of a conversation. It encapsulates all the information need in order to represent a single entry on a message list.

The message object contains these fields:

- `_id` : The message id
- `rid` : The room id - Identify the room the message belongs
- `msg` : The textual message
- `ts` : The message time stamp (date of creation on client)
- `u` : The user that sent the message
- `_updatedAt` : The time stamp when the message got saved on the server
- `editedAt` : *(Optional)* The time stamp of when the message was edited
- `editedBy` : *(Optional)* The user that edited the message
- `urls` : *(Optional)* A collection of URLs metadata. Available when the message contains at least one URL
- `attachments` : *(Optional)* A collection of **attachment objects**, available only when the message has at least one attachment
- `alias` : *(Optional)* A way to display the message is "sent" from someone else other than the user who sent the message
- `avatar` : *(Optional)* A url to an image, that is accessible to anyone, to display as the avatar instead of the message user's account avatar
- `groupable` : *(Optional)* Boolean that states whether or not this message should be grouped together with other messages from the same user
- `parseUrls` : *(Optional)* Whether Rocket.Chat should try and parse the urls or not

The user presented on `u` and `editedBy` fields are a simplified version of the user information:

- `_id` : The user id
- `username` : The username

The URL metadata contains several informational fields:

- `url` : The URL itself (just as it appears on the message)
- `meta` : URL metadata (varies accord to the URL)
- `headers` : Some HTTP headers (varies accord to the URL)
- `parsedUrl` : The parsed URL broken into its parts

The attachment object is fully described here

```

1  {
2      "messages": [
3          {
4              "_id": "message-id",
5              "rid": "room-id",
6              "msg": "Hello World!",
7              "ts": { "$date": 1480377601 },
8              "u": {
9                  "_id": "user-id",
10                 "username": "username"
11             },
12             "_updatedAt": { "$date": 1480377601 }
13         },
14         {
15             "_id": "message-id",
16             "rid": "room-id",
17             "msg": "Hello!",
18             "ts": { "$date": 1480377601 },
19             "u": {
20                 "_id": "user-id",
21                 "username": "username"
22             },
23             "_updatedAt": { "$date": 1480377601 },
24             "editedAt": { "$date": 1480377601 },
25             "editedBy": {
26                 "_id": "user-id",
27                 "username": "username"
28             }
29         }
30     ]
31 }
```

Schema Definition v2 (Draft)

 This is a work in progress

Pull Requests related to the schema v2

- <https://github.com/RocketChat/Rocket.Chat/pull/15711>

Events

Events are the core of our system now, everything goes through the events, which are unique and federable.

EventContext

The event context holds all the possible context types, currently we only have the `room` context, and it is defined as an ENUM, as seem below.

```
1  export enum EventContext {
2      ROOM = 'room',
3 }
```

IEvent

Interface Definition

```
1 interface IEvent<T extends EDataDefinition> {
2     _id: string;
3     clid?: string;
4     pids: Array<string>;
5     v: number;
6     ts: Date;
7     src: string;
8     ct: EventContext;
9     cid: string;
10    t: EventTypeDescriptor;
11    dHash: string;
12    o: T;
13    d: T;
14    isLeaf?: boolean;
15    deletedAt?: Date;
16 }
```

Property Description

_id	The event id, which is a SHA256 hash of the <code>src</code> , <code>ct</code> , <code>cid</code> , <code>_pids</code> , <code>t</code> , <code>ts</code> and <code>dHash</code>
-----	--

clid	This is the "id generated by the client", on V1 our clients generate the <code>_id</code> of the message, so this needs to be here for backward compatibility
------	---

pids	The ids of the previous events
------	--------------------------------

v	The version of the schema
---	---------------------------

src	The source of this event, which server, this will be used to federate events later on
-----	---

ct	This is part of the context, holds the type of the context (for example, "room"), more in EventContext
----	--

cid	This is also part of the context, holds the of the context (like the room id)
t	The type of the event, details in EventTypeDescriptor
dHash	The hash of some or all of the properties on <code>d</code> , depending on the event type
o	The event's data when it was created, this never changes and it is used to calcute the hash when a integrity verification is realized
d	The event's data, the payload, details in EventDataDefinition , this can change when and event is updated using IEventDataUpdate
isLeaf	This will on appear when it is set to true and determines wheter or not this is the latest item of a chain, the leaf of the tree. More than one event may have the <code>isLeaf</code> flagged as true in the same context
deletedAt	The date that this event was deleted

Example of types "room" and "msg":

```

1  {
2      room_events: [
3          {
4              "_id": "8aa776ae767c37254c6a85f914a6151d3bb558ff0a9d639f13f0fe
5              "pids": [],
6              "v": 2,
7              "ts": ISODate("2020-06-11T19:46:40.192Z"),
8              "src": "peerc.allskar.com",
9              "ct": "room",
10             "cid": "a7c5MQFQGe4XKMyEo",
11             "t": "room",
12             "dHash": "469d7080b26464d8e684dc72c409dd669676ff0e6bbdd4b6f339
13             "o": {
14                 ...
15             },
16             "d": {
17                 ...
18             }
19         },
20         {
21             "_id": "f53baadb1090c2b4f9d445d030902e142065b5e177a50799c2cc6b
22             "clid": "D2Hznvc4jt7YRSaQy",
23             "pids": [

```

```
24          "8aa776ae767c37254c6a85f914a6151d3bb558ff0a9d639f13f0fe5f1
25      ],
26      "v": 2,
27      "ts": ISODate("2020-06-11T19:46:42.385Z"),
28      "src": "localhost",
29      "ct": "room",
30      "cid": "a7c5MQFQGe4XKMyEo",
31      "t": "msg",
32      "dHash": "793dd0f58f7f9b243ecb35fed189260633058e0ee1468813148b
33      "o": {
34          ...
35      },
36      "d": {
37          ...
38      }
39  }
40 ]
41 }
```

EventContext

The context is important to be able to differently handle any number of types of events, even determine which ones need to be federated, for example.

The EventContext is an enum, ready to accept more context types, as seen here:

```
1  export enum EventContext {  
2      ROOM = 'room',  
3  }
```

EventTypeDescriptor

The EventTypeDescriptor is an enum:

```
1  export enum EventTypeDescriptor {
2      PING = 'ping'
3  }
```

Descriptor Why does it exist?

PING The ping event is used to determine whether or not the server is alive. Still not being used, but it will be very important when federation is up using the new event system.

EventMessageTypeDescriptor

The EventMessageTypeDescriptor is an enum:

```
1  export enum EventMessageTypeDescriptor {
2      MESSAGE = 'msg',
3      MESSAGE_PINNED = 'message_pinned',
4      DISCUSSION_CREATED = 'discussion-created',
5  }
```

Descriptor	Why does it exist?
MESSAGE	A regular message, nothing special
MESSAGE_PINNED	A message that was pinned to a channel
DISCUSSION_CREATED	When a discussion is created, this message type is assigned

EventDataDefinition

This is a merge of all possible event payloads, like [IEventDataEmpty](#) and [IEventDataUpdate](#), it also comprehends secondary payloads, like those present in [RoomEventDataDefinition](#).

IEventDataEmpty

When the payload on an event should be empty, this data type is used. And it is what it appears to be: empty.

IEventDataUpdate

The interface definition is:

```
1  export interface IEventDataUpdate<T extends EventDataDefinition> {  
2      [key: string]: T;  
3 }
```

IRoomEvent

Interface Definition

```
1  export interface IRoomEvent {  
2      ct: EventContext.ROOM;  
3      t: RoomEventTypeDescriptor;  
4      d: EventRoomDataDefinition;  
5  }
```

Property Description

ct	This is the context type, hardcoded to <code>EventContext.ROOM</code> (EventContext), because this is the room specialized event.
t	The type of the event, in this case, one of the possibilities described on RoomEventTypeDescriptor
d	The payload of the event, one of the RoomEventDataDefinition

RoomEventTypeDescriptor

The RoomEventTypeDescriptor is an enum:

```
1  export enum RoomEventTypeDescriptor {
2      ROOM = 'room',
3      DELETE_ROOM = 'droom',
4      PRUNE_ROOM_MESSAGES = 'prune',
5      MESSAGE = 'msg',
6      EDIT_MESSAGE = 'emsg',
7      DELETE_MESSAGE = 'dmsg',
8  }
```

Descriptor	Why does it exist?
ROOM	This is the beginning of the history of a room, the genesis of that context (and context id). The payload can be seen IEventDataRoom
DELETE_ROOM	When a room is deleted, an event of this type is generated. This event has no payload, so it uses IEventDataEmpty
MESSAGE	Every sent message generates an event of type MESSAGE, check more of the payload in IEventDataMessage
EDIT_MESSAGE	When the message's text is edited, or a reaction is added, or any other kind of update happens, this event is generated. It uses the same payload as the MESSAGE event, but wrapped like IEventDataUpdate<IEventDataMessage>
DELETE_MESSAGE	When a message is deleted, an event of this type is generated. This event has no payload, so it uses IEventDataEmpty

RoomEventDataDefinition

Check this item's child sections to learn more about all the possible payloads.

IRoomEventDataRoom

The definition is:

```
1  export interface IEventDataRoom {
2      room: IRoom;
3  }
```

Property What is it?

room Holds an [IRoom](#) interface, which is related to the legacy [Room](#) definition.

IRoomEventDataMessage

This payload holds the necessary message properties, the definition is:

```
1  export interface IEventDataMessage {  
2      t: EventMessageTypeDescriptor;  
3      u: IUser;  
4      msg: string;  
5      mentions?: Array<string>;  
6      channels?: Array<string>;  
7      reactions?: Array<object>;  
8      drid?: string;  
9      file?: {  
10          _id: string;  
11          name: string;  
12          type: string;  
13      };  
14      pinned?: Array<object>;  
15      starred?: Array<object>;  
16      deleted?: boolean;  
17 }
```

Property	What is it?
t	The type descriptor of the message, more in EventMessageTypeDescriptor
u	This holds the user which emitted this message, you can see the definition in IUser
msg	The actual message, the text
mentions	[OPTIONAL] Holds the user mentions related to this message
channels	[OPTIONAL] Holds the channel mentions related to this message
drid	[OPTIONAL] Holds the discussion room id, the discussion which this message is attached to
file	[OPTIONAL] If there is an attachment, the file information will be here
pinned	[OPTIONAL] <add description>

starred [OPTIONAL] <add description>

deleted [OPTIONAL] Was this message deleted?

IFederationInfo

IMessage

IRoom

Interface Definition

```
1  export interface IRoom {  
2      _id: string;  
3      federation: IFederationInfo;  
4 }
```

IUser

Interface Definition



Realtime API

Point your client to the Websocket of the server you want to connect to:

```
wss://[ABC.DOMAIN.COM]/websocket
```

Our real-time API is composed of two elements: **Method Calls** and **Subscriptions**. Both of them are supported directly in the websocket connection.

To make it possible to have everything working on the same connection we use RPC with the following format.

```
1  {
2      "msg": "type-of-communication",
3      "id": "unique-id",
4      ... // per call defined data
5 }
```

The type of communication is defined according to the call:

- **Method Calls:** method
- **Subscriptions:** sub

Please note, the server will send you "ping" and you must respond with "pong" otherwise the server will close the connection.

Before requesting any method / subscription you have to send a connect message:

```
1  {
2      "msg": "connect",
3      "version": "1",
4      "support": ["1"]
5 }
```

Resources

- A basic example script that uses the 'ddp' NodeJS package to subscribe to the Realtime-API stream of a Group/Channel here
<https://github.com/jszaszvari/rocketchat-ddp-listener>
- `Rocket.Chat.RealTime.API.RxJS` Abstraction for Utilizing `Rocket.Chat`'s Realtime API Methods with RxJS. <https://github.com/inf3cti0n95/Rocket.Chat.RealTime.API.RxJS>

Method Calls

TODO: List all the calls and possible responses - as directories

Method calls are used to trigger actions based on the passed data. The response to any method call is completely asynchronous and there's no way to guarantee the order on the fulfillment of the calls. Because of that, it is really important that a `unique-id` is used on the method call since the same ID will be used on the response so the client may know the result of a call.

Example: Create a private room

Method call

```
1  {
2    "msg": "method",
3    "id": "42",
4    "method": "createPrivateGroup",
5    "params": [ "example-private-room", [], false ]
6 }
```

Response

```
1  {
2    "msg": "result",
3    "id": "42",
4    "result": { "rid": "server-generated-room-id" }
5 }
```

The provided ID is the only way to recognize to which call a response is been given.

Archive Rooms

Archiving a room marks it as read only and then removes it from the channel list on the left.

Requirements

Logged In	Permission	Setting
Yes	archive-room	<i>none</i>

Payload

1. String - the id of the room to archive
-

Example Call

```
1  {
2      "msg": "method",
3      "method": "archiveRoom",
4      "id": "97",
5      "params": [
6          "roomId"
7      ]
8  }
```

Example Response

```
1  {
2    "msg": "result",
3    "id": "97"
4 }
```

See Also

- [Create Channels](#)
- [Create Private Groups](#)
- [Delete Rooms](#)
- [Unarchive Rooms](#)
- [Save Room Settings](#)

Create Channels

Creates a public channel.

Requirements

Logged In	Permission	Setting
Yes	create-c	<i>none</i>

Payload

1. String - name of the channel
 2. Array of strings - usernames of the people to add to the channel when it is created. This can be empty and only the caller of the method will be a member of the channel.
 3. Boolean - whether the channel is read only or not
-

Example Call

```
1  {
2      "msg": "method",
3      "method": "createChannel",
4      "id": "85",
5      "params": [
6          "channel-name",
7          ["array-of-usernames", "who-are-in-the-channel"],
8          true/false
9      ]
10 }
```

Example Response

The response will be an object with one property `rid` which is the id of the newly created room.

```
1  {
2      "msg": "result",
3      "id": "85",
4      "result": [
5          { "rid": "BBkfgYT2azf7RPTTg" }
6      ]
7 }
```

See Also

- [Create Private Groups](#)
- [Delete Channels](#)
- [Archive Rooms](#)
- [Unarchive Rooms](#)

Create Direct Message

Prior to sending a direct message to a user, a direct message should be created. This will return a roomID which you can then use to send future messages.

The only parameter you need to send is the desired target username as follows:

```
1  {
2      "msg": "method",
3      "method": "createDirectMessage",
4      "id": "42",
5      "params": ["username-goes-here"]
6 }
```

The response is as follows:

```
1  {
2      "msg": "result",
3      "id": "42",
4      "result": {
5          "rid": "room-id-would-be-here"
6      }
7 }
```

Notes

Making this call additional times, if a room has already been established, does not result in an error. In addition, the same roomID is returned. This means it's safe to use this method to obtain the roomID every time prior to making calls to loadHistory.

Create Private Groups

Creates a private group.

Requirements

Logged In	Permission	Setting
Yes	create-p	<i>none</i>

Payload

1. String - name of the channel
 2. Array of strings - usernames of the people to add to the private group when it is created. This can be empty and only the caller of the method will be a member of the group.
-

Example Call

```
1  {
2      "msg": "method",
3      "method": "createPrivateGroup",
4      "id": "89",
5      "params": [
6          "channel-name",
7          ["array-of-usernames", "who-are-in-the-channel"]
8      ]
9  }
```

Example Response

The response will be an object with one property `rid` which is the id of the newly created private group.

```
1  {
2      "msg": "result",
3      "id": "89",
4      "result": [
5          { "rid": "BBkfgYT2azf7RPTTg" }
6      ]
7 }
```

See Also

- [Create Channels](#)
- [Delete Private Groups](#)
- [Archive Rooms](#)
- [Unarchive Rooms](#)

Delete Message

There is only way one to delete a message inside of Rocket.Chat, but it only requires the message's `_id` being passed in.

Requirements

Logged In	Permission	Setting
Yes	<code>delete-message</code>	<code>Message_AllowDeleting</code> - "Allow Message Deleting"

Example Call

All that is needed to delete a message is passing the `_id` of the message.

```
1  {
2    "msg": "method",
3    "method": "deleteMessage",
4    "id": "42",
5    "params": [ { "_id": message_id } ]
6 }
```

Example Response

```
1  {
2    "msg": "result",
3    "id": "42",
4    "result": []
5 }
```

Additional Information

As mentioned in requirements, you must be logged in to be able to delete a message. Two settings apply, `Message_AllowDeleting` and then

`Message_AllowDeleting_BlockDeleteInMinutes`. The first setting is a boolean, true/false, and the second setting is an integer that can be `0` for always being allowed to delete or it can be greater than zero which the deleting will be disallowed/blocked after the time has passed.

Having the permission `delete-message` will allows deletion of any message, but still respects `Message_AllowDeleting_BlockDeleteInMinutes` setting.

Having the permission `force-delete-message` allows the deletion of any message, and it ignores the setting `Message_AllowDeleting_BlockDeleteInMinutes`.

See Also

- [Send Message Method](#)
- [Update Message Method](#)

Delete Rooms

Deleting a room, either a private group or public channel, is actually completed via the method `eraseRoom`. The user deleting the room must have permission to do so, by either being owner or admin.

Requirements

Logged In	Permission	Setting
Yes	<code>delete-c</code> <i>or</i> <code>delete-p</code>	<i>none</i>

Payload

1. String - the id of the room to delete
-

Example Call

```
1  {
2      "msg": "method",
3      "method": "eraseRoom",
4      "id": "92",
5      "params": [
6          "roomId"
7      ]
8  }
```

Example Response

```
1  {
2      "msg": "result",
3      "id": "92",
4      "result": 1
5 }
```

See Also

- [Create Channels](#)
- [Create Private Groups](#)
- [Archive Rooms](#)
- [Unarchive Rooms](#)
- [Save Room Settings](#)

End-to-End Encryption API

Fetch My Keys

This method returns the user's private and public keys stored on the server.

Requirements

Logged In	Permission	Setting
Yes	<i>none</i>	<i>none</i>

Example Call

```
1  {
2      "msg": "method",
3      "method": "e2e.fetchMyKeys",
4      "id": "8",
5      "params": []
6 }
```

Example Response

```
1  {
2      "msg": "result",
3      "id": "8",
4      "result": {
5          "public_key": "{\"alg\":\"RSA-OAEP-256\",\"e\":\"AQAB\",\"ext\":true}",
6          "private_key": "{$binary}:"PGi/iGgGPHK0kACoLE+EGIrP+c/e+a64z2N
7      }
8 }
```

Change Log

Version	Description
0.70.0	Added

Get Users of Room Without Key

This method returns all users of an encrypted room that don't yet have the E2E key on their subscription.

Requirements

Logged In	Permission	Setting
Yes	<i>none</i>	<i>none</i>

Payload

1. String - the id of the room to get the users from.
-

Example Call

```
1  {
2      "msg": "method",
3      "method": "e2e.getUsersOfRoomWithoutKey",
4      "id": "8",
5      "params": [
6          "rid"
7      ]
8  }
```

Example Response

```
1  {
2      "msg": "result",
3      "id": "8",
4      "result": {
5          "users": [
6              {"_id": "AAoQ7b25AAcJQryY",
7                  "e2e": {
8                      "public_key": "{\"alg\": \"RSA-OAEP-256\", \"e\": \"AQAB\", \"e2e\": \"wAdMMfMhp7KBiKjuz\", \"public_key\": \"\"}"]
9                  }
10             },
11             {"_id": "wAdMMfMhp7KBiKjuz",
12                 "e2e": {
13                     "public_key": "{\"alg\": \"RSA-OAEP-256\", \"e\": \"AQAB\", \"e2e\": \"\"}"]
14                 }
15             }
16         }
17     }
```

Change Log

Version	Description
0.70.0	Added

Set Room Key ID

This method marks an existing room as encrypted.

Requirements

Logged In	Permission	Setting
Yes	<i>none</i>	<i>none</i>

Payload

1. String - the id of the room that is being updated.
 2. String - the keyID to add to the room.
-

Example Call

```
1  {
2      "msg": "method",
3      "method": "e2e.setRoomKeyID",
4      "id": "8",
5      "params": [
6          "rid",
7          "keyID"
8      ]
9  }
```

Example Response

```
1  {
2    "msg": "result",
3    "id": "8",
4    "result": 1
5 }
```

Change Log

Version	Description
0.70.0	Added

Set User Public and Private Keys

This method changes the user private and public keys on the server.

Requirements

Logged In	Permission	Setting
Yes	<i>none</i>	<i>none</i>

Payload

1. Object - An object containing the `private_key` and `public_key` attributes, both of them Strings.
-

Example Call

```
1  {
2      "msg": "method",
3      "method": "e2e.setUserPublicAndPrivateKeys",
4      "id": "8",
5      "params": [
6          "public_key": "{\"alg\":\"RSA-OAEP-256\",\"e\":\"AQAB\",\"ext\":true}",
7          "private_key": "{\"$binary\":\"kpB3QnQ12JSiUtJziTGv3r4MFgN9ZrpcKKeM\"}"
8      ]
9  }
```

Example Response

```
1  {
2    "msg": "result",
3    "id": "8"
4 }
```

Change Log

Version	Description
0.70.0	Added

Update Group Key

This method updates the user subscription to save the E2E key

Requirements

Logged In	Permission	Setting
Yes	<i>none</i>	<i>none</i>

Payload

1. String - the room id of the subscription
 2. String - the user id of the subscription
 3. String - the key that will be saved on the subscription
-

Example Call

```
1  {
2      "msg": "method",
3      "method": "e2e.updateGroupKey",
4      "id": "8",
5      "params": [
6          "rid",
7          "uid",
8          "key"
9      ]
10 }
```

Example Response

```
1  {
2      "msg": "result",
3      "id": "8",
4      "result": {
5          "_id": "hBGYqzTeAkiMvpM8S",
6          "open": true,
7          "alert": false,
8          "unread": 0,
9          "userMentions": 0,
10         "groupMentions": 0,
11         "ts": {"$date": 1542114790461},
12         "rid": "iKPufpLHcHaQLF6Bv",
13         "name": "a1a",
14         "fname": "a1a",
15         "customFields": {},
16         "t": "p",
17         "u": {
18             "_id": "AAoQ7b25AAcJQryY",
19             "username": "usuario3-1",
20             "name": null
21         },
22         "ls": {
23             "$date": 1542114790461
24         },
25         "_updatedAt": {
26             "$date": 1542114878329
27         },
28         "roles": ["owner"],
29         "E2EKey": "eyJhbGciOiJBT4JwSCA+9LNyOPfSw94DhcJNgUpzfXsNABiHEV7W01Uw
30     }
31 }
```

Change Log

Version	Description
0.70.0	Added

Favoriting Rooms

When a user makes a room as a favorite, the yellow star appears and it moves the room up to the "favorites" section of the list of rooms.

Requirements

Logged In	Permission	Setting
Yes	<i>none</i>	<i>none</i>

Payload

1. String - the id of the room to leave
 2. Boolean - whether the room is a favorite or not, defaults to true
-

Example Call

```
1  {
2      "msg": "method",
3      "method": "toggleFavorite",
4      "id": "16",
5      "params": [
6          "roomId",
7          true/false
8      ]
9  }
```

Example Response

```
1  {
2      "msg": "result",
3      "id": "16",
4      "result": 1
5 }
```

See Also

- [Joining Channels](#)
- [Leaving Rooms](#)

Get Permissions

Use this call to get a collection with all the permissions of the server. Each permission will have the roles it applies to.

You may use this information to change your UI according to the permissions a user has (hiding what he can't do for example).

Example call:

```
1  {
2      "msg": "method",
3      "method": "permissions/get",
4      "id": "42",
5      "params": []
6 }
```

Response:

```
1  {
2      "msg": "result",
3      "id": "42",
4      "result": [
5          ... // permissions
6      ]
7 }
```

The Permission object

The permission object describes a permission as:

- `_id` : The permission's id
- `roles` : A collection of roles that this permission applies to

- `_updatedAt` : (Optional) The last time this permission object was updated in the database
- `meta` : Metadata about the permission (described below)
- `$loki` : An internal property which can be ignored (it is removed on the web client before being added to the web client's database)

The `meta` object is defined as:

- `revision` : The revision of the permission
- `created` : Date the permission was created
- `version` : The permission version
- `updated` : (Optional) Date the permission was last updated

Example:

```
1  {
2      "_id": "snippet-message",
3      "roles": [
4          "owner",
5          "moderator",
6          "admin"
7      ],
8      "_updatedAt": { "$date": 1480377601 },
9      "meta": {
10          "revision": 3,
11          "created": 1480377601,
12          "version": 0,
13          "updated": 1480377601
14      },
15      "$loki": 1
16 }
```

Get Public Settings

This method is used to retrieve the public settings, such as Site Name. It accepts a timestamp as the first and only parameter which causes the results to be an object that contains the updated and removed settings after the provided time. If you want to retrieve all of the public settings, just omit the parameter.

Example call to retrieve everything

```
1  {
2      "msg": "method",
3      "method": "public-settings/get",
4      "id": "42"
5 }
```

Response

The `_id` value is id of the setting and is how it should be referenced. The `value` is exactly what the name describes, the value of the setting.

```
1  [
2      "msg": "result",
3      "id": "42",
4      "result": [
5          {
6              "_id": "uniqueID",
7              "value": "z3cDS83TYjCfHatrF"
8          },
9          {
10              "_id": "Accounts_AllowDeleteOwnAccount",
11              "value": false
12          },
13          ...
14      ]
15 }
```

Example call to retrieve the updated and removed ones since the provided date

```
1  {
2      "msg": "method",
3      "method": "public-settings/get",
4      "id": "42",
5      "params": [ { "$date": 1480377601 } ]
6 }
```

Response

The `_id` value is id of the setting and is how it should be referenced. The `value` is exactly what the name describes, the value of the setting.

```
1  {
2      "msg": "result",
3      "id": "42",
4      "result": {
5          "update": [
6              {
7                  "_id": "uniqueID",
8                  "value": "z3cDS83TYjCfHatrF"
9              },
10             {
11                 "_id": "Accounts_AllowDeleteOwnAccount",
12                 "value": true
13             },
14             ...
15         ],
16         "remove": [
17             {
18                 "_id": "somethingElse",
19                 "value": "valueRemoved"
20             },
21             ...
22         ]
23     }
24 }
```

```
20          "_id": "OldSomething",
21          "value": false
22      },
23      ...
24  ]
25 }
26 }
```

Get Room Roles

This method call is used to get room-wide special users and their roles. You may send a collection of room ids (at least one).

The `result` is a collection of users and its roles per room.

The user roles per room object is defined as:

- `rid` : The room id this user and role belongs to
- `u` : A simple user object with the user id and username
- `roles` : The collection of roles of the user in the room
- `_id` : the id of this object

Example call:

```
1  {
2      "msg": "method",
3      "method": "getRoomRoles",
4      "id": "42",
5      "params": [ "room-id" ]
6 }
```

Response:

```
1  {
2      "msg": "result",
3      "id": "42",
4      "result": [
5          {
6              "rid": "room-id",
7              "u": { "_id": "user-id", "username": "username" },
8              "roles": [ "role-name" ],
9              "_id": "id"
10         }
11     ]
12 }
```

Get Rooms

This is the method call used to get all the rooms a user belongs to. It accepts a timestamp with the latest client update time in order to just send what changed since last call. If it's the first time calling, just send a `0` as date.

The `result` is an object with two fields: `update` and `remove`.

Remove field

The `remove` field is a collection of room id identifying the rooms that were removed from the server.

Update field

The `update` field is a collection of `room` and its content varies according to the `room type`. You can read more about the `room` object on [its own page](#).

Example call

```
1  {
2    "msg": "method",
3    "method": "rooms/get",
4    "id": "42",
5    "params": [ { "$date": 1480377601 } ]
6 }
```

Response

```
1  {
2      "msg": "result",
3      "id": "42",
4      "result": {
5          "update": [
6              ... // rooms
7          ],
8          "remove": [
9              ... // room ids
10         ]
11     }
12 }
```

Get Subscriptions

Returns a `result` with a user's subscription collection. You may pass no params or a date param with the time stamp of your last update. If a date is passed the result will only contains changes to the subscriptions.

Example call:

```
1  {
2    "msg": "method",
3    "method": "subscriptions/get",
4    "id": "42",
5    "params": [ { "$date": 1480377601 } ]
6 }
```

Response:

```
1  {
2    "msg": "result",
3    "id": "42",
4    "result": [
5      ... // subscriptions
6    ]
7 }
```

Subscription Object

The subscription object contains the information about the room and the user relation to it.

- `t` : The room type (the same used on the [room object][1])
- `ts` : Timestamp the room was created at, so this should equal the room's `ts` field
- `ls` : Last seen timestamp (The last time the user has seen a message in the room)
- `name` : The room name

- `rid` : The room id
- `u` : An simple `user` object with its id and username
- `open` : Whether the room the subscription is for has been opened or not (defaults to `false` on direct messages). This is used in the clients to determine whether the user can see this subscription in their list, since you can hide rooms from being visible without leaving them.
- `alert` : Whether there is an alert to be displayed to the user
- `roles` : (Optional) The collection of roles the user belongs to (at least one `role-name` will be present)
- `unread` : The total of unread messages
- `_updatedAt` : Timestamp of when the subscription record was updated
- `_id` : The subscription id

An example:

```

1  {
2    "t": "d",
3    "ts": { "$date": 1480377601 },
4    "ls": { "$date": 1480377601 },
5    "name": "username",
6    "rid": "room-id",
7    "u": { "_id": "user-id", "username": "username" },
8    "open": true,
9    "alert": false,
10   "unread": 0,
11   "_updatedAt": { "$date": 1480377601 },
12   "_id": "subscription-id"
13 }
```

Get User Roles

This method call is used to get server-wide special users and their roles. You may send an empty list of parameters.

The `result` will be a collection of `user`. The `user` is defined as:

- `username` : The username of the user
- `roles` : The collection of roles the user belongs to (at least one `role-name` will be present)
- `_id` : The user id

That information is used to identify key users on the server (ex.: admins).

Example call:

```
1      {
2          "msg": "method",
3          "method": "getUserRoles",
4          "id": "42",
5          "params": []
6      }
```

Response:

```
1      {
2          "msg": "result",
3          "id": "42",
4          "result": [
5              { "username": "username", "roles": [ "role-name" ], "_id": "us
6                  ... // more users
7              ]
8      }
```

Hiding Rooms

When you hide a room, that room no longer shows up on the list of channels and marks the property `open` to `false` on the user's subscription of the room.

Requirements

Logged In	Permission	Setting
Yes	<i>none</i>	<i>none</i>

Payload

1. String - the id of the room to leave
-

Example Call

```
1  {
2      "msg": "method",
3      "method": "hideRoom",
4      "id": "14",
5      "params": [
6          "roomId"
7      ]
8  }
```

Example Response

```
1  {
2    "msg": "result",
3    "id": "14",
4    "result": 1
5 }
```

See Also

- [Joining Channels](#)
- [Leaving Rooms](#)
- [Create Channels](#)

Joining Channels

You can only join yourself to public channels, private groups are not joinable. Some public channels require you to enter a `joinCode`.

Requirements

Logged In	Permission	Setting
Yes	<code>view-c-room</code>	<code>none</code>

Payload

1. String - the id of the room to join
 2. String - the `joinCode`, only needed if the room requires a code to join
-

Example Call

```
1  {
2      "msg": "method",
3      "method": "joinRoom",
4      "id": "99",
5      "params": [
6          "roomId",
7          "joinCode"
8      ]
9  }
```

Example Response

The response from calling the method will return a result of whether the user joined the channel or not.

```
1  {
2      "msg": "result",
3      "id": "99",
4      "result": true/false
5 }
```

See Also

- [Create Channels](#)
- [Delete Rooms](#)
- [Archive Rooms](#)

Leaving Rooms

You can leave any rooms, except for direct messages and except for rooms you are the last owner of.

Requirements

Logged In	Permission	Setting
Yes	<i>none</i>	<i>none</i>

Payload

1. String - the id of the room to leave
-

Example Call

```
1  {
2      "msg": "method",
3      "method": "leaveRoom",
4      "id": "11",
5      "params": [
6          "roomId"
7      ]
8  }
```

Example Response

```
1  {
2    "msg": "result",
3    "id": "11"
4 }
```

See Also

- [Joining Channels](#)
- [Create Channels](#)

List Custom Emoji

Returns a list of custom emoji registered with the server. There's no need for parameters.

The `result` will be a collection of `emoji`. The `emoji` is defined as:

- `_id` : The emoji id
- `name` : The emoji friendly name
- `aliases` : A collection of alias for the emoji. The alias is used to identify the emoji on text and for fast reference from typing - the famous `:emoji-alias:` . (Each emoji alias is unique per server)
- `extension` : The emoji file extension
- `_updatedAt` : The date when the emoji was updated to the server

Example request:

```
1  {
2      "msg": "method",
3      "method": "listEmojiCustom",
4      "id": "42",
5      "params": []
6 }
```

Response:

```
1  {
2      "msg": "result",
3      "id": "42",
4      "result": [
5          {
6              "_id": "emoji-id",
7              "name": "emoji-name",
8              "aliases": [ "emoji-alias" ],
9              "extension": "emoji-file-extension",
10             "_updatedAt": { "$date": 1480377601 }
11         },
12         ...
13     ]
```

```
14 }
```

Showing the emoji image

To show the custom emoji images, you simply need to request this URL:

```
| ${path}/emoji-custom/${encoded(name)}.${extension}.
```

Example:

```
1  {
2    "_id": "emoji-id",
3    "name": "Emoji Name",
4    "aliases": [ "emoji-alias" ],
5    "extension": "png",
6    "_updatedAt": { "$date": 1480377601 }
7 }
```

The URL to access this emoji will be:

```
http://yourhost.com/emoji-custom/Emoji%20Name.png
```

Load History

Use this method to make the initial load of a room. After the initial load you may subscribe to the room messages stream (see [Stream Room Message](#)).

This method accepts 4 parameters in the following order:

- The room id
- The NEWEST message timestamp date (or null) to only retrieve messages before this time. - this is used to do pagination
- The message quantity
- A date object - the date of the last time the client got data for the room

The result is composed of the `messages` collection and the `unreadNotLoaded` counter.

The `message` object is fairly complex and have [its own section describing it](#). The `unreadNotLoaded` counts the quantity of unread messages not loaded by the call.

Examples:

Request of the latest 50 messages

```
1  {
2    "msg": "method",
3    "method": "loadHistory",
4    "id": "42",
5    "params": [ "room-id", null, 50, { "$date": 1480377601 } ]
6 }
```

Request of the latest 50 messages, using pagination

```
1  {
2    "msg": "method",
3    "method": "loadHistory",
4    "id": "42",
5    "params": [ "room-id", { "$date": 1480377205 }, 50, { "$date": 1480377
```

```
6  }
```

Response

```
1  {
2    "msg": "result",
3    "id": "42",
4    "result": {
5      "messages": [
6        ... // messages
7      ],
8      "unreadNotLoaded": 0
9    }
10 }
```

Login

Used for user login. It supports a plethora of authentication services, including a simple username and password combination. The list of supported auth services (and its configurations) lies in the `meteor_accounts_loginServiceConfiguration` collection.

The overall format of the login message is:

```
1  {
2    "msg": "method",
3    "method": "login",
4    "id": "42",
5    "params": [ ... ] // changes according to the auth used
6 }
```

Independently of the auth provider, upon a successful login we'll receive back an `result` with the `user-id`, `auth-token` and the token expiration date. As the example bellow shows:

```
1  {
2    "msg": "result",
3    "id": "42",
4    "result": {
5      "id": "user-id",
6      "token": "auth-token",
7      "tokenExpires": { "$date": 1480377601 }
8    }
9 }
```

That information should be saved locally in order to automatically authenticate the user the next time he tries to use the client. Look at [\[Using an authentication token\]](#) for information about how to do it.

Username and Password

The user has an account directly with the RC server. It's important to say that we must not pass the user password as plain-text, applying a hashing algorithm makes things better (sha-256). Make sure your digest is lower-case!

Request

```
1  {
2      "msg": "method",
3      "method": "login",
4      "id": "42",
5      "params": [
6          {
7              "user": { "username": "example-user" },
8              "password": {
9                  "digest": "some-digest",
10                 "algorithm": "sha-256"
11             }
12         }
13     ]
14 }
```

In the case there's an error on the request, a possible error response would be:

```
1  {
2      "msg": "result",
3      "id": "42",
4      "error": {
5          "error": 403,
6          "reason": "Incorrect password",
7          "message": "Incorrect password [403]",
8          "errorType": "Meteor.Error"
9      }
10 }
```

Using Authentication providers

We're using OAuth to support additional auth providers.

Here's a example request.

```
1  {
2      "msg": "method",
3      "method": "login",
4      "id": "42",
5      "params": [
6          {
7              "oauth": {
8                  "credentialToken": "credential-token",
9                  "credentialSecret": "credential-secret"
10             }
11         }
12     ]
13 }
```

Using an authentication token

If you have a saved user authentication you may use the provided `auth-token` to automatically log the user in.

```
1  {
2      "msg": "method",
3      "method": "login",
4      "id": "42",
5      "params": [
6          { "resume": "auth-token" }
7      ]
8 }
```

A successful call will return the same message as a successful login (which it is).

About token expiration date

As the token expires, you have to call the login method again in order to obtain a new token with a new expiration date.

NB: You don't have to wait until the token is expired before asking for a new token.

See Also

- [Logout](#)

Logout

See Also

- [Login](#)

Notify Room Stream

Use this call to notify a room about some specific events.

The method used is `stream-notify-room` and the general format is:

```
1  {
2      "msg": "method",
3      "method": "stream-notify-room",
4      "id": "42",
5      "params": [
6          ... // params varies
7      ]
8 }
```

Typing

One of the events you may notify a room about is the user typing on the keyboard. This call needs three params.

- `room-id/typing` : The first param is the room id followed with `/typing`
- `username` : The second param is the username
- `flag` : The third param is a boolean indicating if the user is typing (`true`) or it's not typing (`false`)

Example call (the user is typing):

```
1  {
2      "msg": "method",
3      "method": "stream-notify-room",
4      "id": "42",
5      "params": [
6          "room-id/typing",
7          "username",
8          true
```

```
9      ]  
10 }
```

Opening Rooms

When you open a room, that room shows up on the list of channels and marks the property `open` to `true` on the user's subscription of the room.

Requirements

Logged In	Permission	Setting
Yes	<i>none</i>	<i>none</i>

Payload

1. String - the id of the room to open
-

Example Call

```
1  {
2      "msg": "method",
3      "method": "openRoom",
4      "id": "19",
5      "params": [
6          "roomId"
7      ]
8  }
```

Example Response

```
1  {
2    "msg": "result",
3    "id": "19",
4    "result": 1
5 }
```

See Also

- [Hiding Rooms](#)
- [Joining Channels](#)
- [Leaving Rooms](#)
- [Create Channels](#)

Pin Message

Pinning a message allows administrators and owners of rooms to easily store messages which are important.

Requirements

Logged In	Permission	Setting
Yes	<i>none</i>	Message_AllowPinning - "Allow Message Pinning"

Example Call

The only parameter that needs to be passed in is the [Message Object](#) and as of release 0.46 you need to send the entire message object otherwise an internal error will happen ([fixed via pull request #5087](#)).

```
1  {
2    "msg": "method",
3    "method": "pinMessage",
4    "id": "19",
5    "params": [ fullMessageObject ]
6 }
```

Example Response

The response of a message being pinned is a new chat message which contains the broadcast of the message pinning. See [Message Object Details](#) for information about the response format.

```
1  {
2      "msg": "result",
3      "id": "19",
4      "result": {
5          "t": "message_pinned",
6          "rid": "QFtTnPj4XbG634Skm",
7          "ts": { "$date": 1480613343046 },
8          "msg": "",
9          "u": {
10              "_id": "gHwBwDomPrCoQj7i2",
11              "username": "bradley"
12          },
13          "groupable": false,
14          "attachments": [
15              {
16                  "text": "test",
17                  "author_name": "bradley",
18                  "author_icon": "/avatar/bradley.jpg?_dc=0",
19                  "ts": { "$date": 1480613302330 }
20              ],
21              {
22                  "_updatedAt": { "$date": 1480613343046 },
23                  "_id": "sBYLyaHFkMdr7LKgt"
24              }
25      }
```

See Also

- [The Message Object](#)
- [Pinning Messages User Guide](#)

Register User

Used for user registration.

There is the possibility of `customFields` as extra fields for user registration. The information of extra fields is found in setting `Accounts_CustomFields` in JSON format.

The call format is:

```
1  {
2      "msg": "method",
3      "method": "registerUser",
4      "id": "42",
5      "params": [
6          "email": "String",
7          "pass": "String",
8          "name": "String",
9          "secretURL": "String" // Optional
10     ]
11 }
```

The response is just the created user's _id:

```
1  {
2      "msg": "result",
3      "id": "42",
4      "result": "random_id"
5 }
```

Errors

In the case there's an error on the request, a possible error response would be in this format:

```
1  {
2      "msg": "result",
3      "id": "42",
4      "error": {
5          "error": 403,
6          "reason": "...",
7          "message": "... [403]",
8          "errorType": "Meteor.Error"
9      }
10 }
```

Possible errors

```
1 "reason": "error-user-registration-disabled",
2 "message": "User registration is disabled [403]",
3
4 "reason": "error-user-registration-secret",
5 "message": "User registration is only allowed via Secret URL [403]",
6
7 "reason": "error-invalid-customfield-json",
8 "message": "Invalid JSON for Custom Fields [403]",
```

Custom Fields related

```
1 "reason": "error-user-registration-custom-field",
2 "message": "Field ${ fieldName } is required [403]",
3
4 "reason": "error-user-registration-custom-field",
5 "message": "Value for field ${ fieldName } is invalid [403]",
6
7 "reason": "error-user-registration-custom-field",
8 "message": "Max length of field ${ fieldName } ${ field.maxLength } [403]"
9
10 "reason": "error-user-registration-custom-field",
11 "message": "Min length of field ${ fieldName } ${ field.minLength } [403]"
```

Accounts_CustomFields

Example of what we can find inside the `Accounts_CustomFields` **encoded as JSON**

```
1  {
2      "role": {
3          "type": "select",
4          "defaultValue": "student",
5          "options": ["teacher", "student"],
6          "required": true,
7          "modifyRecordField": {
8              "array": true,
9              "field": "roles"
10         }
11     },
12     "twitter": {
13         "type": "text",
14         "required": true,
15         "minLength": 2,
16         "maxLength": 10
17     }
18 }
```

Save Room Settings

Requirements

Logged In	Permission	Setting
Yes	edit-room	<i>none</i>

Payload

1. String - the id of the room to leave
2. String - the room setting to save
3. Any - the value of the setting to save, this value type depends on what the setting accepts

Available Settings

Setting	Accepted Values
roomName	String
roomTopic	String
roomDescription	String
roomType	c or p
readOnly	Boolean
systemMessages	Boolean
default	Boolean

```
joinCode
```

```
String
```

Example Call

```
1  {
2      "msg": "method",
3      "method": "saveRoomSettings",
4      "id": "16",
5      "params": [
6          "roomId",
7          "setting",
8          "value"
9      ]
10 }
```

Example Response

```
1  {
2      "msg": "result",
3      "id": "16",
4      "result": [
5          "result": true,
6          "rid": "roomId"
7      ]
8 }
```

See Also

- [Create Channels](#)
- [Create Private Groups](#)

- Delete Rooms
- Archive Rooms
- Unarchive Rooms

Send Message

There are a few ways to communicate when using Rocket.Chat, as described below.

Sending a text message

The most basic (and more common) way to communicate is through plain text. You're going to use the `sendMessage` method and send a single parameter: the `message`.

The `sendMessage` is described as:

- `_id` : The message id
- `rid` : The room id for where to send this message
- `msg` : The message body (the text of the message itself)

```
1  {
2      "msg": "method",
3      "method": "sendMessage",
4      "id": "42",
5      "params": [
6          {
7              "_id": "message-id",
8              "rid": "room-id",
9              "msg": "Hello World!"
10         }
11     ]
12 }
```

Sending a file

For sending a file we now recommend using the rest api:

[See rooms.upload](#)

See Also

- [The Message Object](#)
- [Delete Message Method](#)
- [Update Message Method](#)

Set Reaction

Reacting to messages is a convenient way to "react" to messages with emojis, allowing for a wide variety of options for integrations, questions, and whatever else people can come up with.

Requirements

Logged In	Permission	Setting
Yes	<i>none</i>	<i>none</i>

Example Call

Setting and removing a reaction requires a call to the same method with the same parameters. The first parameter is the reaction which is also the `emoji`, the second parameter is the message id of which the reaction is happening against and the third parameter is optional and work like a setter.

```
1  {
2      "msg": "method",
3      "method": "setReaction",
4      "id": "22",
5      "params": [
6          ":nerd:",
7          "messageId"
8          true/false
9      ]
10 }
```

Example Response

```
1  {
2      "msg": "result",
3      "id": "22"
4 }
```

See Also

- [Reacting to Messages User Guide](#)

Spotlight (search)

Search for rooms or users.

Requirements

Logged In	Permission	Setting
Yes	<i>none</i>	<i>none</i>

Payload

1. `searchString` - String - the name of the thing being searched for
 2. `exclude` - Array of strings - any items to exclude from the results (e.g. users you already know about)
 3. `options` - Object - what kinds of items to search for, containing `users` and `rooms` - both booleans.
-

Example Call

```
1  {
2      "msg": "method",
3      "method": "spotlight",
4      "params": ["test", ["testuser1"], {
5          "users": true,
6          "rooms": false
7      }],
8      "id": "unique-id"
9  }
```

Example Response

```
1  {
2      "msg": "result",
3      "id": "unique-id",
4      "result": {
5          "users": [
6              {
7                  "_id": "q779pX2DPQdHpJuid",
8                  "status": "online",
9                  "name": "Admin",
10                 "username": "admin"
11             },
12             {
13                 "_id": "T5xeLJEC2FSagZp5X",
14                 "username": "alice",
15                 "status": "offline",
16                 "name": "alice"
17             },
18             {
19                 "_id": "rocket.cat",
20                 "name": "Rocket.Cat",
21                 "username": "rocket.cat",
22                 "status": "online"
23             }
24         ],
25         "rooms": [
26             {
27                 "_id": "YGKfY5j7hAGKjhWP",
28                 "name": "chatter",
29                 "t": "c",
30                 "lastMessage": {
31                     "_id": "0innp70tD4z2AdmNk",
32                     "rid": "YGKfY5j7hAGKjhWP",
33                     "msg": "Hello",
34                     "ts": {
35                         "$date": 1588252316473
36                     },
37                     "u": {
38                         "_id": "T5xeLJEC2FSagZp5X",
39                         "username": "alice"
40                     },
41                     "mentions": [],
42                     "channels": [],
43                     "_updatedAt": {
44                         "$date": 1588252316498
45                     }
46                 }
47             }
48         ]
49     }
50 }
```

```
47     },
48     {
49         "_id": "GENERAL",
50         "t": "c",
51         "name": "general",
52         "lastMessage": {
53             "_id": "KqcDbY6Ab3wmEY5N9",
54             "rid": "GENERAL",
55             "msg": "test1",
56             "ts": {
57                 "$date": 1589383273561
58             },
59             "u": {
60                 "_id": "q779pX2DPQdHpJuid",
61                 "username": "admin"
62             },
63             "mentions": [],
64             "channels": [],
65             "_updatedAt": {
66                 "$date": 1589383273645
67             }
68         }
69     },
70     {
71         "_id": "xbE8idJpsTuNzi8Ax",
72         "name": "testing",
73         "t": "c"
74     }
75 ]
76 }
77 }
```

Star Message

Starring messages allows a user to quickly save for future reference, or something similar, for their own personal usage.

Requirements

Logged In	Permission	Setting
Yes	<code>none</code>	<code>Message_AllowStarring</code> - "Allow Message Starring"

Example Call

The only parameter that needs to be passed in is an object with the `_id`, `rid`, and `starred` properties. The `starred` property is a boolean and it determines whether the user is starring the message or removing the star from the message.

```
1  {
2      "msg": "method",
3      "method": "starMessage",
4      "id": "21",
5      "params": [
6          {"_id": "sBYLyaHFkMdr7LKGt",
7              "rid": "QFtTnPJ4XbG634Skm",
8              "starred": true || false
9      ]
10 }
```

Example Response

```
1  {
2    "msg": "result",
3    "id": "21",
4    "result": 1
5 }
```

See Also:

- [Starring Messages User Guide](#)

Unarchive Rooms

Unarchiving a room removes it from being read only and then adds it back to the channel list on the left.

Requirements

Logged In	Permission	Setting
Yes	unarchive-room	<i>none</i>

Payload

1. String - the id of the room to unarchive
-

Example Call

```
1  {
2      "msg": "method",
3      "method": "unarchiveRoom",
4      "id": "99",
5      "params": [
6          "roomId"
7      ]
8  }
```

Example Response

```
1  {
2    "msg": "result",
3    "id": "99"
4 }
```

See Also

- [Create Channels](#)
- [Create Private Groups](#)
- [Delete Rooms](#)
- [Archive Rooms](#)
- [Save Room Settings](#)

Unpin Message

Unpinning a message allows administrators and owners of rooms to remove pinned items from being pinned after they are no longer valid.

Requirements

Logged In	Permission	Setting
Yes	<i>none</i>	Message_AllowPinning - "Allow Message Pinning"

Example Call

The only parameter that needs to be passed in is the [Message Object](#) and as of release 0.46 you need to send the entire message object otherwise an internal error will happen ([fixed via pull request #5087](#)).

```
1  {
2    "msg": "method",
3    "method": "unpinMessage",
4    "id": "20",
5    "params": [ fullMessageObject ]
6 }
```

Example Response

The response of a message being pinned is a new chat message which contains the broadcast of the message pinning. See [Message Object Details](#) for information about the response format.

```
1  {
2      "msg": "result",
3      "id": "20",
4      "result": {
5          "t": "message_pinned",
6          "rid": "QFtTnPj4XbG634Skm",
7          "ts": { "$date": 1480613343046 },
8          "msg": "",
9          "u": {
10              "_id": "gHwBwDomPrCoQj7i2",
11              "username": "bradley"
12          },
13          "groupable": false,
14          "attachments": [
15              {
16                  "text": "test",
17                  "author_name": "bradley",
18                  "author_icon": "/avatar/bradley.jpg?_dc=0",
19                  "ts": { "$date": 1480613302330 }
20              ],
21              {
22                  "_updatedAt": { "$date": 1480613343046 },
23                  "_id": "sBYLyaHFkMdr7LKgt"
24              }
25      }
```

See Also

- [Pinning Messages User Guide](#)

Update Message

Updating a message is what should be called when a user "edits" a message, passing in the changed message with the updates.

Requirements

Logged In	Permission	Setting
Yes	edit-message	Message_AllowEditing - "Allow Message Editing"

Example Call

The only parameter that needs to be passed in is the **Message Object** which contains the updated message properties, such as the text.

```
1  {
2    "msg": "method",
3    "method": "updateMessage",
4    "id": "42",
5    "params": [ messageObject ]
6 }
```

Example Response

```
1  {
2    "msg": "result",
3    "id": "42",
4    "result": []
```

```
5 }
```

Additional Information

As mentioned in requirements, you must be logged in to be able to edit a message along with having the permission `edit-message`. Two settings apply, `Message_AllowEditing` and then `Message_AllowEditing_BlockEditInMinutes`. The first setting is a boolean, true/false, and the second setting is an integer that can be `0` for unlimited editing or anything greater and editing will be blocked after the time in minutes has passed.

See Also

- [The Message Object](#)
- [Send Message Method](#)
- [Delete Message Method](#)

User Presence

Used to set the user presence status. There are two ways to use this call; to set the default status and to set a temporary status.

There are four different statuses available:

- `online`
 - `busy`
 - `away`
 - `offline`
-

Setting the default status

In order to set the default status we need to call the `UserPresence:setDefaultStatus` method passing the status in the parameters of the call.

Example call (setting the user presence to `online`):

```
1  {
2      "msg": "method",
3      "method": "UserPresence:setDefaultStatus",
4      "id": "42",
5      "params": [ "online" ]
6 }
```

Setting a temporary status

Setting a temporary status requires a call to `UserPresence:{status}` with empty params. Only `away` and `online` are accepted. This method call is useful when the client identifies that the user is not using the application (and therefore away) and when he got back.

Example call (setting the user presence to `away`):

```
1  [
2    "msg": "method",
3    "method": "UserPresence:away",
4    "id": "42",
5    "params": []
6  }
```

Two Factor Authentication

Visit [the Two Factor Authorization page](#) for general information about Two Factor Authorization.

Errors

When a call that requires two factor is made it will return an error `totp-require`. The details object will list the method that has been required (email on this example) so it's possible to inform the user to check his email for the code.

- **method**: The method selected by the server. Useful to inform the user where to look for the code.
- **codeGenerated**: Email only. Used to inform if the code was generated or if there are tokens available already.
- **codeCount**: (optional) Email only. The number of available codes already sent via email.
- **codeExpires**: (optional) Email only. A list of expiration dates of the tokens.
- **availableMethods**: The list of available methods for Two Factor. When calling an api it's possible to define the method to use.

```
1  {
2    "msg": "result",
3    "id": "1",
4    "error": {
5      "isClientSafe": true,
6      "error": "totp-required",
7      "reason": "TOTP Required",
8      "details": {
9        "method": "email",
10       "codeGenerated": false,
11       "codeCount": 1,
12       "codeExpires": [
13         "2019-12-31T22:05:22.159Z"
14       ],
15       "availableMethods": [
16         "email"
```

```
17      ]
18    },
19    "message": "TOTP Required [totp-required]",
20    "errorType": "Meteor.Error"
21  }
22 }
```

Calling a method with Two Factor

After receive the error it's necessary to pass the informed code to the API. For that we need to call a new method called `callWithTwoFactorRequired` passing the information as an object:

Request

- **code**: (string) The code informed by the user;
- **ddpMethod**: (string) The original method called;
- **method**: (string) The desired method to check the Two Factor, usually the same from the error;
- **params**: (any[]) An array of parameters used for the original method;

Result

- If the two factor was accepted the **result** and the **error** will came from the original method;
- If the two factor was not accepted the **error** `totp-invalid` will be returned;

Example

```
Meteor.call('callWithTwoFactorRequired', { code, ddpMethod, method, params:
```

Requesting a new email code

If the user didn't receive the code it's possible to request the server to send a new code via email by calling the DDP Method `sendEmailCode` passing the user's email or username. It's required to pass the email or username because this Method can be called when the user is not logged in.

Request

- `sendEmailCode`: (string) The user's username or email

Result

- If success: **array of emails** to where the code was sent;
- If error: **error-parameter-required** if the parameter `emailOrUsername` was not provided;
- If error: **error-invalid-user** if the user was not found with the provided `emailOrUsername` ;

Example

```
Meteor.call('sendEmailCode', emailOrUsername, (error, result) => {});
```

Enabling the Two Factor via Email

It's possible to enable the email check by calling the Method `2fa:enable-email` . Note that the two factor via email will only work if the user has at least one verified email.

Result

- If success: **true** is returned;

- If error: **not-authorized** if the user is not logged in;

Example

```
Meteor.call('2fa:enable-email', (error, result) => {});
```

Disabling the Two Factor via Email

It's possible to disabled the email check by calling the Method `2fa:disable-email`. Note this Method requires the two factor to be executed.

Result

- If success: **true** is returned;
- If error: A two factor verification error is returned;

Example

```
Meteor.call('2fa:disable-email', (error, result) => {});
```

Livechat Realtime API

This API is intended to be used for having a Livechat conversation;

- First of all you need to generate a visitor token (any random string);
- Call `livechat:getInitialData` passing `visitorToken` as first argument, the response will be an object containing a Livechat [configuration object](#) with following properties:

Field	Type	Description
<code>enabled</code>	<code>Boolean</code>	If whether Livechat is enabled for that server or not
<code>title</code>	<code>String</code>	The Livechat widget title
<code>color</code>	<code>Hexadecimal</code>	The hexadecimal color of the Livechat widget title bar when the Livechat is online
<code>registrationForm</code>	<code>Boolean</code>	If the registration form should be displayed or not.
<code>room</code>	<code>Object</code>	The current conversation room for the current guest user
<code>visitor</code>	<code>Object</code>	The current guest user
<code>triggers</code>	<code>Array</code>	Array of Livechat triggers.
<code>departments</code>	<code>Array</code>	Array of Livechat departments.
<code>allowSwitchingDepartments</code>	<code>Boolean</code>	If client-side department switching is allowed
<code>online</code>	<code>Boolean</code>	If there are Livechat agents online
<code>offlineColor</code>	<code>Hexadecimal</code>	The hexadecimal color of the Livechat widget title bar when the Livechat is offline

offlineMessage	String	The message that will be displayed on the Livechat offline form
offlineSuccessMessage	String	The message that will be displayed after send a message using the offline form
offlineUnavailableMessage	String	The message that will be displayed when the Livechat is offline
displayOfflineForm	Boolean	If the offline form will be displayed when the Livechat is offline
videoCall	Boolean	If the VideoCall feature is available
conversationFinishedMessage	Boolean	The system message that will be send to the current guest user when the conversation is closed
nameFieldRegistrationForm	Boolean	If the name field will be displayed on Livechat registration form
emailFieldRegistrationForm	Boolean	If the email field will be displayed on Livechat registration form
offlineTitle	String	The title of the widget when the Livechat is offline
language	String	The default user language to be set in the Livechat widget
transcript	Boolean	If the Livechat widget will ask the current guest user if they would like a transcript after the conversation is closed
transcriptMessage	String	The message to be displayed when asking about transcript
agentData	Object	The current agent attending the chart

- Call `livechat:registerGuest` to register guest and get the response containing the visitor's data.
- Before sending the first message you have to generate a random `room _id` ;
- Now you can send messages to method `sendMessageLivechat`
- Subscribe to: `stream-room-messages` and `stream-livechat-room`
- Get agent info by calling `livechat:getAgentData`

livechat:getInitialData

DDP message

```
{"msg":"method","method":"livechat:getInitialData","params":["7T4jzes7rX3Fr6
```

Response

```
1  {
2    enabled: true,
3    title: 'Rocket.Chat',
4    color: '#C1272D',
5    registrationForm: true,
6    room: null,
7    visitor: undefined,
8    triggers: [],
9    departments: [],
10   allowSwitchingDepartments: true,
11   online: true,
12   offlineColor: '#666666',
13   offlineMessage: 'We are not online right now. Please leave us a message',
14   offlineSuccessMessage: '',
15   offlineUnavailableMessage: '',
16   displayOfflineForm: true,
17   videoCall: false,
18   offlineTitle: 'Leave a message',
19   language: '',
20   transcript: false,
21   transcriptMessage: 'Would you like a copy of this chat emailed?',
22   agentData: undefined,
23   conversationFinishedMessage: 'Conversation finished',
24   nameFieldRegistrationForm: true,
25   emailFieldRegistrationForm: true
26 }
```

livechat:registerGuest

DDP message

```
{"msg":"method","method":"livechat:registerGuest","params": [{"token":"TF5rZ7BZ9mZCSq3xN"}]}
```

Response

```
1  {
2    userId: 'G3DukvFBhDkDnw6uS',
3    visitor: {
4      name: 'Guest Name',
5      token: 'TF5rZ7BZ9mZCSq3xN',
6      username: 'guest-1',
7      visitorEmails: [
8        {
9          address: 'guest@rocket.chat'
10        }
11      ]
12    }
13 }
```

livechat:sendOfflineMessage

DDP message

```
{"msg":"method","method":"livechat:sendOfflineMessage","params": [{"name":"Vi
```

sendMessageLivechat

DDP message

```
{"msg":"method","method":"sendMessageLivechat","params": [{"_id":"XqEEHhQHvhFmK3Zoz","rid":"TT9iMmzusfcLq8sv2","msg":"test","token":"7T4jzes7rX3Fr6cQ2","alias":"poqiqwp1o2","ts": {"$date":1494874057495}, "u": {"_id":"4Zo8JcXynkMrXbFMN","username":"guest-82"}, "_updatedAt": {"$date":1494874057498}, "newRoom":true, "showConnecting":false} ]}
```

The property `_id` should be generated randomly before sending the message.

Response

```
1  {
2    "_id": "XqEEHhQHvhFmK3Zoz",
3    "rid": "TT9iMmzusfcLq8sv2",
4    "msg": "test",
5    "token": "7T4jzes7rX3Fr6cQ2",
6    "alias": "poqiqwp1o2",
7    "ts": {
8      "$date": 1494874057495
9    },
10   "u": {
11     "_id": "4Zo8JcXynkMrXbFMN",
12     "username": "guest-82"
13   },
14   "_updatedAt": {
15     "$date": 1494874057498
16   },
17   "newRoom": true,
18   "showConnecting": false
19 }
```

stream-livechat-room

DDP message

```
1  {
2      "msg":"sub",
3      "id":"6NctZomXL3ZdtKNsn",
4      "name":"stream-livechat-room",
5      "params":[
6          "KTZqPAR9DQGxKcxzf",
7          {
8              "useCollection":false,
9              "args":[
10                 {
11                     "token":"jkGaw6duhiuh45"
12                 }
13             ]
14         }
15     ]
16 };
```

Response

```
1  {
2      "msg":"ready",
3      "subs":[
4          "JHalkajwdh67"
5      ]
6  }
```

Subscriptions

Streams are the way to plug into a continuous source of updates (changes). Any subscriber registered will receive the latest changes as they occur.

In order to subscribe to a stream you must send a message with `msg: sub`, an unique `id`, the stream `name` and the `params` to be applied on the stream.

```
1  {
2    "msg": "sub",
3    "id": "unique-id",
4    "name": "the-stream",
5    "params": [ "event", false ]
6 }
```

The new stream API will propagate only changes to subscribers, which may break some drivers. In order to keep it back-compatible the last parameter in the parameters must be a boolean: whatever or not you required back-compatibility. If set to `true` you should receive an `add` event every time something new is created.

NOTE: before you can subscribe to any stream or use any Subscription related features, you have to send a [login request](#) using the RealTime API.

Unsubscribing from a stream can be done in the following way:

```
1  {
2    "msg": "unsub",
3    "id": "subscription-id"
4 }
```

Where the `subscription_id` is the unique `id` that was used when you initially made the call to subscription. The response will be following:

```
1  {
2    "msg": "nosub",
3    "id": "subscription-id"
4 }
```

Be aware that even a call to an invalid or nonexistent `subscription_id` will produce a non-error response containing at least `"msg": "nosub"`, and whatever `id` you provided. The `id` field is omitted if no `id` was provided.

stream-notify-all

General user-wide stream.

Events available:

- roles-change
- updateEmojiCustom
- deleteEmojiCustom
- updateAvatar
- public-settings-changed
- permissions-changed

```
1  {
2      "msg": "sub",
3      "id": "unique-id",
4      "name": "stream-notify-all",
5      "params": [
6          "event",
7          false
8      ]
9  }
```

stream-notify-logged

Stream for logged users

Events available:

- Users:NameChanged
- Users:Deleted
- updateAvatar
- updateEmojiCustom
- deleteEmojiCustom
- roles-change
- **user-status**

```
1  {
2      "msg": "sub",
3      "id": "unique-id",
4      "name": "stream-notify-logged",
5      "params": [
6          "event",
7          false
8      ]
9 }
```

user-status

Event about user status changes. Example:

```
1  {
2      "msg": "changed",
3      "collection": "stream-notify-logged",
4      "id": "id",
5      "fields": {
6          "eventName": "user-status",
7          "args": [["uNqJeFuag2344i62k", "rocket.cat", 1]]
8      }
}
```

```
9 }
```

The `args` property will always be an array with the following values:

- User `_id`
- `username`
- Status indicator, where:

	Value	Meaning
0	Offline	
1	Online	
2	Away	
3	Busy	

stream-notify-room-users

This is the stream for room users. Currently used to notify about WebRTC Calls.

Events available:

- webrtc

```
1  {
2    "msg": "sub",
3    "id": "unique-id",
4    "name": "stream-notify-room-users",
5    "params": [
6      "user-id/event",
7      false
8    ]
9 }
```

stream-notify-room

This is a room stream.

Replace event from one in the list Events available:

- deleteMessage
- typing

```
1  {
2    "msg": "sub",
3    "id": "unique-id",
4    "name": "stream-notify-room",
5    "params": [
6      "room-id/event",
7      false
8    ]
9 }
```

stream-notify-user

This is the user stream.

Events available:

- message
- otr (Off the Record Message)
- webrtc
- notification
- rooms-changed
- subscriptions-changed

```
1  {
2      "msg": "sub",
3      "id": "unique-id",
4      "name": "stream-notify-user",
5      "params": [
6          "user-id/event",
7          false
8      ]
9  }
```

stream-room-messages

This is the room messages stream. You just need the room ID for subscribing.

```
1  {
2    "msg": "sub",
3    "id": "unique-id",
4    "name": "stream-room-messages",
5    "params": [
6      "room-id",
7      false
8    ]
9 }
```

To unsubscribe from a room, send "unsub", along with the ID of the previous subscribe message.

```
1  {
2    "msg": "unsub",
3    "id": "unique-id",
4 }
```

If you want to subscribe to a user's messages stream, just pass `--my_messages--` as the room ID.

REST API

The REST API allows you to control and extend Rocket.Chat with ease.

This API is a work in progress, so feel free to test, ask us questions, and submit Pull Requests!

If you are an end-user and not a dev or a tester, [create a New Feature Request](#) to request new APIs – and consider [making a donation](#) to the project.

All API calls in the documentation are made using `curl`. However, you are free to use Java / Python / PHP / Golang / Ruby / Swift / Objective-C / Rust / Scala / C# or any other programming languages.

Production Security Concerns

When calling a production Rocket.Chat server, ensure it is running via HTTPS and has a valid SSL Certificate. The login method requires you to post your username and password in plaintext, which is why we highly suggest only calling the REST login API over HTTPS. Also, few things to note:

- Only call via HTTPS
- Implement a timed authorization token expiration strategy
- Ensure the calling user only has permissions for what they are calling and no more

Miscellaneous Information

Url	Short Description	Details Page
/api/info	Information about the Rocket.Chat server.	Link
/api/v1/directory	Search by all users and channels available on server.	Link

/api/v1/shield.svg	Gets the shield svg(badge) to add in your website.	Link
/api/v1/spotlight	Searches for users or rooms that are visible to the user.	Link
/api/v1/statistics	Statistics about the Rocket.Chat server.	Link
/api/v1/statistics.list	Selectable statistics about the Rocket.Chat server.	Link
/api/v1/instances.get	Gets all running instances.	Link

Assets

Url	Short Description	Details Page
/api/v1/assets.setAsset	Set an asset image by name.	Link
/api/v1/assets.unsetAsset	Unset an asset by name	Link

AutoTranslate

Url	Short Description	Details Page
/api/v1/autotranslate.getSupportedLanguages	Get the supported languages by the autotranslate.	Link
/api/v1/autotranslate.saveSettings	Save some settings about autotranslate.	Link
/api/v1/autotranslate.translateMessage	Translate the message.	Link

Authentication

Url	Short Description	Details Page
/api/v1/login	Authenticate with the REST API.	Link
/api/v1/login	Authenticate with google.	Link
/api/v1/login	Authenticate with facebook.	Link
/api/v1/login	Authenticate with twitter.	Link
/api/v1/logout	Invalidate your REST API authentication token.	Link
/api/v1/me	Displays information about the authenticated user.	Link

Users

Url	Short Description	Details Page
/api/v1/users.presence	Gets all connected users presence.	Link
/api/v1/users.create	Create a new user.	Link
/api/v1/users.createToken	Create a user authentication token.	Link
/api/v1/users.deactivateIdle	Deactivate idle users.	Link
/api/v1/users.delete	Deletes an existing user.	Link
/api/v1/users.deleteOwnAccount	Deletes your own user.	Link
/api/v1/users.forgotPassword	Send email to reset your password.	Link
/api/v1/users.generatePersonalAccessToken	Generate Personal Access Token.	Link

/api/v1/users.getAvatar	Gets the URL for a user's avatar.	Link
/api/v1/users.getPersonalAccessTokens	Gets the user's personal access tokens.	Link
/api/v1/users.getPreferences	Gets all preferences of user.	Link
/api/v1/users.getPresence	Gets the online presence of a user.	Link
/api/v1/users.getStatus	Gets the user's status.	Link
/api/v1/users.getUsernameSuggestion	Gets a suggestion a new username to user.	Link
/api/v1/users.info	Gets a user's information, limited to the caller's permissions.	Link
/api/v1/users.list	All of the users and their information, limited to permissions.	Link
/api/v1/users.regeneratePersonalAccessToken	Regenerate a user personal access token.	Link
/api/v1/users.register	Register a new user.	Link
/api/v1/users.removeOtherTokens	Remove all other user tokens	Link
/api/v1/users.removePersonalAccessToken	Remove a personal access token.	Link
/api/v1/users.requestDataDownload	Request users download data.	Link
/api/v1/users.resetAvatar	Reset a user's avatar	Link

/api/v1/users.setAvatar	Set a user's avatar	Link
/api/v1/users.setPreferences	Set user's preferences	Link
/api/v1/users.setStatus	Set the user's status	Link
/api/v1/users.setActiveStatus	Set a user's active status.	Link
/api/v1/users.update	Update an existing user.	Link
/api/v1/users.updateOwnBasicInfo	Update basic information of own user.	Link

Channels

Url	Short Description	Details Page
/api/v1/channels.addAll	Adds all of the users on the server to a channel.	Link
/api/v1/channels.addLeader	Gives the role of Leader for a user in the current channel.	Link
/api/v1/channels.addOwner	Gives the role of owner for a user in the current channel.	Link
/api/v1/channels.anonymousread	Gets the messages in public channels to an anonymous user	Link
/api/v1/channels.archive	Archives a channel.	Link
/api/v1/channels.close	Removes a channel from a user's list of channels.	Link
/api/v1/channels.counters	Gets channel counters.	Link

/api/v1/channels.create	Creates a new channel.	Link
/api/v1/channels.delete	Removes a channel.	Link
/api/v1/channels.files	Gets a list of files from a channel.	Link
/api/v1/channels.getAllUserMentionsByChannel	Gets all the mentions of a channel.	Link
/api/v1/channels.getIntegrations	Gets the channel's integration.	Link
/api/v1/channels.history	Retrieves the messages from a channel.	Link
/api/v1/channels.info	Gets a channel's information.	Link
/api/v1/channels.invite	Adds a user to a channel.	Link
/api/v1/channels.join	Joins yourself to a channel.	Link
/api/v1/channels.kick	Removes a user from a channel.	Link
/api/v1/channels.leave	Removes the calling user from a channel.	Link
/api/v1/channels.list	Retrieves all of the channels from the server.	Link
/api/v1/channels.list.joined	Gets only the channels the calling user has joined.	Link
/api/v1/channels.members	Retrieves all channel users.	Link

/api/v1/channels.messages	Retrieves all channel messages.	Link
/api/v1/channels.moderators	List all moderators of a channel.	Link
/api/v1/channels.online	List all online users of a channel.	Link
/api/v1/channels.open	Adds the channel back to the user's list of channels.	Link
/api/v1/channels.removeleader	Removes the role of Leader for a user in the current channel.	Link
/api/v1/channels.rename	Changes a channel's name.	Link
/api/v1/channels.roles	Gets the user's roles in the channel.	Link
/api/v1/channels.setCustomFields	Sets a channel's custom fields.	Link
/api/v1/channels.setAnnouncement	Sets a channel's announcement.	Link
/api/v1/channels.setDefault	Sets whether a channel is a default channel or not.	Link
/api/v1/channels.setDescription	Sets a channel's description.	Link
/api/v1/channels.setJoinCode	Sets the channel's code required to join it.	Link
/api/v1/channels.setPurpose	Sets a channel's description.	Link

/api/v1/channels.setReadOnly	Sets whether a channel is read only or not.	Link
/api/v1/channels.setTopic	Sets a channel's topic.	Link
/api/v1/channels.setType	Sets the type of room the channel should be.	Link
/api/v1/channels.unarchive	Unarchives a channel.	Link
/api/v1/channels.addOwner	Gives the role of owner for a user in the current channel.	Link
/api/v1/channels.removeOwner	Removes the role of owner from a user in the current channel.	Link

Groups

Url	Short Description	Details Page
/api/v1/groups.archive	Archives a private group.	Link
/api/v1/groups.addLeader	Gives the role of Leader for a user in the current group.	Link
/api/v1/groups.close	Removes a private group from the list of groups.	Link
/api/v1/groups.counters	Gets group counters.	Link
/api/v1/groups.create	Creates a new private group.	Link
/api/v1/groups.delete	Removes a private group.	Link
/api/v1/groups.files	Gets a list of files from a private group.	Link

/api/v1/groups.history	Retrieves the messages from a private group.	Link
/api/v1/groups.info	Gets the information about a private group.	Link
/api/v1/groups.invite	Adds a user to the private group.	Link
/api/v1/groups.kick	Removes a user from a private group.	Link
/api/v1/groups.leave	Removes the calling user from the private group.	Link
/api/v1/groups.list	List the private groups the caller is part of.	Link
/api/v1/groups.listAll	List all the private groups.	Link
/api/v1/groups.members	Gets the users of participants of a private group.	Link
/api/v1/groups.messages	Retrieves all group messages.	Link
/api/v1/groups.moderators	List all moderators of a group.	Link
/api/v1/groups.online	List all online users of a group.	Link
/api/v1/groups.open	Adds the private group back to the list of groups.	Link
/api/v1/groups.removeLeader	Removes the role of Leader for a user in the current group.	Link
/api/v1/groups.rename	Changes the name of the private group.	Link
/api/v1/groups.roles	Gets the user's roles in the private group.	Link
/api/v1/groups.setAnnouncement	Sets a group's announcement.	Link
/api/v1/groups.setCustomFields	Sets private group's custom fields.	Link

/api/v1/groups.setDescription	Sets a private group's description.	Link
/api/v1/groups.setPurpose	Sets a private group's description.	Link
/api/v1/groups.setReadOnly	Sets whether the room is read only or not.	Link
/api/v1/groups.setTopic	Sets a private group's topic.	Link
/api/v1/groups.setType	Sets the type of room this group will be.	Link
/api/v1/groups.unarchive	Unarchives a private group.	Link
/api/v1/groups.addOwner	Gives the role of owner for a user in the current group.	Link
/api/v1/groups.removeOwner	Removes the role of owner from a user in the current Group.	Link

Chat

Url	Short Description	Details Page
/api/v1/chat.delete	Deletes an existing chat message.	Link
/api/v1/chat.followMessage	Follows an existing chat message.	Link
/api/v1/chat.getDeletedMessages	Retrieves the deleted messages since specific date.	Link
/api/v1/chat.getDiscussions	Retrieves the discussions messages of a room.	Link
/api/v1/chat.getMentionedMessages	Retrieves mentioned messages.	Link
/api/v1/chat.getMessage	Retrieves a single chat message.	Link
/api/v1/chat.getMessageReadReceipts	Retrieves message read receipts.	Link

/api/v1/chat.getPinnedMessages	Retrieve pinned messages from a room.	Link
/api/v1/chat.getSnippedMessages	Retrieves snipped messages.	Link
/api/v1/chat.getSnippedMessageById	Retrieves snipped message by id.	Link
/api/v1/chat.getStarredMessages	Retrieves starred messages.	Link
/api/v1/chat.getThreadMessages	Retrieves thread's messages.	Link
/api/v1/chat.getThreadsList	Retrieves channel's threads.	Link
/api/v1/chat.ignoreUser	Ignores an user from a chat.	Link
/api/v1/chat.pinMessage	Pins a chat message to the message's channel.	Link
/api/v1/chat.postMessage	Posts a new chat message.	Link
/api/v1/chat.react	Sets/unsets the user's reaction to an existing chat message.	Link
/api/v1/chat.reportMessage	Reports a message.	Link
/api/v1/chat.search	Search for messages in a channel.	Link
/api/v1/chat.starMessage	Stars a chat message for the authenticated user.	Link
/api/v1/chat.sendMessage	Send new chat message.	Link
/api/v1/chat.syncThreadMessages	Retrieves synced thread's messages.	Link
/api/v1/chat.syncThreadsList	Retrieves thread's synced channel threads.	Link
/api/v1/chat.unfollowMessage	Unfollows an existing chat message.	Link

/api/v1/chat.unPinMessage	Removes the pinned status of the provided chat message.	Link
/api/v1/chat.unStarMessage	Removes the star on the chat message for the authenticated user.	Link
/api/v1/chat.update	Updates the text of the chat message.	Link

Custom Sounds

Url	Short Description	Details Page
/api/v1/custom-sounds.list	Retrieves a list of custom sounds.	Link

IM

Url	Short Description	Details Page
/api/v1/im.close	Removes a direct message from the list of direct messages.	Link
/api/v1/im.counters	Gets counters of direct messages.	Link
/api/v1/im.create	Create a direct message session with another user.	Link
/api/v1/im.history	Retrieves the messages from a direct message.	Link
/api/v1/im.files	Retrieves a list of files from a direct message.	Link
/api/v1/im.members	Retrieves the users of participants of a direct message.	Link

/api/v1/im.messages	Retrieves the messages from specific direct message.	Link
/api/v1/im.messages.others	Retrieves the messages from any direct message in the server.	Link
/api/v1/im.list	List the direct messages the caller is part of.	Link
/api/v1/im.list.everyone	List all direct message the caller in the server.	Link
/api/v1/im.open	Adds the direct message back to the list of direct messages.	Link
/api/v1/im.setTopic	Sets a direct message topic.	Link

Integrations

Url	Short Description	Details Page
/api/v1/integrations.create	Creates an integration.	Link
/api/v1/integrations.get	Gets an integration.	Link
/api/v1/integrations.history	Lists all history of the specified integration.	Link
/api/v1/integrations.list	Lists all of the integrations.	Link
/api/v1/integrations.remove	Removes an integration.	Link

Invites

Url	Short Description	Details Page

/api/v1/findOrCreateInvite	Created a new Invite or returns an existing one with the same parameters.	Link
/api/v1/listInvites	Lists all of the invite tokens.	Link
/api/v1/removeInvite	Removes an invite.	Link
/api/v1/useInviteToken	Report to the server that an invite token was used.	Link
/api/v1/validateInviteToken	Checks if an invite token is valid.	Link

Livechat

Url	Short Description	Details Page
/api/v1/livechat/inquiries.list	Retrieves a list of open inquiries.	Link
/api/v1/livechat/inquiries.take	Take an open inquiry.	Link
/api/v1/livechat/rooms	Retrieves a list of livechat rooms.	Link

OAuthApps

Url	Short Description	Details Page
/api/v1/oauth-apps.get	Retrieves an OAuth App by id or client id.	Link
/api/v1/oauth-apps.list	Retrieves a list of OAuth Apps.	Link

Permissions

Url	Short Description	Details Page
/api/v1/permissions.listAll	Lists permissions on the server.	Link
/api/v1/permissions.update	Edits permissions on the server.	Link

Roles

Url	Short Description	Details Page
/api/v1/roles.create	Create a new role in the system.	Link
/api/v1/roles.list	Gets all the roles in the system.	Link
/api/v1/roles.sync	Gets all the roles in the system which are updated after a given date.	Link
/api/v1/roles.addUserToRole	Assign a role to an user.	Link
/api/v1/roles.getUsersInRole	Gets the users that belongs to a role.	Link

Push Token

Url	Method	Short Description	Details Page
/api/v1/push.token	POST	Saves push token.	Link
/api/v1/push.token	DELETE	Removes push token.	Link

Rooms

Url	Short Description	Details Page
/api/v1/rooms.adminRooms	Retrieve all rooms (requires special permission).	Link
/api/v1/rooms.cleanHistory	Cleans up a room's history, requires special permission.	Link
/api/v1/rooms.createDiscussion	Creates a new discussion.	Link

/api/v1/rooms.favorite	Favorite/Unfavorite room.	Link
/api/v1/rooms.get	Gets rooms.	Link
/api/v1/rooms.getDiscussions	Gets room's discussions.	Link
/api/v1/rooms.info	Gets info from a room.	Link
/api/v1/rooms.leave	Leaves a room.	Link
/api/v1/rooms.saveNotification	Sets the notifications settings of specific channel.	Link
/api/v1/rooms.upload/:rid	Upload a message with attached file.	Link

Teams

Url	Short Description	Details Page
/api/v1/teams.addMembers	Adds users to the team.	Link
/api/v1/teams.addRooms	Adds rooms to the team.	Link
/api/v1/teams.autocomplete	Lists the teams whose names match a given pattern.	Link
/api/v1/teams.create	Creates a new team.	Link
/api/v1/teams.delete	Removes a team.	Link
/api/v1/teams.info	Gets a team's information.	Link
/api/v1/teams.leave	Leaves a team.	Link
/api/v1/teams.list	Lists the public and private teams the caller is part of.	Link
/api/v1/teams.listAll	Lists all of the teams and their information.	Link

/api/v1/teams.listRooms	Lists all rooms of the team.	Link
/api/v1/teams.listRoomsOfUser	Lists only the team's rooms the user has joined.	Link
/api/v1/teams.members	Retrieves all team members.	Link
/api/v1/teams.removeMember	Removes a member from a team.	Link
/api/v1/teams.removeRoom	Removes a room from a team.	Link
/api/v1/teams.update	Updates an existing team (name and type).	Link
/api/v1/teams.updateMember	Updates a team member's roles, limited to permissions.	Link
/api/v1/teams.updateRoom	Updates a room from a team, limited to permissions.	Link

Command Methods

Url	Short Description	Details Page
/api/v1/commands.get	Get specification of the slash command.	Link
/api/v1/commands.list	Lists all available slash commands.	Link
/api/v1/commands.run	Execute a slash command in the specified room.	Link

Custom User Status

Url	Short Description	Details Page

/api/v1/custom-user-status.list	Lists all available custom user's status.	Link
---------------------------------	---	----------------------

Emoji Custom

Url	Short Description	Details Page
/api/v1/emoji-custom.list	List the custom emojis available.	Link
/api/v1/emoji-custom.create	Create new custom emoji.	Link
/api/v1/emoji-custom.delete	Delete an existent custom emoji.	Link
/api/v1/emoji-custom.update	Update an existent custom emoji.	Link

Settings

Url	Method	Short Description	Details Page
/api/v1/settings	GET	Lists all private settings.	Link
/api/v1/settings.public	GET	Lists all public settings.	Link
/api/v1/settings.oauth	GET	Return list of all available oauth services.	Link
/api/v1/service.configurations	GET	Lists all service configurations.	Link
/api/v1/settings/:_id	GET	Gets a setting.	Link
/api/v1/settings/:_id	POST	Updates a setting.	Link

Subscriptions



Url	Method	Short Description	Details Page
/api/v1/subscriptions.get	GET	Get all subscriptions.	Link
/api/v1/subscriptions.getOne	GET	Get the subscription by room Id.	Link
/api/v1/subscriptions.read	POST	Mark a room as read.	Link
/api/v1/subscriptions.unread	POST	Mark messages as unread.	Link

Video Conference

Url	Short Description	Details Page
/api/v1/video-conference/jitsi.update-timeout	Updates the timeout of Jitsi video conference in a channel.	Link

Webdav

Url	Short Description	Details Page
/api/v1/webdav.getMyAccounts	Retrieves the user's webdav accounts.	Link

Licenses

Url	Method	Short Description	Details Page
/api/v1/licenses.get	GET	Gets all active licenses.	Link
/api/v1/licenses.add	POST	Adds a new license.	Link

Language specific wrappers

Java

- [rocket-chat-rest-client](#)

PHP

- [rocketchat-php](#)

Python

- [rocketchat_API](#)
- [rocket-python](#)

Ruby

- [rocketchat-ruby](#)

Clojure

- [rocketchat-clojure](#)

Golang

- [rocketchat-golang](#)

Avatars

Avatars are fetched from the server for each room and user, and gives the client control over the size and format of the image that is returned.

URL	Requires Auth	HTTP Method
/avatar/{subject}	no	GET

Parameters

Argument	Example	Type	Required
subject	alice , @general	URL component	Required
format	png	Query String	Optional
size	50	Query String	Optional
rc_uid	aobEdbYhXfu5hkeqG	Query String	Optional
rc_token	9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq	Query String	Optional

Notes

- Whilst this is a RESTful endpoint, this sits separately from the REST API in the server codebase, and behaves slightly differently.
- The `rc_uid` and `rc_token` can alternately be provided as cookies (already present in the web interface), but not as auth headers.
- When requesting the avatar of a user, if they have a custom avatar set, this will override user choices of format or size
- If no uploaded avatar exists for a user, a default one will be generated for them based on the initial of their username in a randomly coloured square
 - If `UI_Use_Name_Avatar` is set AND the user has a real name set, the generated avatar will be based on their actual name instead of the username.
- If no format is requested, or the format requested isn't supported, the format returned will be SVG.

Example Call

```
curl http://localhost:3000/avatar/alice?size=50
```

Methods

Apps Endpoints

WhatsApp Endpoints

Template Message

URL

Input
Method Data
Format

REST API URL can be found on Apps Page

Sample Url for eg:

<http://localhost:3000/api/apps/public/30bf52a3-d65b-4ab1-9b11-7cfdddf1ef29/templateMessage>

POST

JSON

Request structure

```
1  {
2      "phoneNumbers": [
3          "917785887264", "551452001478"
4      ],
5      "connectedWhatsAppNo": "565412001234",
6      "targetAgent": "murtaza98",
7      "targetDepartment": "SalesDepartment",
8      "template": {
9          "namespace": "your-namespace",
10         "name": "your-template-name",
11         "language": {
12             "code": "your-language-and-locale-code",
13             "policy": "deterministic"
14         },
15         "components": [
16             {
17                 "type": "header",
18                 "parameters": [
19                     {
20                         "type": "text",
21                         "text": "your-text-string"
22                     }
23                 ]
24             },
25             {
26                 "type": "body",
27                 "parameters": [
28                     {
```

```

29                     "type": "text",
30                     "text": "your-text-string"
31                 },
32                 {
33                     "type": "text",
34                     "text": "your-text-string"
35                 }
36             ],
37         },
38         {
39             "type": "footer",
40             "parameters": [
41                 {
42                     "type": "text",
43                     "text": "your-text-string"
44                 }
45             ]
46         },
47     ]
48 }
49 }
```

Parameters

Param Name	Param Type	Description	Dependency	Example
phoneNumbers	Array of String	<p>Recipient WhatsApp Id(s) where you want to send the template message. Note you can validate a WhatsApp id from here</p>	Required	<pre>"phoneNumbers": ["917785887264", "551452001478"]</pre>

connectedWhatsAppNo	String	The WhatsApp Number from which you want to send this template message	Required	"connectedWhatsAppNo": "565412001234"
targetAgent	String	Username of the targetAgent you wish to transfer the chat to when the WhatsApp user replies back	Optional	"targetAgent": "murtaza98"
targetDepartment	String	Department Name or Id where you wish to transfer the chat to when the WhatsApp user replies back	Optional	"targetDepartment": "SalesDepartment"

template

Same as
WhatsApp-
template-
object

Provide
information
about the
template
message
you wish to
send

Required

Please refer to the
template property in al
examp

Sample Call

Curl

```
1 curl --location --request POST 'http://localhost:3000/api/apps/public/08
2 --header 'Content-Type: application/json' \
3 --data-raw '{
4     "phoneNumbers": [
5         "917738772967"
6     ],
7     "connectedWhatsAppNo": "565412001234",
8     "targetAgent": "murtaza98",
9     "targetDepartment": "SalesDepartment",
10    "template": {
11        "namespace": "aa498c38_6c0c_448d_a652_38f5d33b8c7f",
12        "name": "sales_welcome_template",
13        "language": {
14            "code": "en",
15            "policy": "deterministic"
16        }
17    }
18 }'
```

HTTP

```
1 POST /api/apps/public/085b7345-53e7-498c-adcb-d909e11f8100/templateMessa
2 Host: localhost:3000
3 Content-Type: application/json
4 Content-Length: 312
```

```
5
6  {
7    "phoneNumbers": [
8      "917738772967"
9    ],
10   "connectedWhatsAppNo": "565412001234",
11   "targetAgent": "murtaza98",
12   "targetDepartment": "SalesDepartment",
13   "template": {
14     "namespace": "aa498c38_6c0c_448d_a652_38f5d33b8c7f",
15     "name": "sales_welcome_template",
16     "language": {
17       "code": "en",
18       "policy": "deterministic"
19     }
20   }
21 }
```

Response

Success Response

Error Response

Code: 400 BAD REQUEST

Code: 200

Content:

```
{ "success": true }
```

Content:

```
{ "success": false, error: "Too many phone numbers!
Please make sure that phoneNumbers are <= 250" }
```

Code: 200

Content:

```
{ "success": true,
"invalidContacts": [
"917738772967" ]
}
```

Code: 400 BAD REQUEST

Content:

```
{ "success": false, error: "No Agent found with
username abc98" }
```

Code: 400 BAD REQUEST

Content:

```
{ "success": false, error: "No Department found with  
id/Name SalesDepartment" }
```

Code: 400 BAD REQUEST

Content:

```
{ "success": false, error: "Error! Missing  
"connectedWhatsAppNo" field within the request" }
```

Code: 400 BAD REQUEST

Content:

```
{ "success": false, error: "Connected WhatsApp Number  
(connectedWhatsAppNo) i.e. 565412001234 is not a valid  
WhatsApp contact. Contact Status is invalid" }
```

Code: 400 BAD REQUEST

Content:

```
{ "success": false, error: "Error! Connected WhatsApp  
Number (connectedWhatsAppNo) i.e. 565412001234 has not  
been registered yet on Rocket.Chat. Please goto  
#omnichannel-whatsapp-setup channel and connect this  
WhatsApp number first there before sending this request  
again." }
```

Code: 500 Internal Server Error

Content:

```
{ "success": false, error : "Error occurred while  
processing request. Details: [Error Details]" }
```

Assets

Url	Short Description	Details Page
/api/v1/assets.setAsset	Set an asset image by name.	Link
/api/v1/assets.unsetAsset	Unset an asset by name.	Link

Set Asset

Set an asset by name. Requires `manage-assets` permission.

URL	Requires Auth	HTTP Method
<code>/api/v1/assets.setAsset</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>assetName</code>	<code>mylogo.png</code>	Required	Asset File to upload.
<code>refreshAllClients</code>	<code>true</code>	Optional	If needs refresh all clients to apply changes immediately.

`assetName` can be one of these: [`logo` , `background` , `favicon_ico` , `favicon` , `favicon_16` , `favicon_32` , `favicon_192` , `favicon_512` , `touchicon_180` , `touchicon_180_pre` , `tile_70` , `tile_144` , `tile_150` , `tile_310_square` , `tile_310_wide` , `safari_pinned`]

Upload a file

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -F "logo=@mylogo.png" \
4     http://localhost:3000/api/v1/assets.setAsset
```

Example Result

```
1  {
2      "success": true
3 }
```

Change Log

Version	Description
0.69.0	Added

Unset Asset

Unset an asset by name. Requires `manage-assets` permission.

URL	Requires Auth	HTTP Method
<code>/api/v1/assets.unsetAsset</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>assetName</code>	<code>"logo"</code>	Required	Asset name to unset.
<code>refreshAllClients</code>	<code>true</code>	Optional	If needs refresh all clients to apply changes immediately.

`assetName` can be one of these: [`logo` , `background` , `favicon_ico` , `favicon` , `favicon_16` , `favicon_32` , `favicon_192` , `favicon_512` , `touchicon_180` , `touchicon_180_pre` , `tile_70` , `tile_144` , `tile_150` , `tile_310_square` , `tile_310_wide` , `safari_pinned`]

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/assets.unsetAsset \
4   -H "Content-Type:application/json" \
5   -d '{ "assetName": "logo", "refreshAllClients": true}'
```

Example Result

```
1  {
2    "success": true
3 }
```

Change Log

Version	Description
0.69.0	Added

Authentication

The authentication with the REST API is a two step process.

1. Passing your username and password to the `/api/v1/login`
2. Using the `authToken` and `userId` provided back on every method

Url	Short Description	Details Page
<code>/api/v1/login</code>	Authenticate with the REST API.	Link
<code>/api/v1/login</code>	Authenticate with facebook.	Link
<code>/api/v1/login</code>	Authenticate with google.	Link
<code>/api/v1/login</code>	Authenticate with twitter.	Link
<code>/api/v1/logout</code>	Invalidate your REST API authentication token.	Link
<code>/api/v1/me</code>	Displays information about the authenticated user.	Link

Login with Facebook

Authenticate with Facebook.

URL	Requires Auth	HTTP Method
/api/v1/login	no	POST

Payload

Argument	Example	Required	Description
serviceName	facebook	Required	The desired OAuth service name
accessToken	hash	Required	Access token provided by facebook oauth
secret	hash	Required	The secret provided by facebook for app
expiresIn	200	Required	Lifetime of token(in seconds)

Example Call - As JSON

```
1 curl -H "Content-type:application/json" \
2       http://localhost:3000/api/v1/login \
3       -d '{ "serviceName": "facebook", "accessToken": "hash",
4             "secret": "hash", "expiresIn": 200 }'
```

Result

```
1  {
2      "status": "success",
3      "data": {
4          "authToken": "9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq",
5          "userId": "aobEdbYhXfu5hkeqG",
6          "me": {
7              "_id": "aYjNnig8BEAWeQzMh",
8              "name": "Rocket Cat",
9              "emails": [
10                  {
11                      "address": "rocket.cat@rocket.chat",
12                      "verified": false
13                  }
14              ],
15              "status": "offline",
16              "statusConnection": "offline",
17              "username": "rocket.cat",
18              "utcOffset": -3,
19              "active": true,
20              "roles": [
21                  "admin"
22              ],
23              "settings": {
24                  "preferences": {}
25              },
26              "avatarUrl": "http://localhost:3000/avatar/test"
27          }
28      }
29  }
```

Change Log

Version	Description
1.0.0	Added <code>avatarUrl</code> property to response
0.64.0	Added <code>me</code> property to response
0.63.0	Added

Logout

Invalidate your REST API authentication token.

URL	Requires Auth	HTTP Method
/api/v1/logout	yes	POST

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -X POST http://localhost:3000/api/v1/logout
```

Result

```
1 {
2   "status": "success",
3   "data": {
4     "message": "You've been logged out!"
5   }
6 }
```

Login

Login with your username and password.

URL	Requires Auth	HTTP Method
/api/v1/login	no	POST

Payload

Argument	Example	Required	Description
user	foo@bar.com OR myusername	Optional	Your username or email
password	my\$up3rP@ssw0rd	Optional	Your password
resume	9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq	Optional	Your previously issued authToken

Notes

- Whilst none of the arguments are required in every request, every request requires some arguments to be provided. You MUST provide either `user` AND `password`, or provide `resume`.
- You will need to provide the `authToken` and `userId` for any of the authenticated methods.
- If your user has two-factor(2FA) authentication enabled, you must send a request like [this](#).

- If LDAP authentication is enabled, you must maintain the login in the same way as you normally do. Similarly if 2FA is enabled for an LDAP user. Everything stays the same.
-

Example Call - As Form Data

```
1 curl http://localhost:3000/api/v1/login \
2       -d "user=myusername&password=mypassword"
```

```
1 curl http://localhost:3000/api/v1/login \
2       -d "user=my@email.com&password=mypassword"
```

Example Call - As JSON

```
1 curl -H "Content-type:application/json" \
2       http://localhost:3000/api/v1/login \
3       -d '{ "user": "myusername", "password": "mypassword" }'
```

```
1 curl -H "Content-type:application/json" \
2       http://localhost:3000/api/v1/login \
3       -d '{ "user": "my@email.com", "password": "mypassword" }'
```

Example Call - When two-factor(2FA) authentication is enabled

```
1 curl -H "Content-type:application/json" \
2       http://localhost:3000/api/v1/login \
3       -d '{ "user": "myusername", "password": "mypassword", "code": "22461"
```

```
1 curl -H "Content-type:application/json" \
2       http://localhost:3000/api/v1/login \
3       -d '{ "user": "my@email.com", "password": "mypassword", "code": "22461"
```

Result

```
1  {
2      "status": "success",
3      "data": {
4          "authToken": "9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq",
5          "userId": "aobEdbYhXfu5hkeqG",
6          "me": {
7              "_id": "aYjNnig8BEAWeQzMh",
8              "name": "Rocket Cat",
9              "emails": [
10                  {
11                      "address": "rocket.cat@rocket.chat",
12                      "verified": false
13                  }
14              ],
15              "status": "offline",
16              "statusConnection": "offline",
17              "username": "rocket.cat",
18              "utcOffset": -3,
19              "active": true,
20              "roles": [
21                  "admin"
22              ],
23              "settings": {
24                  "preferences": {}
25              },
26              "avatarUrl": "http://localhost:3000/avatar/test"
27          }
28      }
29  }
```

Change Log

Version	Description
1.0.0	Added <code>avatarUrl</code> property to response
0.64.0	Added <code>me</code> property to response
0.60.0	Added

Login with Twitter

Authenticate with Twitter.

URL	Requires Auth	HTTP Method
/api/v1/login	no	POST

Payload

Argument	Example	Required	Description
serviceName	twitter	Required	The desired OAuth service name
accessToken	hash	Required	Access token provided by twitter oauth
accessTokenSecret	hash	Required	Access token secret provided by twitter oauth
appSecret	hash	Required	The secret provided by twitter for app
appId	hash	Required	The id provided by twitter for app
expiresIn	200	Required	Lifetime of token(in seconds)

Example Call - As JSON

```
1 curl -H "Content-type:application/json" \
2       http://localhost:3000/api/v1/login \
3       -d '{ "serviceName": "twitter", "accessToken": "hash", accessTokenSe
4       "appSecret": "hash", "appId": "hash", "expiresIn": 200}'
```

Result

```
1  {
2      "status": "success",
3      "data": {
4          "authToken": "9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq",
5          "userId": "aobEdbYhXfu5hkeqG",
6          "me": {
7              "_id": "aYjNnig8BEAWeQzMh",
8              "name": "Rocket Cat",
9              "emails": [
10                  {
11                      "address": "rocket.cat@rocket.chat",
12                      "verified": false
13                  }
14              ],
15              "status": "offline",
16              "statusConnection": "offline",
17              "username": "rocket.cat",
18              "utcOffset": -3,
19              "active": true,
20              "roles": [
21                  "admin"
22              ],
23              "settings": {
24                  "preferences": {}
25              },
26              "avatarUrl": "http://localhost:3000/avatar/test"
27          }
28      }
29  }
```

Change Log

Version	Description
1.0.0	Added <code>avatarUrl</code> property to response
0.64.0	Added <code>me</code> property to response

0.63.0

Added

Me

Quick information about the authenticated user.

URL	Requires Auth	HTTP Method
/api/v1/me	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/me
```

Example Result

```
1 {
2   "_id": "aobEdbYhXfu5hkeqG",
3   "name": "Example User",
4   "emails": [
5     {
6       "address": "example@example.com",
7       "verified": true
8     }
9   ],
10  "status": "offline",
11  "statusConnection": "offline",
12  "username": "example",
13  "utcOffset": 0,
14  "active": true,
15  "roles": [
16    "user",
17    "admin"
18  ],
```

```

19   "settings": {
20     "preferences": {
21       "enableAutoAway": false,
22       "idleTimeoutLimit": 300,
23       "desktopNotificationDuration": 0,
24       "audioNotifications": "mentions",
25       "desktopNotifications": "mentions",
26       "mobileNotifications": "mentions",
27       "unreadAlert": true,
28       "useEmojis": true,
29       "convertAsciiEmoji": true,
30       "autoImageLoad": true,
31       "saveMobileBandwidth": true,
32       "collapseMediaByDefault": false,
33       "hideUsernames": false,
34       "hideRoles": false,
35       "hideFlexTab": false,
36       "hideAvatars": false,
37       "roomsListExhibitionMode": "category",
38       "sidebarViewMode": "medium",
39       "sidebarHideAvatar": false,
40       "sidebarShowUnread": false,
41       "sidebarShowFavorites": true,
42       "sendOnEnter": "normal",
43       "messageViewMode": 0,
44       "emailNotificationMode": "all",
45       "roomCounterSidebar": false,
46       "newRoomNotification": "door",
47       "newMessageNotification": "chime",
48       "muteFocusedConversations": true,
49       "notificationsSoundVolume": 100
50     }
51   },
52   "customFields": {
53     "twitter": "@userstwi"
54   },
55   "avatarUrl": "http://localhost:3000/avatar/test",
56   "customFields": {
57     "twitter": "@userstwi"
58   },
59   "success": true
60 }

```

Note

- The `customFields` will not be returned if it does not exist on the server.

Change Log

Version	Description
1.0.0	Added <code>avatarUrl</code> property to response
0.68.0	Added <code>customFields</code> property.
0.48.0	Added

Login with Google

Authenticate with Google.

URL	Requires Auth	HTTP Method
/api/v1/login	no	POST

Payload

Argument	Example	Required	Description
serviceName	google	Required	The desired OAuth service name
accessToken	hash	Required	Access token provided by google oauth
idToken	hash	Required	Id token provided by google auth
expiresIn	200	Required	Lifetime of token(in seconds)
scope	profile	Optional	Google scopes for API

Example Call - As JSON

```
1 curl -H "Content-type:application/json" \
2       http://localhost:3000/api/v1/login \
3       -d '{ "serviceName": "google", "accessToken": "hash",
4             "idToken": "hash", "expiresIn": 200, "scope": "profile" }'
```

Result

```
1  {
2      "status": "success",
3      "data": {
4          "authToken": "9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq",
5          "userId": "aobEdbYhXfu5hkeqG",
6          "me": {
7              "_id": "aYjNnig8BEAWeQzMh",
8              "name": "Rocket Cat",
9              "emails": [
10                  {
11                      "address": "rocket.cat@rocket.chat",
12                      "verified": false
13                  }
14              ],
15              "status": "offline",
16              "statusConnection": "offline",
17              "username": "rocket.cat",
18              "utcOffset": -3,
19              "active": true,
20              "roles": [
21                  "admin"
22              ],
23              "settings": {
24                  "preferences": {}
25              },
26              "avatarUrl": "http://localhost:3000/avatar/test"
27          }
28      }
29  }
```

Change Log

Version	Description
1.0.0	Added <code>avatarUrl</code> property to response
0.64.0	Added <code>me</code> property to response

0.60.0

Added

Auto Translate

To be able to use the auto translate feature, you must enable it on the admin panel under (Admin → Message → Auto-Translate) and the user must have the necessary permission (auto-translate) in addition to a valid Google Translate API token.

Url	Short Description	Details Page
/api/v1/autotranslate.getSupportedLanguages	Get the supported languages by the autotranslate.	Link
/api/v1/autotranslate.saveSettings	Save some settings about autotranslate.	Link
/api/v1/autotranslate.translateMessage	Translate the message.	Link

Get Supported Languages

Get the supported languages by the autotranslate.

URL	Requires Auth	HTTP Method
/api/v1/autotranslate.getSupportedLanguages	yes	GET

Query Parameters

Argument	Example	Required	Description
targetLanguage	en	No	Language in which language names will be returned.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/autotranslate.getSupportedLanguages?targ
```

Example Result

```
1 {
2     "languages": [
3         {
4             "language": "af",
5             "name": "Africâner"
```

```
6      },
7      {
8          "language": "sq",
9          "name": "Albanês"
10     },
11     {
12         "language": "de",
13         "name": "Alemão"
14     },
15     {
16         "language": "am",
17         "name": "Amárico"
18     }
19 ],
20     "success": true
21 }
```

Change Log

Version	Description
1.3.0	Added

Save Settings

Save some settings about autotranslate.

URL	Requires Auth	HTTP Method
/api/v1/autotranslate.saveSettings	yes	POST

Payload

Argument	Example	Required	Description
roomId	7aDSXtjMA3KPLxLjt	Required	The room's id to apply setting.
field	autoTranslate Or autoTranslateLanguage	Required	The setting to apply to user's subscription.
value	true Or en	Required	Boolean if the setting is autoTranslate and a string (the language) if the setting is autoTranslateLanguage .
defaultLanguage	en	Optional	The default language.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: Z3cpIYN6CNK2oXWKv" \
3     -H "Content-type:application/json" \
4     http://localhost:3000/api/v1/autotranslate.saveSettings \
```

```
5      -d '{ "roomId": "7aDSXtjMA3KPLxLjt", "field": "autoTranslate", "value
```

Example Result

```
1  {
2    "success": true
3 }
```

Change Log

Version	Description
1.3.0	Added

Translate Message

Translate the provided message.

URL	Requires Auth	HTTP Method
/api/v1/autotranslate.translateMessage	yes	POST

Payload

Argument	Example	Required	Description
messageId	Fq7sQNSnDEFzj8qoS	Required	The message's id to be translated.
targetLanguage	en	Optional	The target language that the message will be translated.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: Z3cpIYN6CNK2oXWKv" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/autotranslate.translateMessage \
5   -d '{
6     "messageId": "Fq7sQNSnDEFzj8qoS",
7     "targetLanguage": "en"
8   }'
```

Example Result

```
1  {
2    "message": {
3      "_id": "Fq7sQNSnDEFzj8qoS",
4      "rid": "GENERAL",
5      "msg": "Isso é um teste",
6      "ts": "2019-06-27T15:35:20.753Z",
7      "u": {
8        "_id": "pC6Z2N2ijivxdsYSu",
9        "username": "marcos",
10       "name": "marcos"
11     },
12     "_updatedAt": "2019-06-27T15:47:01.486Z",
13     "mentions": [],
14     "channels": [],
15     "translations": {
16       "en": "This is a test"
17     }
18   },
19   "success": true
20 }
```

Change Log

Version	Description
1.3.0	Added

Commands

Url	Short Description	Details Page
/api/v1/commands.get	Get specification of the slash command.	Link
/api/v1/commands.list	Lists all available slash commands.	Link
/api/v1/commands.run	Execute a slash command in the specified room.	Link
/api/v1/commands.preview	Gets the preview data for the command and execute preview item.	Link

Get

Get specification of the slash command.

URL	Requires Auth	HTTP Method
/api/v1/commands.get	yes	GET

Payload

Argument	Example	Required	Description
command	me	Required	The name of the command.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/commands.get?command=me
```

Example Result

```
1 {
2   "command": {
3     "command": "me",
4     "params": "your_message",
5     "description": "Displays_action_text",
6     "clientOnly": false
7   },
8 }
```

```
8      "success": true
9  }
```

Change Log

Version	Description
0.60.2	Added

List

Lists all available slash commands.

URL	Requires Auth	HTTP Method
/api/v1/commands.list	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/commands.list
```

Example Result

```
1 {
2   "commands": [
3     {
4       "command": "invite-all-from",
5       "clientOnly": false
6     },
7     {
8       "command": "slackbridge-import",
9       "clientOnly": false
10    },
11    {
12      "command": "gimme",
13      "params": "your_message_optional",
14      "description": "Slash_Gimme_Description",
15      "clientOnly": false
16    },
17    {
18      "command": "lennyface",
```

```
19         "params": "your_message_optional",
20         "description": "Slash_LennyFace_Description",
21         "clientOnly": false
22     },
23     {
24         "command": "shrug",
25         "params": "your_message_optional",
26         "description": "Slash_Shrug_Description",
27         "clientOnly": false
28     },
29     {
30         "command": "tableflip",
31         "params": "your_message_optional",
32         "description": "Slash_Tableflip_Description",
33         "clientOnly": false
34     },
35     {
36         "command": "unflip",
37         "params": "your_message_optional",
38         "description": "Slash_TableUnflip_Description",
39         "clientOnly": false
40     },
41     {
42         "command": "create",
43         "clientOnly": false
44     },
45     {
46         "command": "help",
47         "clientOnly": false
48     },
49     {
50         "command": "invite",
51         "clientOnly": false
52     },
53     {
54         "command": "invite-all-to",
55         "clientOnly": false
56     },
57     {
58         "command": "archive",
59         "clientOnly": false
60     },
61     {
62         "command": "join",
63         "clientOnly": false
64     },
65     {
66         "command": "kick",
67         "clientOnly": false
68     },
69     {
```

```
70         "command": "leave",
71         "clientOnly": false
72     },
73     {
74         "command": "part",
75         "clientOnly": false
76     },
77     {
78         "command": "me",
79         "params": "your_message",
80         "description": "Displays_action_text",
81         "clientOnly": false
82     },
83     {
84         "command": "msg",
85         "clientOnly": false
86     },
87     {
88         "command": "mute",
89         "clientOnly": false
90     },
91     {
92         "command": "unmute",
93         "clientOnly": false
94     },
95     {
96         "command": "topic",
97         "params": "Slash_Topic_Params",
98         "description": "Slash_Topic_Description",
99         "clientOnly": false
100    },
101    {
102        "command": "unarchive",
103        "clientOnly": false
104    }
105 ],
106 "offset": 0,
107 "count": 22,
108 "total": 22,
109 "success": true
110 }
```

Change Log

Version	Description
0.60.2	Added

Preview

Fetches the command's preview data or executes the preview item

Get command's preview data

URL	Requires Auth	HTTP Method
/api/v1/commands.preview	yes	GET

Payload

Argument	Example	Required	Description
command	unmute	Required	The name of the command to be executed.
roomId	ByehQjC44FwMeiLbX	Required	The ID of the room.
params	@user123	Optional Default: undefined	Parameters of the command if required.

Execute a command's preview item

URL	Requires Auth	HTTP Method
/api/v1/commands.preview	yes	POST

Payload

Argument	Example	Required	Description
command	unmute	Required	The name of the command to be executed.
roomId	ByehQjC44FwMeiLbX	Required	The ID of the room.
tmid	Hsj2435g4bvrj4bjh	Optional	The ID of the thread to run the command.
params	@user123	Optional Default: undefined	Parameters of the command if required.
triggerId	awovufbukuefzuper	Optional	The triggerId generated by the client that allows the command to interact with the UI
previewItem	Object	Required	The data of the preview item to be executed by the command
previewItem.id	nucobdpokaikazia	Required	The ID of the selected preview item
previewItem.type	image	Required	The type of the selected preview item
priviewItem.value	https://dev.null/gif	Required	The value of the selected preview item

Change Log

Version	Description

3.0.0	Added <code>tmid</code> and <code>triggerId</code> fields
0.65.0	Added

Run

Execute a slash command in the specified room.

URL	Requires Auth	HTTP Method
/api/v1/commands.run	yes	POST

Payload

Argument	Example	Required	Description
command	unmute	Required	The name of the command to be executed.
roomId	ByehQjC44FwMeiLbX	Required	The ID of the room.
params	@user123	Optional Default: undefined	Parameters of the command if required.
tmid	Hsj2435g4bvrj4bjh	Optional	The ID of the thread to run the command.
triggerId	awovufbukuefzuper	Optional	The triggerId generated by the client that allows the command to interact with the UI

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
```

```
3      -H "Content-type: application/json" \
4      http://localhost:3000/api/v1/commands.run \
5      -d '{"command": "unmute", "roomId": "ByehQjC44FwMeiLbX", "params": "@user1"}'
```

Example Result

```
1  {
2    "success": true
3 }
```

Change Log

Version	Description
3.0.0	Added <code>triggerId</code> property
2.0.0	Added <code>tmid</code> property
0.60.2	Added

Channels

These methods apply to public channels only. Use `groups.*` methods for private channels.

Url	Short Description	Details Page
<code>/api/v1/channels.addAll</code>	Adds all of the users on the server to a channel.	Link
<code>/api/v1/channels.addLeader</code>	Gives the role of Leader for a user in the current channel.	Link
<code>/api/v1/channels.addModerator</code>	Gives the role of moderator to a user in a channel.	Link
<code>/api/v1/channels.addOwner</code>	Gives the role of owner to a user in a channel.	Link
<code>/api/v1/channels.anonymousread</code>	Gets the messages in public channels to an anonymous user	Link
<code>/api/v1/channels.archive</code>	Archives a channel.	Link
<code>/api/v1/channels.close</code>	Removes a channel from a user's list of channels.	Link
<code>/api/v1/channels.counters</code>	Gets channel counters.	Link
<code>/api/v1/channels.create</code>	Creates a new channel.	Link
<code>/api/v1/channels.delete</code>	Removes a channel.	Link
<code>/api/v1/channels.getAllUserMentionsByChannel</code>	Gets all the mentions of a channel.	Link

/api/v1/channels.files	Gets a list of files from a channel.	Link
/api/v1/channels.getIntegrations	Gets the channel's integration.	Link
/api/v1/channels.history	Retrieves the messages from a channel.	Link
/api/v1/channels.info	Gets a channel's information.	Link
/api/v1/channels.invite	Adds a user to a channel.	Link
/api/v1/channels.join	Joins yourself to a channel.	Link
/api/v1/channels.kick	Removes a user from a channel.	Link
/api/v1/channels.leave	Removes the calling user from a channel.	Link
/api/v1/channels.list	Retrieves all of the channels from the server.	Link
/api/v1/channels.list.joined	Gets only the channels the calling user has joined.	Link
/api/v1/channels.members	Retrieves all channel users.	Link
/api/v1/channels.messages	Retrieves all channel messages.	Link
/api/v1/channels.moderators	List all moderators of a channel.	Link
/api/v1/channels.online	List all online users of a channel.	Link

/api/v1/channels.open	Adds the channel back to the user's list of channels.	Link
/api/v1/channels.removeleader	Removes the role of Leader for a user in the current channel.	Link
/api/v1/channels.removeModerator	Removes the role of moderator from a user in a channel.	Link
/api/v1/channels.removeOwner	Removes the role of owner from a user in a channel.	Link
/api/v1/channels.rename	Changes a channel's name.	Link
/api/v1/channels.roles	Gets the user's roles in the channel.	Link
/api/v1/channels.setAnnouncement	Sets a channel's announcement.	Link
/api/v1/channels.setCustomFields	Sets a channel's custom fields.	Link
/api/v1/channels.setDefault	Sets a channel's default status.	Link
/api/v1/channels.setDescription	Sets a channel's description.	Link
/api/v1/channels.setJoinCode	Sets the channel's code required to join it.	Link
/api/v1/channels.setPurpose	Sets a channel's description.	Link

/api/v1/channels.setReadOnly	Sets whether a channel is read only or not.	Link
/api/v1/channels.setTopic	Sets a channel's topic.	Link
/api/v1/channels.setType	Sets the type of room the channel should be.	Link
/api/v1/channels.unarchive	Unarchives a channel.	Link

Channel Add All

Adds all of the users of the Rocket.Chat server to the channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.addAll	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
activeUsersOnly	true	Optional Default: false	Add active users only

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.addAll \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
```

```
2     "channel": {
3         "_id": "ByehQjC44FwMeiLbX",
4         "name": "channelname",
5         "t": "c",
6         "usernames": [
7             "example",
8             "rocket.cat"
9         ],
10        "msgs": 0,
11        "u": {
12            "_id": "aobEdbYhXfu5hkeqG",
13            "username": "example"
14        },
15        "ts": "2016-05-30T13:42:25.304Z"
16    },
17    "success": true
18 }
```

Change Log

Version	Description
0.55.0	Added <code>activeUsersOnly</code> param
0.48.0	Renamed to <code>channels.addAll</code> from <code>channel.addAll</code>

Channel Add Leader

Gives the role of Leader for a user in the current channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.addLeader	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
userId	oCHkav5Zf6vmpu2W2	Required	The user's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.addLeader \
5   -d '{"roomId": "ByehQjC44FwMeiLbX", "userId": "oCHkav5Zf6vmpu2W2"}'
```

Success Example Result

```
1 {
2   "success": true
```

```
3 }
```

Bad Request Example Result

If the user is already a leader, it will return a `400 bad request` status.

```
1 {
2   "success": false,
3   "error": "User is already a leader [error-user-already-leader]",
4   "errorType": "error-user-already-leader"
5 }
```

Change Log

Version	Description
0.75.0	Added.

Channel Add Moderator

Gives the role of moderator for a user in the current channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.addModerator	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
userId	nSYqWzz4GsKTX4dyK	Required	The user id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.addModerator \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "userId": "nSYqWzz4GsKTX4dyK" }'
```

Example Result

```
1 {
2   "success": true
```

```
3 }
```

Bad Request Example Result

If the user is already a moderator, it will return a `400 bad request` status.

```
1 {
2   "success": false,
3   "error": "User is already a moderator [error-user-already-moderator]",
4   "errorType": "error-user-already-moderator"
5 }
```

Change Log

Version	Description
0.49.4	Added

Channel Add Owner

Gives the role of owner for a user in the current channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.addOwner	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
userId	nSYqWzz4GsKTX4dyK	Required	The user id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.addOwner \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "userId": "nSYqWzz4GsKTX4dyK" }'
```

Example Result

```
1 {
2     "success": true
```

```
3 }
```

Bad Request Example Result

If the user is already a owner, it will return a `400 bad request` status.

```
1 {
2   "success": false,
3   "error": "User is already a owner [error-user-already-owner]",
4   "errorType": "error-user-already-owner"
5 }
```

Change Log

Version	Description
0.49.4	Added

Channel Anonymous Read

Gets the messages in public channels to an anonymous user, if the server has the setting `Accounts_AllowAnonymousRead` enabled. It supports the [Offset](#), [Count](#), and [Sort Query Parameters](#) along with [Query and Fields Query Parameters](#).

URL	Requires Auth	HTTP Method
<code>/api/v1/channels.anonymousread</code>	no	GET

Payload

Argument	Example	Required	Description
<code>roomId</code>	<code>ByehQjC44FwMeiLbX</code>	Required(if no <code>roomName</code>)	The channel's id
<code>roomName</code>	<code>testChannel</code>	Required(if no <code>roomId</code>)	The channel's name

Example Call

```
1 curl -H "Content-type: application/json" \
2     http://localhost:3000/api/v1/channels.anonymousread?roomId=ByehQjC44F
```

Example Result

```
1 {
2     "messages": [
```

```
3      {
4          "_id": "xadufzmxzYQp4H9py",
5          "alias": "test",
6          "msg": "Example message",
7          "attachments": [],
8          "parseUrls": true,
9          "bot": {
10              "i": "MnQyfhWt5LqZotyfc"
11          },
12          "groupable": false,
13          "ts": "2018-10-05T01:10:47.524Z",
14          "u": {
15              "_id": "rocket.cat",
16              "username": "rocket.cat",
17              "name": "Rocket.Cat"
18          },
19          "rid": "GENERAL",
20          "_updatedAt": "2018-10-05T13:42:51.163Z",
21          "reactions": {
22              ":grin:": {
23                  "usernames": [
24                      "marcos.defendi"
25                  ]
26              }
27          },
28          "mentions": [],
29          "channels": [],
30          "starred": {
31              "_id": "KPkEYwKKBKZnEEPpt"
32          }
33      },
34      {
35          "_id": "id-1538701845987",
36          "rid": "GENERAL",
37          "msg": "This message was edited via API",
38          "alias": "Gruggy",
39          "emoji": ":smirk:",
40          "avatar": "http://res.guggy.com/logo_128.png",
41          "attachments": [
42              {
43                  "collapsed": false,
44                  "color": "#ff0000",
45                  "text": "Yay for gruggy!",
46                  "ts": "2016-12-09T16:53:06.761Z",
47                  "message_link": "https://google.com",
48                  "thumb_url": "http://res.guggy.com/logo_128.png",
49                  "author_name": "Bradley Hilton",
50                  "author_link": "https://rocket.chat/",
51                  "author_icon": "https://avatars.githubusercontent.com/",
52                  "title": "Attachment Example",
53                  "title_link": "https://youtube.com",
```

```
54         "title_link_download": true,
55         "image_url": "http://res.guggy.com/logo_128.png",
56         "audio_url": "http://www.w3schools.com/tags/horse.mp3"
57         "video_url": "http://www.w3schools.com/tags/movie.mp4"
58     "fields": [
59         {
60             "short": true,
61             "title": "Test",
62             "value": "Testing out something or other"
63         },
64         {
65             "short": true,
66             "title": "Another Test",
67             "value": "[Link](https://google.com/) somethin
68         }
69     ]
70 ],
71 "ts": "2018-10-05T01:10:45.994Z",
72 "u": {
73     "_id": "rocketchat.internal.admin.test",
74     "username": "rocketchat.internal.admin.test",
75     "name": "RocketChat Internal Admin Test"
76 },
77 "_updatedAt": "2018-10-05T01:10:47.064Z",
78 "editedBy": {
79     "_id": "rocketchat.internal.admin.test",
80     "username": "rocketchat.internal.admin.test"
81 },
82 "editedAt": "2018-10-05T01:10:46.384Z",
83 "reactions": {
84     ":smile:": {
85         "usernames": [
86             "rocketchat.internal.admin.test"
87         ]
88     },
89     ":squid:": {
90         "usernames": [
91             "rocketchat.internal.admin.test"
92         ]
93     },
94     ":bee:": {
95         "usernames": [
96             "rocketchat.internal.admin.test"
97         ]
98     },
99     ":ant:": {
100         "usernames": [
101             "rocketchat.internal.admin.test"
102         ]
103     }
104 }
```

```
105     },
106     "mentions": [],
107     "channels": [],
108     "urls": []
109   },
110 ],
111 "count": 2,
112 "offset": 0,
113 "total": 2,
114 "success": true
115 }
```

Change Log

Version	Description
1.2.0	Added

Channel Archive

Channel Archive

Archives a channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.archive	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -H "Content-type: application/json" \
4     http://localhost:3000/api/v1/channels.archive \
5     -d '{ "roomId": "ByehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
2     "success": true
3 }
```

Bad Request Example Result

If the channel is already archived, it will return a `400 bad request` status.

```
1  {
2    "success": false,
3    "error": "The channel, {Channel name}, is archived [error-room-archived]
4    "errorType": "error-room-archived"
5 }
```

Change Log

Version	Description
0.48.0	Added

Channel Close

Removes the channel from the user's list of channels.

URL	Requires Auth	HTTP Method
/api/v1/channels.close	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.close \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.48.0	Added

Channel Counters

Gets channel counters.

URL	Requires Auth	HTTP Method
/api/v1/channels.counters	yes	GET

Payload

One of roomId or roomName is required.

Argument	Example	Required	Description
roomId	GENERAL	Required	The channel's id
roomName	general	Required	The channel's name
userId	RtycPC29hqLJfT9xj	Optional	Counters for provided user id (need to have a view-room-administration right for calling user)

Example Call

```
1 curl -H "X-Auth-Token: 8h2mKAwxB3AQrFSjLVKMooJyjdCFaA7W45sWlHP8IzO" \
2     -H "X-User-Id: ew28FnZqipDpvKw3R" \
3     -H "Content-type: application/json" \
4     http://localhost:3000/api/v1/channels.counters?roomId=GENERAL
```

Example Result

```
1  {
2    "joined":true,
3    "members":78,
4    "unread":2,
5    "unreadFrom":"2018-02-23T17:15:51.907Z",
6    "msgs":304,
7    "latest":"2018-02-23T17:17:03.110Z",
8    "userMentions":0,
9    "success":true
10 }
11
12 where:
13
14 joined - boolean flag that shows that user is joined the room or not
15 members - count of current room members
16 unread - count of unread messages for specified user (calling user or pro
17 unreadFrom - start date-time of unread interval for specified user
18 msgs - count of messages in the room
19 latest - end date-time of unread interval for specified user (or date-time
20 userMentions - count of user mentions in messages
```

Change Log

Version	Description
0.65.0	Added

Channel Create

Creates a new public channel, optionally including specified users. The channel creator is always included.

URL	Requires Auth	HTTP Method
/api/v1/channels.create	yes	POST

Payload

Argument	Example	Required	Description
name	channelname	Required	The name of the new channel
members	["rocket.cat"]	Optional []	Default: The users to add to the channel when it is created.
readOnly	true	Optional false	Default: Set if the channel is read only or not.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.create \
5   -d '{ "name": "channelname" }'
```

Example Result

```
1  {
2      "channel": {
3          "_id": "ByehQjC44FwMeiLbX",
4          "name": "channelname",
5          "t": "c",
6          "usernames": [
7              "example"
8          ],
9          "msgs": 0,
10         "u": {
11             "_id": "aobEdbYhXfu5hkeqG",
12             "username": "example"
13         },
14         "ts": "2016-05-30T13:42:25.304Z"
15     },
16     "success": true
17 }
```

Change Log

Version	Description
0.13.0	Added

Channel Delete

Remove a public channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.delete	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required (if no roomName)	The channel's id
roomName	general	Required (if no roomId)	The channel's name

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.delete \
5   -d '{ "roomName": "channelname" }'
```

Example Result

```
1 {
2   "success": true
```

```
3 }
```

Change Log

Version	Description
0.71.0	Removed <code>channel</code> property
0.49.0	Added

Channel Files

Retrieves the files from a channel. It supports the **Offset**, **Count**, and **Sort Query Parameters** along with **Query and Fields Query Parameters**.

URL	Requires Auth	HTTP Method
/api/v1/channels.files	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX		The room id.
roomName	test	Required	The room name.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/channels.files?roomId=ByehQjC44FwMeiLbX
```

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/channels.files?roomName=test
```

Example Result

```
1  {
2      "files": [
3          {
4              "_id": "S78TNnvaWGwdYRaCD",
5              "name": "images.jpeg",
6              "size": 9778,
7              "type": "image/jpeg",
8              "rid": "GENERAL",
9              "description": "",
10             "store": "GridFS:Uploads",
11             "complete": true,
12             "uploading": false,
13             "extension": "jpeg",
14             "progress": 1,
15             "user": {
16                 "_id": "ksKsKmrjvxzkzxkw",
17                 "username": "rocket.cat",
18                 "name": "Rocket Cat"
19             },
20             "_updatedAt": "2018-03-08T14:47:37.003Z",
21             "instanceId": "uZG54xuoKauKHykbQ",
22             "etag": "jPavis9qG22xC5sDC",
23             "path": "/ufs/GridFS:Uploads/S78TNnvaWGwdYRaCD/images.jpeg",
24             "token": "28cAb868d9",
25             "uploadedAt": "2018-03-08T14:47:37.295Z",
26             "url": "/ufs/GridFS:Uploads/S78TNnvaWGwdYRaCD/images.jpeg"
27         }
28     ],
29     "count": 1,
30     "offset": 0,
31     "total": 1,
32     "success": true
33 }
```

Change Log

Version	Description
0.64.0	Change <code>userId</code> to <code>user</code> object in response

0.59.0

Added

Channel History

Retrieves the messages from a channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.history	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
latest	2016-09-30T13:42:25.304Z	Optional Default: now	The end of time range of messages to retrieve
oldest	2016-05-30T13:42:25.304Z	Optional Default: n/a	The start of the time range of messages to retrieve
inclusive	true	Optional Default: false	Whether messages which land on latest and oldest should be included
offset	10	Optional Default: 0	The offset of the list messages to retrieve
count	100	Optional Default: 20	The amount of messages to retrieve
unread	false	Optional Default: false	Whether the amount of unread should be included.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/channels.history?roomId=ByehQjC44FwMeiLb
```

Example Result

```
1 {
2   "messages": [
3     {
4       "_id": "AkzpHAvZpdnuchw2a",
5       "rid": "ByehQjC44FwMeiLbX",
6       "msg": "hi",
7       "ts": "2016-12-09T12:50:51.555Z",
8       "u": {
9         "_id": "y65tAmHs93aDChMWu",
10        "username": "testing"
11      },
12      "_updatedAt": "2016-12-09T12:50:51.562Z"
13    },
14    {
15      "_id": "vkLMxccR4MuTxreF",
16      "t": "uj",
17      "rid": "ByehQjC44FwMeiLbX",
18      "ts": "2016-12-08T15:41:37.730Z",
19      "msg": "testing2",
20      "u": {
21        "_id": "bRtgdhzM6PD9F8pSx",
22        "username": "testing2"
23      },
24      "groupable": false,
25      "_updatedAt": "2016-12-08T16:03:25.235Z"
26    },
27    {
28      "_id": "bfRW658nEyEBg75rc",
29      "t": "uj",
30      "rid": "ByehQjC44FwMeiLbX",
31      "ts": "2016-12-07T15:47:49.099Z",
32      "msg": "testing",
33      "u": {
```

```
34         "_id": "nSYqWzz4GsKTX4dyK",
35         "username": "testing1"
36     },
37     "groupable": false,
38     "_updatedAt": "2016-12-07T15:47:49.099Z"
39   },
40   {
41     "_id": "pbuFiGadhRZTKouhB",
42     "t": "uj",
43     "rid": "ByehQjC44FwMeiLbX",
44     "ts": "2016-12-06T17:57:38.635Z",
45     "msg": "testing",
46     "u": {
47       "_id": "y65tAmHs93aDChMWu",
48       "username": "testing"
49     },
50     "groupable": false,
51     "_updatedAt": "2016-12-06T17:57:38.635Z"
52   }
53 ],
54   "success": true
55 }
```

Change Log

Version	Description
0.75.0	Added <code>offset</code> property
0.47.0	Added

Channel Info

Retrieves the information about the channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.info	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required (if no roomName)	The channel's id
roomName	general	Required (if no roomId)	The channel's name

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/channels.info?roomId=ByehQjC44FwMeiLbX
```

Example Result

```
1 {
2   "channel": {
3     "_id": "ByehQjC44FwMeiLbX",
4     "name": "testing",
```

```
5      "fname": "testing",
6      "t": "c",
7      "msgs": 0,
8      "usersCount": 2,
9      "u": {
10         "_id": "HKKPmF8rZh45GMHWH",
11         "username": "marcos.defendi"
12     },
13     "customFields": {},
14     "broadcast": false,
15     "encrypted": false,
16     "ts": "2020-05-21T13:14:07.070Z",
17     "ro": false,
18     "default": false,
19     "sysMes": true,
20     "_updatedAt": "2020-05-21T13:14:07.096Z"
21   },
22   "success": true
23 }
```

Change Log

Version	Description
0.48.0	Added

Channel Invite

Adds a user to the channel.

Required Permissions

For a user to be able to invite others, they must match at least one of the premises below.

- If the user is part of a room of any type and has the `add-user-to-joined-room` permission.
- If the user is part of a public room (`t: 'c'`) and has the `add-user-to-any-c-room` permission.
- If the user is part of a private room (`t: 'p'`) and has the `add-user-to-any-p-room` permission.

URL	Requires Auth	HTTP Method
<code>/api/v1/channels.invite</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>roomId</code>	<code>ByehQjC44FwMeiLbX</code>	Required	The channel's id
<code>userId</code>	<code>nSYqWzZ4GsKTX4dyK</code>	Required	The user id of who to invite

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
```

```
3      -H "Content-type: application/json" \
4      http://localhost:3000/api/v1/channels.invite \
5      -d '{ "roomId": "ByehQjC44FwMeiLbX", "userId": "nSYqWzZ4GsKTX4dyK" }'
```

Example Result

```
1  {
2    "channel": {
3      "_id": "ByehQjC44FwMeiLbX",
4      "ts": "2016-11-30T21:23:04.737Z",
5      "t": "c",
6      "name": "testing",
7      "usernames": [
8        "testing",
9        "testing1"
10     ],
11     "msgs": 1,
12     "_updatedAt": "2016-12-09T12:50:51.575Z",
13     "lm": "2016-12-09T12:50:51.555Z"
14   },
15   "success": true
16 }
```

Change Log

Version	Description
0.48.0	Added

Channel Join

Joins yourself to the channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.join	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
joinCode	1234	Optional	The join code of the room

(i) The join code isn't needed if the user has the permission
join-without-join-code

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.join \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "joinCode": "1234" }'
```

Example Result

```
1  {
2      "channel": {
3          "_id": "nxXKHF2o2nzKYtFMM",
4          "name": "test",
5          "fname": "test",
6          "t": "c",
7          "msgs": 8,
8          "usersCount": 2,
9          "u": {
10             "_id": "rocketchat.internal.admin.test",
11             "username": "rocketchat.internal.admin.test"
12         },
13         "customFields": {},
14         "broadcast": false,
15         "encrypted": false,
16         "ts": "2019-01-16T12:00:04.783Z",
17         "ro": false,
18         "sysMes": true,
19         "default": false,
20         "_updatedAt": "2019-01-16T12:06:30.426Z",
21         "joinCodeRequired": true
22     },
23     "success": true
24 }
```

Change Log

Version	Description
0.49.0	Added

Channel Kick

Removes a user from the channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.kick	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
userId	nSYqWzz4GsKTX4dyK	Required	The user id of who to kick

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.kick \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "userId": "nSYqWzz4GsKTX4dyK" }'
```

Example Result

```
1 {
2   "channel": {
```

```
3      "_id": "ByehQjC44FwMeiLbX",
4      "name": "invite-me",
5      "t": "c",
6      "usernames": [
7          "testing1"
8      ],
9      "msgs": 0,
10     "u": {
11         "_id": "aobEdbYhXfu5hkeqG",
12         "username": "testing1"
13     },
14     "ts": "2016-12-09T15:08:58.042Z",
15     "ro": false,
16     "sysMes": true,
17     "_updatedAt": "2016-12-09T15:22:40.656Z"
18 },
19     "success": true
20 }
```

Change Log

Version	Description
0.48.0	Added

Channel Leave

Causes the callee to be removed from the channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.leave	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.leave \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
2   "channel": {
3     "_id": "ByehQjC44FwMeiLbX",
4     "name": "invite-me",
5     "t": "c",
```

```
6      "usernames": [
7          "testing2"
8      ],
9      "msgs": 0,
10     "u": {
11         "_id": "aobEdbYhXfu5hkeqG",
12         "username": "testing1"
13     },
14     "ts": "2016-12-09T15:08:58.042Z",
15     "ro": false,
16     "sysMes": true,
17     "_updatedAt": "2016-12-09T15:22:40.656Z"
18 },
19     "success": true
20 }
```

Change Log

Version	Description
0.48.0	Added

Channel List Joined

Lists all of the channels the calling user has joined. It supports the [Offset, Count, and Sort Query Parameters](#) along with just the [Fields Query Parameter](#).

URL	Requires Auth	HTTP Method
/api/v1/channels.list.joined	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUFD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/channels.list.joined
```

Example Result

```
1 [
2     "channels": [
3         {
4             "_id": "ByehQjC44FwMeiLbX",
5             "name": "invite-me",
6             "t": "c",
7             "usernames": [
8                 "testing1"
9             ],
10            "msgs": 0,
11            "u": {
12                "_id": "aobEdbYhXfu5hkeqG",
13                "username": "testing1"
14            },
15            "ts": "2016-12-09T15:08:58.042Z",
16            "ro": false,
17            "sysMes": true,
```

```
18          "_updatedAt": "2016-12-09T15:22:40.656Z"
19      }
20  ],
21  "success": true
22 }
```

Query Example Call

This example shows a list of Direct Messages' Rooms filtered by "customFields.field1" ended with "5" using a regular expression.

```
1 curl -H "X-Auth-Token: 0KoJelLu8rYtbyc3c5YtTwxIE-UvT1FzWv9cdq1XPI1" \
2   -H "X-User-Id: hw5DThnhQmxDWhavu" \
3   http://localhost:3000/api/v1/channels.list.joined?query=%7B%20%22name%
```

Query Example Result

```
1  {
2    "channels": [
3      {
4        "_id": "GENERAL",
5        "ts": "2018-01-21T20:58:41.142Z",
6        "t": "c",
7        "name": "general",
8        "msgs": 1,
9        "default": true,
10       "_updatedAt": "2018-01-21T21:03:43.736Z",
11       "username": "user2"
12     }
13   ],
14   "offset": 0,
15   "count": 1,
16   "total": 1,
17   "success": true
18 }
```

Change Log

Version	Description
0.62.0	Add 'query' parameter support.
0.49.0	Count and offset query parameters supported.
0.48.0	Added

Channel List

Lists all of the channels on the server. It supports the [Offset](#), [Count](#), and [Sort Query Parameters](#) along with [Query and Fields Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/channels.list	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/channels.list
```

Example Result

```
1  [
2      "channels": [
3          {
4              "_id": "ByehQjC44FwMeiLbX",
5              "name": "test-test",
6              "t": "c",
7              "usernames": [
8                  "testing1"
9              ],
10             "msgs": 0,
11             "u": {
12                 "_id": "aobEdbYhXfu5hkeqG",
13                 "username": "testing1"
14             },
15             "ts": "2016-12-09T15:08:58.042Z",
16             "ro": false,
17             "sysMes": true,
```

```
18         "_updatedAt": "2016-12-09T15:22:40.656Z"
19     },
20     {
21         "_id": "t7qapfhZjANMRAi5w",
22         "name": "testing",
23         "t": "c",
24         "usernames": [
25             "testing2"
26         ],
27         "msgs": 0,
28         "u": {
29             "_id": "y65tAmHs93aDChMWu",
30             "username": "testing2"
31         },
32         "ts": "2016-12-01T15:08:58.042Z",
33         "ro": false,
34         "sysMes": true,
35         "_updatedAt": "2016-12-09T15:22:40.656Z"
36     }
37 ],
38 "offset": 0,
39 "count": 1,
40 "total": 1,
41 "success": true
42 }
```

Change Log

Version	Description
0.49.0	Count and offset query parameters supported.
0.48.0	Added

Channel Members List

Lists all channel users. It supports the [Offset, Count, and Sort Query Parameters](#).

-  The list of elements a user can use to sort the list is limited. The current sortable element is: `username`

URL	Requires Auth	HTTP Method
<code>/api/v1/channels.members</code>	yes	GET

Query Parameters

Argument	Example	Required	Description
<code>roomId</code>	<code>ByehQjC44FwMeiLbX</code>	Required (if no <code>roomName</code>)	The channel's id
<code>roomName</code>	<code>general</code>	Required (if no <code>roomId</code>)	The channel's name

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/channels.members?roomId=ByehQjC44FwMeiLb
```

Example Result

```
1  {
2      "members": [
3          {
4              "_id": "Loz7qh9ChSqHMPymx",
5              "username": "customField_apiuser.test.1529436896005",
6              "name": "customField_apiuser.test.1529436896005",
7              "status": "offline"
8          },
9          {
10             "_id": "Zc3Y3cRW7ZtS7Y8Hk",
11             "username": "customField_apiuser.test.1529436997563",
12             "name": "customField_apiuser.test.1529436997563",
13             "status": "offline"
14         }
15     ],
16     "count": 2,
17     "offset": 0,
18     "total": 35,
19     "success": true
20 }
```

Change Log

Version	Description
0.59.0	Added

Channel messages

Lists all of the specific channel messages on the server. It supports the [Offset, Count, and Sort Query Parameters](#) along with [Query and Fields Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/channels.messages	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required(if no <code>roomName</code>)	The channel's id
roomName	testChannel	Required(if no <code>roomId</code>)	The channel's name

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/channels.messages?roomId=ByehQjC44FwMeiL
```

Example Result

```
1 {
2     "messages": [
3         {
```

```
4         "_id": "xadufzmxzYQp4H9py",
5         "alias": "test",
6         "msg": "Example message",
7         "attachments": [],
8         "parseUrls": true,
9         "bot": {
10             "i": "MnQyfhWt5LqZotyfc"
11         },
12         "groupable": false,
13         "ts": "2018-10-05T01:10:47.524Z",
14         "u": {
15             "_id": "rocket.cat",
16             "username": "rocket.cat",
17             "name": "Rocket.Cat"
18         },
19         "rid": "GENERAL",
20         "_updatedAt": "2018-10-05T13:42:51.163Z",
21         "reactions": {
22             ":grin:": {
23                 "usernames": [
24                     "marcos.defendi"
25                 ]
26             }
27         },
28         "mentions": [],
29         "channels": [],
30         "starred": {
31             "_id": "KPkEYwKKBKZnEEPpt"
32         }
33     },
34     {
35         "_id": "id-1538701845987",
36         "rid": "GENERAL",
37         "msg": "This message was edited via API",
38         "alias": "Gruggy",
39         "emoji": ":smirk:",
40         "avatar": "http://res.guggy.com/logo_128.png",
41         "attachments": [
42             {
43                 "collapsed": false,
44                 "color": "#ff0000",
45                 "text": "Yay for gruggy!",
46                 "ts": "2016-12-09T16:53:06.761Z",
47                 "message_link": "https://google.com",
48                 "thumb_url": "http://res.guggy.com/logo_128.png",
49                 "author_name": "Bradley Hilton",
50                 "author_link": "https://rocket.chat/",
51                 "author_icon": "https://avatars.githubusercontent.com/",
52                 "title": "Attachment Example",
53                 "title_link": "https://youtube.com",
54                 "title_link_download": true,
```

```
55         "image_url": "http://res.guggy.com/logo_128.png",
56         "audio_url": "http://www.w3schools.com/tags/horse.mp3"
57         "video_url": "http://www.w3schools.com/tags/movie.mp4"
58     "fields": [
59         {
60             "short": true,
61             "title": "Test",
62             "value": "Testing out something or other"
63         },
64         {
65             "short": true,
66             "title": "Another Test",
67             "value": "[Link](https://google.com/) somethin"
68         }
69     ]
70 ],
71 "ts": "2018-10-05T01:10:45.994Z",
72 "u": {
73     "_id": "rocketchat.internal.admin.test",
74     "username": "rocketchat.internal.admin.test",
75     "name": "RocketChat Internal Admin Test"
76 },
77 "_updatedAt": "2018-10-05T01:10:47.064Z",
78 "editedBy": {
79     "_id": "rocketchat.internal.admin.test",
80     "username": "rocketchat.internal.admin.test"
81 },
82 "editedAt": "2018-10-05T01:10:46.384Z",
83 "reactions": {
84     ":smile:": {
85         "usernames": [
86             "rocketchat.internal.admin.test"
87         ]
88     },
89     ":squid:": {
90         "usernames": [
91             "rocketchat.internal.admin.test"
92         ]
93     },
94     ":bee:": {
95         "usernames": [
96             "rocketchat.internal.admin.test"
97         ]
98     },
99     ":ant:": {
100         "usernames": [
101             "rocketchat.internal.admin.test"
102         ]
103     }
104 },
105 }
```

```
106         "mentions": [],
107         "channels": [],
108         "urls": []
109     },
110 ],
111     "count": 2,
112     "offset": 0,
113     "total": 2,
114     "success": true
115 }
```

Change Log

Version	Description
0.59.0	Added

Channel Moderators List

Lists all channel moderators.

URL	Requires Auth	HTTP Method
/api/v1/channels.moderators	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required (if no roomName)	The channel's id
roomName	general	Required (if no roomId)	The channel's name

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/channels.moderators?roomId=ByehQjC44FwMe
```

Example Result

```
1 {
2   "moderators": [
3     {
4       "_id": "rocket.cat",
```

```
5           "username": "rocket.cat",
6           "name": Rocket.Cat
7       }
8   ],
9   "success": true
10 }
```

Change Log

Version	Description
0.70.0	Added

Channel Online

Lists all online users of a channel if the channel's id is provided, otherwise it gets all online users of all channels. It supports the [Query Parameters only](#).

URL	Requires Auth	HTTP Method
/api/v1/channels.online	yes	GET

Query Parameters

Argument	Example	Required	Description
query	{"_id": "5HmCfp0B7jp2uibTC"}	Optional	See Query Parameter

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/channels.online
```

Example Result

```
1 {
2   "online": [
3     {
4       "_id": "47cRd58HnWwpqxhaZ",
5       "username": "test"
```

```
6      },
7      {
8          "_id": "BsxzC22xQ43taWdff",
9          "username": "uniqueusername"
10     }
11 ],
12 "success": true
13 }
```

Query Example Call

This example shows how to filter using channel's id.

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCuFD_0YdwnNnunAJF8V47U3QHXSq" \
2      -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3      http://localhost:3000/api/v1/channels.online?query={"_id": "5HmCfpoB7jp2ui
```

Query Example Result

```
1 {
2     "online": [
3         {
4             "_id": "47cRd58HnWwpqxhaZ",
5             "username": "test"
6         }
7     ],
8     "success": true
9 }
```

Change Log

Version	Description
0.52.0	Added

Channel Open

Adds the channel back to the user's list of channels.

URL	Requires Auth	HTTP Method
/api/v1/channels.open	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.open \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.48.0	Added

Channel Remove Leader

Removes the role of Leader for a user in the current channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.removeLeader	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
userId	oCHkav5Zf6vmpu2W2	Required	The user's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2 -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3 -H "Content-type: application/json" \
4 http://localhost:3000/api/v1/channels.removeLeader \
5 -d '{"roomId": "ByehQjC44FwMeiLbX", "userId": "oCHkav5Zf6vmpu2W2"}'
```

Success Example Result

```
1 {
2     "success": true
```

```
3 }
```

Bad Request Example Result

If the user is not a leader, will return a `400 bad request` status.

```
1 {
2   "success": false,
3   "error": "User is not a leader [error-user-not-leader]",
4   "errorType": "error-user-not-leader"
5 }
```

Change Log

Version	Description
0.75.0	Added.

Channel Remove Moderator

Removes the role of moderator from a user in the current channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.removeModerator	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
userId	nSYqWzz4GsKTX4dyK	Required	The user's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.removeModerator \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "userId": "nSYqWzz4GsKTX4dyK" }'
```

Example Result

```
1 {
2   "success": true
```

```
3 }
```

Change Log

Version	Description
0.49.4	Added

Channel Remove Owner

Removes the role of owner from a user in the current channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.removeOwner	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
userId	nSYqWzz4GsKTX4dyK	Required	The user's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.removeOwner \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "userId": "nSYqWzz4GsKTX4dyK" }'
```

Example Result

```
1 {
2   "success": true
```

```
3 }
```

Change Log

Version	Description
0.49.4	Added

Channel Rename

Changes the name of the channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.rename	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
name	new-name	Required	The new name the channel will have, it cannot be the same as another or the existing name

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.rename \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "name": "new-name" }'
```

Example Result

```
1  {
2    "channel": {
3      "_id": "ByehQjC44FwMeiLbX",
4      "name": "new-name",
5      "t": "c",
6      "usernames": [
7        "testing1"
8      ],
9      "msgs": 4,
10     "u": {
11       "_id": "aobEdbYhXfu5hkeqG",
12       "username": "testing1"
13     },
14     "ts": "2016-12-09T15:08:58.042Z",
15     "ro": false,
16     "sysMes": true,
17     "_updatedAt": "2016-12-09T15:57:44.686Z"
18   },
19   "success": true
20 }
```

Change Log

Version	Description
0.48.0	Added

Channel Roles

Lists all user's roles in the channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.roles	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required(if no roomName)	The channel's id
roomName	general	Required(if no roomId)	The channel's name

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/channels.roles?roomId=ByehQjC44FwMeiLbX
```

Example Result

```
1 {
2   "roles": [
3     {
4       "rid": "BaE62jfDLXK3Xo6BA",
```

```
5         "u": {
6             "_id": "BkNkw3iKgNyhMbPyW",
7             "username": "ronnie.dio",
8             "name": "Ronnie James Dio"
9         },
10        "roles": [
11            "moderator"
12        ],
13        "_id": "ehPuGyZBedznJsQHp"
14    }
15 ],
16 "success": true
17 }
```

Change Log

Version	Description
0.65.0	Added

Channel Set Announcement

Sets the announcement for the channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.setAnnouncement	yes	POST

Payload

Argument	Example	Required	Announcement
roomId	ByehQjC44FwMeiLbx	Required	The channel's id
announcement	Test out everything.	Required	The announcement to set for the channel.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.setAnnouncement \
5   -d '{ "roomId": "ByehQjC44FwMeiLbx", "announcement": "Test out everyt' 
```

Example Result

```
1 { 
```

```
2   "announcement": "Test out everything.",  
3   "success": true  
4 }
```

Change Log

Version	Description
0.63.0	Added

Channel Set Custom Fields

Sets the custom fields for the channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.setCustomFields	yes	POST

Payload

One of roomId or roomName is required.

Argument	Example	Required	Description
roomId	GENERAL	Required	The channel's id
roomName	general	Required	The channel's name
customFields	{"organization": "tra-la-la"}	Required	The custom fields to set for the channel.

Example Call

```
1 curl -H "X-Auth-Token: 8h2mKAwxB3AQrFSjLVKMooJyjdCFaA7W45sWlHP8Iz0" \
2   -H "X-User-Id: ew28FnZqipDpvKw3R" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.setCustomFields \
5   -d '{"roomId": "GENERAL", "customFields":{"organization": "tra-la-la"}}
```

Example Result

```
1  {
2    "channel": {
3      "_id": "GENERAL",
4      "ts": "2018-02-18T19:41:02.875Z",
5      "t": "c",
6      "name": "general",
7      "msgs": 236,
8      "default": true,
9      "_updatedAt": "2018-02-23T11:09:05.973Z",
10     "lm": "2018-02-18T22:18:20.288Z",
11     "customFields": {
12       "organization": "tra-la-la"
13     }
14   },
15   "success": true,
16   "developerWarning": "[WARNING]: The \"usernames\" field has been removed"
17 }
```

Change Log

Version	Description
0.62.0	Added

Channel Set Default

Sets whether the channel is a default channel or not.

URL	Requires Auth	HTTP Method
/api/v1/channels.setDefault	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
default	true	Required	Boolean of whether the room is default or not.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.setDefault \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "default": true }'
```

Example Result

```
1 {
```

```
2     "channel": {
3         "_id": "ByehQjC44FwMeiLbX",
4         "name": "testing0",
5         "t": "c",
6         "msgs": 0,
7         "u": {
8             "_id": "aiPqNoGkjpNDiRx6d",
9             "username": "goose160"
10        },
11        "ts": "2017-01-05T18:02:50.754Z",
12        "ro": true,
13        "sysMes": true,
14        "_updatedAt": "2017-01-05T19:02:24.429Z",
15        "usernames": [
16            "goose160",
17            "graywolf336"
18        ],
19        "joinCodeRequired": true,
20        "muted": [],
21        "default": true
22    },
23    "success": true
24 }
```

Change Log

Version	Description
0.66.0	Added

Channel Set Description

Sets the description for the channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.setDescription	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
description	Test out everything.	Required	The description to set for the channel.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.setDescription \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "description": "Test out everyth' 
```

Example Result

```
1 { 
```

```
2   "description": "Testing out everything.",  
3   "success": true  
4 }
```

Change Log

Version	Description
0.48.0	Added

Channel Set Join Code

Sets the code required to join the channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.setJoinCode	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
joinCode	my-join-code	Required	The join code which this channel will require to be joined.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.setJoinCode \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "joinCode": "my-join-code" }'
```

Example Result

```
1 {
```

```
2     "channel": {
3         "_id": "ByehQjC44FwMeiLbX",
4         "name": "testing0",
5         "t": "c",
6         "msgs": 0,
7         "u": {
8             "_id": "aiPqNoGkjpNDiRx6d",
9             "username": "goose160"
10        },
11        "ts": "2017-01-05T18:02:50.754Z",
12        "ro": false,
13        "sysMes": true,
14        "_updatedAt": "2017-01-05T18:41:48.840Z",
15        "usernames": [
16            "goose160",
17            "graywolf336"
18        ],
19        "joinCodeRequired": true
20    },
21    "success": true
22 }
```

Change Log

Version	Description
0.49.0	Added

Channel Set Purpose

Sets the description for the channel (the same as channels.setDescription, obsolete).

URL	Requires Auth	HTTP Method
/api/v1/channels.setPurpose	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
purpose	Test out everything.	Required	The description to set for the channel.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.setPurpose \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "purpose": "Test out everything"}
```

Example Result

```
1 {
2   "purpose": "Testing out everything.",
```

```
3      "success": true
4  }
```

Change Log

Version	Description
0.48.0	Added

Channel Set Read Only

Sets whether the channel is read only or not.

URL	Requires Auth	HTTP Method
/api/v1/channels.setReadOnly	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
readOnly	true	Required	Boolean of whether the room is read only or not.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.setReadOnly \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "readOnly": true }'
```

Example Result

```
1 {
```

```
2     "channel": {
3         "_id": "ByehQjC44FwMeiLbX",
4         "name": "testing0",
5         "t": "c",
6         "msgs": 0,
7         "u": {
8             "_id": "aiPqNoGkjpNDiRx6d",
9             "username": "goose160"
10        },
11        "ts": "2017-01-05T18:02:50.754Z",
12        "ro": true,
13        "sysMes": true,
14        "_updatedAt": "2017-01-05T19:02:24.429Z",
15        "usernames": [
16            "goose160",
17            "graywolf336"
18        ],
19        "joinCodeRequired": true,
20        "muted": []
21    },
22    "success": true
23 }
```

Change Log

Version	Description
0.49.0	Added

Channel Set Topic

Sets the topic for the channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.setTopic	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
topic	Discuss all of the testing.	Required	The channel's topic to set.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.setTopic \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "topic": "Discuss all of the tes'
```

Example Result

```
1 {
2   "topic": "Testing out everything.",
```

```
3      "success": true
4  }
```

Change Log

Version	Description
0.13.0	Added

Channel Set Type

Sets the type of room this channel should be.

URL	Requires Auth	HTTP Method
/api/v1/channels.setType	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
roomName	chat-room-name	Required	The channel's name
type	c or p	Required	The type of room this channel should be, either c or p . c is for channel and p is for private.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.setType \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "type": "p" }'
6
7 or
8
9 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
10  -H "X-User-Id: aobEdbYhXfu5hkeqG" \
11  -H "Content-type: application/json" \
12  http://localhost:3000/api/v1/channels.setType \
```

```
13      -d '{ "roomName": "chat-room-name", "type": "p" }'
```

Example Result

```
1  {
2      "channel": {
3          "_id": "ByehQjC44FwMeiLbX",
4          "name": "testing0",
5          "t": "p",
6          "msgs": 0,
7          "u": {
8              "_id": "aiPqNoGkjpNDiRx6d",
9              "username": "goose160"
10         },
11         "ts": "2017-01-05T18:02:50.754Z",
12         "ro": false,
13         "sysMes": true,
14         "_updatedAt": "2017-01-05T19:02:24.429Z",
15         "usernames": [
16             "goose160",
17             "graywolf336"
18         ],
19         "joinCodeRequired": true,
20         "muted": []
21     },
22     "success": true
23 }
```

Change Log

Version	Description
0.49.0	Added

Channel Unarchive

Unarchives a channel.

URL	Requires Auth	HTTP Method
/api/v1/channels.unarchive	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.unarchive \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.48.0	Added

Get all the mentions of a channel

It supports the [Offset, Count, and Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/channels.getAllUserMentionsByChannel	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/channels.getAllUserMentionsByChannel?roo
```

Example Result

```
1 {
2   "mentions": [
3     {
4       "_id": "Gptx3mc6TjSv5tLwb",
5       "rid": "GENERAL",
6       "msg": "@rocket.cat",
7       "ts": "2018-03-12T14:59:14.166Z",
```

```

8         "u": {
9             "_id": "47cRd58HnWwpqxhaZ",
10            "username": "test",
11            "name": "test"
12        },
13        "mentions": [
14            {
15                "_id": "47cRd58HnWwpqxhaZ",
16                "username": "rocket.cat"
17            }
18        ],
19        "channels": [],
20        "_updatedAt": "2018-03-12T14:59:14.171Z"
21    },
22    {
23        "_id": "rwerwfjuii6TjSv5tLWb",
24        "rid": "GENERAL",
25        "msg": "@rocket.cat",
26        "ts": "2018-03-12T14:59:14.166Z",
27        "u": {
28            "_id": "47cRd58HnWwpqxhaZ",
29            "username": "test",
30            "name": "test"
31        },
32        "mentions": [
33            {
34                "_id": "47cRd58HnWwpqxhaZ",
35                "username": "rocket.cat"
36            }
37        ],
38        "channels": [],
39        "_updatedAt": "2018-03-12T14:59:14.171Z"
40    }
41 ],
42 "count": 2,
43 "offset": 10,
44 "total": 2,
45 "success": true
46 }

```

Change Log

Version

Description

0.63.0

Added

Get Integrations

Retrieves the integrations which the channel has, requires at least one integration permission: `manage-incoming-integrations` , `manage-own-incoming-integrations` , `manage-outgoing-integrations` or `manage-own-outgoing-integrations` . It will return the integrations based on the user permission. It supports the [Offset, Count, and Sort Query Parameters](#) along with [Query and Fields Query Parameters](#).

URL	Requires Auth	HTTP Method
<code>/api/v1/channels.getIntegrations</code>	yes	GET

Query Parameters

Argument	Example	Required	Description
<code>roomId</code>	<code>ByehQjC44FwMeiLbX</code>	Required	The channel's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/channels.getIntegrations?roomId=ByehQjC4
```

Example Result

```
1 {
```

```

2     "integrations": [{  

3         "_id": "WMQDChpnYTRmFre9h",  

4         "enabled": true,  

5         "username": "rocket.cat",  

6         "alias": "Guggy",  

7         "avatar": "http://res.guggy.com/logo_128.png",  

8         "name": "Guggy",  

9         "triggerWords": [  

10            "!guggy",  

11            "guggy",  

12            "gif+"  

13        ],  

14        "urls": [  

15            "http://text2gif.guggy.com/guggify"  

16        ],  

17        "token": "8DFS89DMKLWEN",  

18        "script": "const config = {\n            color: '#ffffff'\n};\n\nclass Scr",  

19        "scriptEnabled": true,  

20        "impersonateUser": false,  

21        "scriptCompiled": "function _classCallCheck(instance,Constructor){",  

22        "scriptError": null,  

23        "type": "webhook-outgoing",  

24        "userId": "rocket.cat",  

25        "channel": [],  

26        "_createdAt": "2017-01-05T17:06:05.660Z",  

27        "_createdBy": {  

28            "username": "graywolf336",  

29            "_id": "R4jgcQaQhvK6K3iY"  

30        },  

31        "_updatedAt": "2017-01-05T17:06:05.660Z"  

32    }],  

33    "success": true  

34 }

```

Change Log

Version	Description
1.1.0	Separate permissions in <code>incoming</code> and <code>outgoing</code> .
0.49.0	Added

Chat

Url	Short Description	Details Page
/api/v1/chat.delete	Deletes an existing chat message.	Link
/api/v1/chat.followMessage	Follows an existing chat message.	Link
/api/v1/chat.getDeletedMessages	Retrieves the deleted messages since specific date.	Link
/api/v1/chat.getDiscussions	Retrieves the discussions of a room.	Link
/api/v1/chat.getMentionedMessages	Retrieves the mentioned messages.	Link
/api/v1/chat.getMessage	Retrieves a single chat message.	Link
/api/v1/chat.getMessageReadReceipts	Retrieves message read receipts.	Link
/api/v1/chat.getPinnedMessages	Retrieve pinned messages from a room.	Link
/api/v1/chat.getSnippedMessages	Retrieves snipped messages.	Link
/api/v1/chat.getSnippedMessageById	Retrieves snipped message by id.	Link
/api/v1/chat.getStarredMessages	Retrieves the starred messages.	Link
/api/v1/chat.getThreadMessages	Retrieves thread's messages.	Link
/api/v1/chat.getThreadsList	Retrieves channel's threads.	Link
/api/v1/chat.ignoreUser	Ignores an user from a chat.	Link
/api/v1/chat.pinMessage	Pins a chat message to the message's channel.	Link

/api/v1/chat.postMessage	Posts a new chat message.	Link
/api/v1/chat.react	Sets/unsets the user's reaction to an existing chat message.	Link
/api/v1/chat.reportMessage	Reports a message.	Link
/api/v1/chat.search	Search for messages in a channel.	Link
/api/v1/chat.starMessage	Stars a chat message for the authenticated user.	Link
/api/v1/chat.sendMessage	Send new chat message.	Link
/api/v1/chat.syncThreadMessages	Retrieves synced thread's messages.	Link
/api/v1/chat.syncThreadsList	Retrieves thread's synced channel threads.	Link
/api/v1/chat.unfollowMessage	Unfollows an existing chat message.	Link
/api/v1/chat.unPinMessage	Removes the pinned status of the provided chat message.	Link
/api/v1/chat.unStarMessage	Removes the star on the chat message for the authenticated user.	Link
/api/v1/chat.update	Updates the text of the chat message.	Link

Delete

URL	Requires Auth	HTTP Method
/api/v1/chat.delete	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The room id of where the message is to delete.
msgId	7aDSXtjMA3KPLxLjt	Required	The message id to delete.
asUser	true	Optional Default: false	Whether the message should be deleted as the user who sent it.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/chat.delete \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "msgId": "7aDSXtjMA3KPLxLjt", "a
```

Example Result

```
1  {
2    "_id": "7aDSXtjMA3KPLxLjt",
3    "ts": 1481741940895,
4    "success": true
5 }
```

Change Log

Version	Description
0.48.0	Added

React

Toggles the authenticated user's reaction to the provided message.

URL	Requires Auth	HTTP Method
/api/v1/chat.react	yes	POST

Payload

Argument	Example	Required	Description
emoji	smile	Required	The emoji to react with.
messageId	7aDSXtjMA3KPLxLjt	Required	The message id to react to.
shouldReact	true	Optional: true or false	Parameter to specify if the reaction should be added or removed.

The `emoji` does not have to contain the `:`'s. However, the emoji *does* have to exist.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/chat.react \
5   -d '{ "messageId": "7aDSXtjMA3KPLxLjt", "emoji": "smile", "shouldReac
```

Example Result

```
1  {
2      "success": true
3 }
```

Change Log

Version	Description
0.64.0	Added parameter "shouldReact" to make endpoint work like a setter
0.63.0	Emoji has to exist now
0.62.2	Added

Update

URL	Requires Auth	HTTP Method
/api/v1/chat.update	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The room id of where the message is.
msgId	7aDSxtjMA3KPLxLjt	Required	The message id to update.
text	Updated text	Required	Updated text for the message.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2 -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3 -H "Content-type:application/json" \
4 http://localhost:3000/api/v1/chat.update \
5 -d '{ "roomId": "ByehQjC44FwMeiLbX", "msgId": "7aDSxtjMA3KPLxLjt", "t'
```

Example Result

```
1 {
2     "message": {
```

```
3      "_id": "qGdhTGDnhMLJPQYY8",
4      "rid": "GENERAL",
5      "msg": "gif+ testing update",
6      "ts": "2017-01-05T17:06:14.403Z",
7      "u": {
8          "_id": "R4jgcQaQhvK6K3iY",
9          "username": "graywolf336"
10     },
11     "_updatedAt": "2017-01-05T19:42:20.433Z",
12     "editedAt": "2017-01-05T19:42:20.431Z",
13     "editedBy": {
14         "_id": "R4jgcQaQhvK6K3iY",
15         "username": "graywolf336"
16     }
17   },
18   "success": true
19 }
```

Change Log

Version	Description
0.49.0	Added

Report Message

Reports a message.

URL	Requires Auth	HTTP Method
/api/v1/chat.reportMessage	yes	POST

Payload

Argument	Example	Required	Description
messageId	7aDSXtjMA3KPLxLjt	Required	The message id to report.
description	test	Required	The description of report.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/chat.reportMessage \
5   -d '{ "messageId": "7aDSXtjMA3KPLxLjt", "description": "test" }'
```

Example Result

```
1 {
2     "success": true
```

```
3 }
```

Change Log

Version	Description
0.64.0	Added

Follows Message

Follows a chat message to the message's channel.

URL	Requires Auth	HTTP Method
/api/v1/chat.followMessage	yes	POST

Payload

Argument	Example	Required	Description
mid	7aDSXtjMA3KPLxLjt	Required	The message id to follow.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: Z3cpIYN6CNK2oXWKv" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/chat.followMessage \
5   -d '{ "mid": "7aDSXtjMA3KPLxLjt" }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
1.0.0	Added

Get Message

Retrieves a single chat message by the provided id. Callee must have permission to access the room where the message resides.

URL	Requires Auth	HTTP Method
/api/v1/chat.getMessage	yes	GET

Payload

Argument	Example	Required	Description
msgId	7aDSXtjMA3KPLxLjt	Required	The id of the message to get.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/chat.getMessage?msgId=7aDSXtjMA3KPLxLjt
```

Example Result

```
1 {
2   "message": {
3     "_id": "7aDSXtjMA3KPLxLjt",
4     "rid": "GENERAL",
```

```
5      "msg": "This is a test!",
6      "ts": "2016-12-14T20:56:05.117Z",
7      "u": {
8          "_id": "y65tAmHs93aDChMWu",
9          "username": "graywolf336"
10     }
11 },
12 "success": true
13 }
```

Change Log

Version	Description
0.47.0	Added

Get Channel Threads

Get channel threads. It supports the [Offset, Count, and Sort Query Parameters](#) along with [Query and Fields Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/chat.getThreadsList	yes	GET

Payload

Argument	Example	Required	Description
rid	7aDSXtjMA3KPLxLjt	Required	The id of the channel.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/chat.getThreadsList?rid=GENERAL
```

Example Result

```
1 {
2   "threads": [
3     {
4       "_id": "gcGai9bRREqokjyPc",
```

```

5           "rid": "GENERAL",
6           "msg": "test",
7           "ts": "2019-04-08T13:15:52.017Z",
8           "u": {
9               "_id": "p4a8YxvLQEhmibKTS",
10              "username": "marcos.defendi",
11              "name": "Marcos Defendi"
12          },
13          "_updatedAt": "2019-04-08T14:40:27.789Z",
14          "mentions": [],
15          "channels": [],
16          "replies": [
17              "p4a8YxvLQEhmibKTS"
18          ],
19          "tcount": 5,
20          "tlm": "2019-04-08T14:40:27.724Z"
21      },
22      {
23          "_id": "5iJjPNPhxMeLtEJQt",
24          "rid": "GENERAL",
25          "msg": "Sample message",
26          "ts": "2019-04-07T20:41:15.965Z",
27          "u": {
28              "_id": "rocketchat.internal.admin.test",
29              "username": "rocketchat.internal.admin.test",
30              "name": "RocketChat Internal Admin Test"
31          },
32          "_updatedAt": "2019-04-08T13:41:55.205Z",
33          "mentions": [],
34          "channels": [],
35          "replies": [
36              "rocketchat.internal.admin.test",
37              "p4a8YxvLQEhmibKTS"
38          ],
39          "tcount": 1,
40          "tlm": "2019-04-08T13:41:55.102Z"
41      }
42  ],
43  "count": 2,
44  "offset": 0,
45  "total": 2,
46  "success": true
47 }

```

Change Log

Version	Description
1.0.0	Added

Get Deleted Messages

Retrieve message read receipts. It supports ONLY the [Offset and Count Query Parameters](#)

URL	Requires Auth	HTTP Method
/api/v1/chat.getDeletedMessages	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The room id
since	2018-12-12T00:11:22.345Z	Required	The date (ISODate) to search deleted messages

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/chat.getDeletedMessages?roomId=ByehQjC44
```

Example Result

```
1 {
2   "messages": [
3     {
```

```
4         "_id": "W32fr3LjPxQ2TQH8A"
5     },
6     {
7         "_id": "D43CH4HC2J43C2JB4"
8     }
9 ],
10 "count": 2,
11 "offset": 2,
12 "total": 91,
13 "success": true
14 }
```

Change Log

Version	Description
0.73.0	Added

Get Discussions of A Room

Retrieve the discussions of a room. It supports the [Offset, Count, and Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/chat.getDiscussions	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The room id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/chat.getDiscussions?roomId=ByehQjC44FwMe
```

Example Result

```
1 {
2   "messages": [
3     {
4       "_id": "WSxSpajNjbNZDkGmq",
5       "t": "discussion-created",
```

```
6      "rid": "GENERAL",
7      "ts": "2019-12-03T17:53:44.600Z",
8      "msg": "d",
9      "u": {
10        "_id": "CaevAPDbXN252kyXa",
11        "username": "marcos.defendi"
12      },
13      "groupable": false,
14      "drid": "iQyCSjH2kixh46R7v",
15      "attachments": [],
16      "_updatedAt": "2019-12-03T17:53:44.600Z"
17    },
18  ],
19  "count": 1,
20  "offset": 0,
21  "total": 4,
22  "success": true
23 }
```

Change Log

Version	Description
2.4.0	Added

Get Mentioned Messages

Retrieve mentioned messages. It supports the [Offset, Count, and Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/chat.getMentionedMessages	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The room id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/chat.getMentionedMessages?roomId=ByehQjC
```

Example Result

```
1 {
2   "messages": [
3     {
4       "_id": "pN5jb4c9fhsDt8rfY",
5       "rid": "GENERAL",
6       "msg": "@rocketchat.internal.admin.test",
7       "ts": "2019-10-08T17:53:18.112Z",
```

```
8         "u": {
9             "_id": "TwHg3x3i6gGHuyFjX",
10            "username": "a",
11            "name": "a"
12        },
13        "_updatedAt": "2019-10-08T17:53:18.184Z",
14        "mentions": [
15            {
16                "_id": "rocketchat.internal.admin.test",
17                "name": "RocketChat Internal Admin Test",
18                "username": "rocketchat.internal.admin.test"
19            }
20        ],
21        "channels": []
22    },
23],
24 "count": 2,
25 "offset": 0,
26 "total": 2,
27 "success": true
28 }
```

Change Log

Version	Description
2.2.0	Added

Get Message Read Receipts

Retrieve message read receipts.

URL	Requires Auth	HTTP Method
/api/v1/chat.getMessageReadReceipts	yes	GET

Query Parameters

Argument	Example	Required	Description
messageId	ByehQjC44FwMeiLbX	Required	The message id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/chat.getMessageReadReceipts?messageId=By
```

Example Result

```
1 {
2   "receipts": [
3     {
4       "_id": "HksCYdTpCiM9DZ7Sa",
5       "roomId": "GENERAL",
6       "userId": "nvw6PBrXTejp4sfQt",
7       " messageId": "WyDsZzjk2wHogtWK2",
```

```
8          "ts": "2018-02-26T20:34:03.907Z",
9          "user": {
10            "username": "rocket.cat",
11            "name": "Rocket cat",
12            "_id": "nw6PBrXTejp4sfQt"
13          }
14        }
15      ],
16      "success": true
17    }
```

Change Log

Version	Description
0.63.0	Added

Get Pinned Messages

Retrieve pinned messages from a room. It supports ONLY the [Offset and Count Query Parameters](#)

URL	Requires Auth	HTTP Method
/api/v1/chat.getPinnedMessages	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The room id
offset	0	Optional	See Offset Parameter
count	1	Optional	See Count Parameter

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/chat.getPinnedMessages?roomId=ByehQjC44Fw
```

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/chat.getPinnedMessages?roomId=ByehQjC44Fw
```

Example Result

```
1  {
2      "messages": [
3          {
4              "_id": "SrhcA3mgthgjkEisJ",
5              "rid": "ByehQjC44FwMeiLbX",
6              "msg": "I pinned this message",
7              "ts": "2019-03-23T00:53:24.388Z",
8              "u": {
9                  "_id": "aobEdbYhXfu5hkeqG",
10                 "username": "user",
11                 "name": "User"
12             },
13             "mentions": [],
14             "channels": [],
15             "_updatedAt": "2019-03-23T00:53:28.813Z",
16             "pinned": true,
17             "pinnedAt": "2019-03-23T00:53:28.813Z",
18             "pinnedBy": {
19                 "_id": "aobEdbYhXfu5hkeqG",
20                 "username": "user"
21             }
22         },
23         {
24             "_id": "m3AZcKrvayKEZSKJN",
25             "rid": "GENERAL",
26             "msg": "Ola",
27             "ts": "2019-03-23T00:53:50.974Z",
28             "u": {
29                 "_id": "aobEdbYhXfu5hkeqG",
30                 "username": "user",
31                 "name": "user"
32             },
33             "mentions": [],
34             "channels": [],
35             "_updatedAt": "2019-03-23T00:53:53.649Z",
36             "pinned": true,
37             "pinnedAt": "2019-03-23T00:53:53.649Z",
38             "pinnedBy": {
39                 "_id": "aobEdbYhXfu5hkeqG",
40                 "username": "user"
41             }
42         }
43     ],
44     "count": 2,
45     "offset": 0,
46     "total": 2,
```

```
47      "success": true
48  }
```

Change Log

Version	Description
2.0.0	Added

Get Snippeted Message By Id

Retrieve snippeted message by Id.

URL	Requires Auth	HTTP Method
/api/v1/chat.getSnippedMessageById	yes	GET

Query Parameters

Argument	Example	Required	Description
messageId	ByehQjC44FwMeiLbX	Required	The message id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/chat.getSnippedMessageById?messageId=B
```

Example Result

```
1 {
2   "message": {
3     "_id": "48sifqi7ezh4ifKaK",
4     "rid": "GENERAL",
5     "msg": "```dsfsdfsdf``",
6     "ts": "2019-10-25T23:26:31.550Z",
7     "u": {
```

```
8         "_id": "CaevAPDbXN252kyXa",
9         "username": "marcos.defendi",
10        "name": "Marcos Defendi"
11    },
12    "_updatedAt": "2019-10-25T23:27:51.300Z",
13    "mentions": [],
14    "channels": [],
15    "snippetName": "rr",
16    "snipped": true,
17    "snippedBy": {
18        "_id": "CaevAPDbXN252kyXa",
19        "username": "marcos.defendi"
20    }
21},
22 "success": true
23 }
```

Change Log

Version	Description
2.3.0	Added

Get Snippeted Messages

Retrieve snippeted messages. It supports the [Offset, Count, and Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/chat.getSnippedMessages	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The room id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/chat.getSnippedMessages?roomId=ByehQjC
```

Example Result

```
1 {
2   "messages": [
3     {
4       "_id": "QQuPFFcLmsqHWeqhn",
5       "rid": "GENERAL",
6       "msg": "```\natest\n```",
7       "ts": "2019-10-25T21:42:39.022Z",
```

```
8         "u": {
9             "_id": "CaevAPDbXN252kyXa",
10            "username": "marcos.defendi",
11            "name": "Marcos Defendi"
12        },
13        "_updatedAt": "2019-10-25T22:54:02.094Z",
14        "mentions": [],
15        "channels": [],
16        "snippetName": "ee",
17        "snipped": true,
18        "snippedBy": {
19            "_id": "CaevAPDbXN252kyXa",
20            "username": "marcos.defendi"
21        }
22    }
23 ],
24 "count": 1,
25 "offset": 0,
26 "total": 1,
27 "success": true
28 }
```

Change Log

Version	Description
2.3.0	Added

Get Starred Messages

Retrieve starred messages. It supports the [Offset, Count, and Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/chat.getStarredMessages	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The room id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/chat.getStarredMessages?roomId=ByehQjC44
```

Example Result

```
1 {
2   "messages": [
3     {
4       "_id": "ePeoiJQ3Qfc4556TW",
5       "rid": "GENERAL",
6       "msg": "fsdfs",
7       "ts": "2019-10-09T19:17:21.883Z",
```

```
8         "u": {
9             "_id": "rocketchat.internal.admin.test",
10            "username": "rocketchat.internal.admin.test",
11            "name": "RocketChat Internal Admin Test"
12        },
13        "_updatedAt": "2019-10-09T19:17:25.196Z",
14        "mentions": [],
15        "channels": [],
16        "starred": [
17            {
18                "_id": "rocketchat.internal.admin.test"
19            }
20        ]
21    },
22    "count": 2,
23    "offset": 0,
24    "total": 2,
25    "success": true
26 }
27 }
```

Change Log

Version	Description
2.2.0	Added

Get Thread Messages

Get thread messages. It supports the [Offset, Count, and Sort Query Parameters](#) along with [Query and Fields Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/chat.getThreadMessages	yes	GET

Payload

Argument	Example	Required	Description
tmid	7aDSXtjMA3KPLxLjt	Required	The id of the the thread message.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/chat.getThreadMessages?tmid=7aDSXtjMA3KP
```

Example Result

```
1 {
2   "messages": [
3     {
4       "_id": "gcGai9bRREqokjyPc",
```

```

5          "rid": "GENERAL",
6          "msg": "Test",
7          "ts": "2019-04-08T13:15:52.017Z",
8          "u": {
9              "_id": "p4a8YxvLQEhmibKTS",
10             "username": "marcos.defendi",
11             "name": "Marcos Defendi"
12         },
13         "_updatedAt": "2019-04-08T14:40:27.789Z",
14         "mentions": [],
15         "channels": [],
16         "replies": [
17             "p4a8YxvLQEhmibKTS"
18         ],
19         "tcount": 5,
20         "tlm": "2019-04-08T14:40:27.724Z"
21     },
22     {
23         "_id": "GfhiijjcjKFyYMuMY",
24         "rid": "GENERAL",
25         "tmid": "gcGai9bRREqokjyPc",
26         "msg": "This is a test!",
27         "ts": "2019-04-08T13:20:22.238Z",
28         "u": {
29             "_id": "p4a8YxvLQEhmibKTS",
30             "username": "marcos.defendi",
31             "name": "Marcos Defendi"
32         },
33         "_updatedAt": "2019-04-08T13:20:22.265Z",
34         "mentions": [],
35         "channels": []
36     }
37 ],
38 "count": 2,
39 "offset": 0,
40 "total": 2,
41 "success": true
42 }
```

Change Log

Version	Description
1.0.0	Added

Ignore User

Ignores an user in a chat. If you pass ignore as false, the user will be unignored.

URL	Requires Auth	HTTP Method
/api/v1/chat.ignoreUser	yes	GET

Payload

Argument	Example	Required	Description
rid	KLTM332QEwWLPJrjRPtcPHimPnLHsusidF	Required	The room ID.
userId	7aDSXtjMA3KPLxLjt	Required	The User ID.
ignore	true	Optional	If the user will be ignored or not, default is true.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/chat.ignoreUser?rid=KLTM332QEwWLPJrjRPtc
```

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/chat.ignoreUser?rid=KLTM332QEwWLPJrjRPtc
```

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2 -H "X-User-Id: aobEdbYhXfu5hkeqqG" \
3 -H "Content-type:application/json" \
4 http://localhost:3000/api/v1/chat.ignoreUser?rid=KLTM332QEwWLPJrjRPtc
```

Example Result

```
1 {
2     "success": true
3 }
```

Change Log

Version	Description
0.64.0	Added

Pins Message

Pins a chat message to the message's channel.

URL	Requires Auth	HTTP Method
/api/v1/chat.pinMessage	yes	POST

Payload

Argument	Example	Required	Description
messageId	7aDSXtjMA3KPLxLjt	Required	The message id to pin.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: Z3cpIYN6CNK2oXWKv" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/chat.pinMessage \
5   -d '{ "messageId": "7aDSXtjMA3KPLxLjt" }'
```

Example Result

```
1 {
2   "message": {
3     "t": "message_pinned",
4     "rid": "GENERAL",
5     "ts": "2017-09-27T20:39:57.921Z",
```

```
6      "msg": "",  
7      "u": {  
8          "_id": "Z3cpIYN6CNK2oXWKv",  
9          "username": "graywolf336"  
10     },  
11     "groupable": false,  
12     "attachments": [  
13         {  
14             "text": "Hello",  
15             "author_name": "graywolf336",  
16             "author_icon": "/avatar/graywolf336?_dc=0",  
17             "ts": "2017-09-27T19:36:01.683Z"  
18         }  
19     ],  
20     "_updatedAt": "2017-09-27T20:39:57.921Z",  
21     "_id": "hmzxXKSWmMkoQyiAd"  
22 },  
23     "success": true  
24 }
```

Change Log

Version	Description
0.59.0	Added

Post Message

IMPORTANT

You only can send `alias` and `avatar` properties if your user has the `bot` role. We implement this rule to avoid users to impersonate other users.

URL	Requires Auth	HTTP Method
<code>/api/v1/chat.postMessage</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>roomId</code>	<code>ByehQjC44FwMeiLbX</code>	Required	The room id of where the message is to be sent. The channel name with the prefix in front of it. # refers to channel, however @ refers to username
<code>channel</code>	<code>#general</code> OR <code>@eric</code>		
<code>text</code>	<code>Sample message</code>	Optional	The text of the message to send, is optional because of attachments.
<code>alias</code>	<code>Some Name</code>	Optional	This will cause the message's name to appear as the given alias, but your username will still display.
<code>emoji</code>	<code>:smirk:</code>	Optional	If provided, this will make the avatar on this message be an emoji. Emoji Cheetsheet

avatar	<code>http://site.com/logo.png</code>	Optional	If provided, this will make the avatar use the provided image url.
attachments	<code>[{}]</code>	Optional	See the below section, Attachments Detail , for details.

Attachments Detail

The attachments is an array of objects with any of the following properties. One attachment can have many sections, including:

- General
- Author Information
- Title Information
- Image
- Audio
- Video
- Table/Fields

Property	Example	Section	Description
color	<code>#ff0000</code>	General	The color you want the order on the left side to be, any value background-css supports.
text	<code>Sample attachment text</code>	General	The text to display for this attachment, it is different than the message's text.
ts	<code>2016-12-09T16:53:06.761Z</code>	General	Displays the time next to the <code>text</code> portion.

<code>thumb_url</code>	<code>https://site.com/img.png</code>	General	An image that displays to the left of the <code>text</code> , looks better when this is relatively small.
<code>message_link</code>	<code>https://rocket.chat</code>	General	Only applicable if the <code>ts</code> is provided, as it makes the time clickable to this link.
<code>collapsed</code>	<code>false</code>	General	Causes the image, audio, and video sections to be hiding when collapsed is true.
<code>author_name</code>	<code>Bradley Hilton</code>	Author	Name of the author.
<code>author_link</code>	<code>https://bit.ly/</code>	Author	Providing this makes the author name clickable and points to this link.
<code>author_icon</code>	<code>https://site.com/img.png</code>	Author	Displays a tiny icon to the left of the Author's name.
<code>title</code>	<code>Attachment Title</code>	Title	Title to display for this attachment, displays under the author.
<code>title_link</code>	<code>https://youtube.com</code>	Title	Providing this makes the title clickable, pointing to this link.

<code>title_link_download</code>	<code>true</code>	Title	When this is true, a download icon appears and clicking this saves the link to file.
<code>image_url</code>	<code>https://site.com/img.png</code>	Image	The image to display, will be "big" and easy to see.
<code>audio_url</code>	<code>https://site.com/aud.mp3</code>	Audio	Audio file to play, only supports what html audio does.
<code>video_url</code>	<code>https://site.com/vid.mp4</code>	Video	Video file to play, only supports what html video does.
<code>fields</code>	<code>[{}]</code>	Fields	An array of Attachment Field Objects .

Attachment Field Objects

The field property of the attachments allows for "tables" or "columns" to be displayed on messages.

Property	Example	Required	Description
<code>short</code>	<code>true</code>	Optional <code>false</code>	Default: Whether this field should be a short field.
<code>title</code>	<code>Status</code>	Required	The title of this field.
<code>value</code>	<code>online</code>	Required	The value of this field, displayed underneath the title value.

Message Object Example

```
1  {
2      "alias": "Gruggy",
3      "avatar": "http://res.guggy.com/logo_128.png",
4      "channel": "#general",
5      "emoji": ":smirk:",
6      "roomId": "Xnb2kLD2Pnhdwe3RH",
7      "text": "Sample message",
8      "attachments": [
9          {
10             "audio_url": "http://www.w3schools.com/tags/horse.mp3",
11             "author_icon": "https://avatars.githubusercontent.com/u/850391?v=3",
12             "author_link": "https://rocket.chat/",
13             "author_name": "Bradley Hilton",
14             "collapsed": false,
15             "color": "#ff0000",
16             "fields": [
17                 {
18                     "short": true,
19                     "title": "Test",
20                     "value": "Testing out something or other"
21                 },
22                 {
23                     "short": true,
24                     "title": "Another Test",
25                     "value": "[Link](https://google.com/) something and this and tha
26                 }
27             ],
28             "image_url": "http://res.guggy.com/logo_128.png",
29             "message_link": "https://google.com",
30             "text": "Yay for gruggy!",
31             "thumb_url": "http://res.guggy.com/logo_128.png",
32             "title": "Attachment Example",
33             "title_link": "https://youtube.com",
34             "title_link_download": true,
35             "ts": "2016-12-09T16:53:06.761Z",
36             "video_url": "http://www.w3schools.com/tags/movie.mp4"
37         }
38     ]
39 }
```

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/chat.postMessage \
5   -d '{ "channel": "#general", "text": "This is a test!" }'
```

Example Result

```
1  {
2    "ts": 1481748965123,
3    "channel": "general",
4    "message": {
5      "alias": "",
6      "msg": "This is a test!",
7      "parseUrls": true,
8      "groupable": false,
9      "ts": "2016-12-14T20:56:05.117Z",
10     "u": {
11       "_id": "y65tAmHs93aDChMWu",
12       "username": "graywolf336"
13     },
14     "rid": "GENERAL",
15     "_updatedAt": "2016-12-14T20:56:05.119Z",
16     "_id": "jC9chsFddTvsbFQG7"
17   },
18   "success": true
19 }
```

Change Log

Version	Description
2.4.0	Added validation on user's identity
0.49.0	The <code>channel</code> can now be a room's id.

-
- 0.48.0 Information about the sent message is now returned.
 - 0.17.0 Messages aren't always processed as from BOT and urls are parsed by default.
 - 0.14.0 Internally using `processWebhookMessage` which enabled more features.
 - 0.13.0 Added
-

Search

Search for messages in a channel by id and text message.

URL	Requires Auth	HTTP Method
/api/v1/chat.search	yes	GET

Payload

Argument	Example	Required	Description
roomId	7aDSXtjMA3KPLxLjt	Required	The id of the channel.
searchText	test to search	Required	The text message to search in messages.
count	10	Optional	The limit of result.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -H "Content-type:application/json" \
4     http://localhost:3000/api/v1/chat.search?roomId=GENERAL&searchText=te
```

Example Result

```
1  {
2      "messages": [
3          {
4              "_id": "px9KLW9G2Sfd5DKFt",
5              "rid": "GENERAL",
6              "msg": "this is a test",
7              "ts": "2018-03-27T14:44:00.549Z",
8              "u": {
9                  "_id": "RtMDEYc28fQ5aHpf4",
10                 "username": "marcos.defendi",
11                 "name": "Marcos Defendi"
12             },
13             "mentions": [],
14             "channels": [],
15             "_updatedAt": "2018-03-27T14:44:00.550Z",
16             "score": 0.5833333333333334
17         }
18     ],
19     "success": true
20 }
```

Change Log

Version	Description
0.67.0	Changed <code>limit</code> query parameter to <code>count</code>
0.60.0	Added

Send Message

The difference between `chat.postMessage` and `chat.sendMessage` is that `chat.sendMessage` allows for passing a value for `_id` and the other one doesn't. Also, `chat.sendMessage` only sends it to one channel whereas the other one allows for sending to more than one channel at a time.

IMPORTANT

You can only send `alias` and `avatar` properties if your user has the `message-impersonate` permission. We implemented this rule to avoid users impersonating other users. By default, only the `bot` role has this permission, but that can be changed in Administration -> Permissions -> `message-impersonate`.

URL	Requires Auth	HTTP Method
<code>/api/v1/chat.sendMessage</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>message._id</code>	<code>ByehQjC44FwMeiLbx</code>	Optional	The <code>_id</code> of message.
<code>message.rid</code>	<code>ByehQjC44FwMeiLbx</code>	Required	The room id of where the message is to be sent.
<code>message.tmid</code>	<code>ByehQCh2435MeiLbx</code>	Optional	The message's id to create a thread.

message.msg	Sample message	Optional	The text of the message to send, is optional because of attachments.
message.alias	Some Name	Optional	This will cause the message's name to appear as the given alias, but your username will still display.
message.emoji	:smirk:	Optional	If provided, this will make the avatar on this message be an emoji. Emoji Cheetsheet
message.avatar	http://site.com/logo.png	Optional	If provided, this will make the avatar use the provided image url.
message.attachments	[{}]	Optional	See the below section, Attachments Detail , for details.

Attachments Detail

The attachments is an array of objects with any of the following properties. One attachment can have many sections, including:

- General
- Author Information
- Title Information
- Image

- Audio
- Video
- Table/Fields

Property	Example	Section	Description
color	#ff0000	General	The color you want the order on the left side to be, any value background-css supports.
text	Sample attachment text	General	The text to display for this attachment, it is different than the message's text.
ts	2016-12-09T16:53:06.761Z	General	Displays the time next to the <code>text</code> portion.
thumb_url	https://site.com/img.png	General	An image that displays to the left of the <code>text</code> , looks better when this is relatively small.
message_link	https://rocket.chat	General	Only applicable if the <code>ts</code> is provided, as it makes the time clickable to this link.
collapsed	false	General	Causes the image, audio, and video sections to be hiding when collapsed is true.
author_name	Bradley Hilton	Author	Name of the author.

<code>author_link</code>	<code>https://bit.ly/</code>	Author	Providing this makes the author name clickable and points to this link.
<code>author_icon</code>	<code>https://site.com/img.png</code>	Author	Displays a tiny icon to the left of the Author's name.
<code>title</code>	<code>Attachment Title</code>	Title	Title to display for this attachment, displays under the author.
<code>title_link</code>	<code>https://youtube.com</code>	Title	Providing this makes the title clickable, pointing to this link.
<code>title_link_download</code>	<code>true</code>	Title	When this is true, a download icon appears and clicking this saves the link to file.
<code>image_url</code>	<code>https://site.com/img.png</code>	Image	The image to display, will be "big" and easy to see.
<code>audio_url</code>	<code>https://site.com/aud.mp3</code>	Audio	Audio file to play, only supports what html audio does.
<code>video_url</code>	<code>https://site.com/vid.mp4</code>	Video	Video file to play, only supports what html video does.
<code>fields</code>	<code>[{}]</code>	Fields	An array of Attachment Field Objects .

Attachment Field Objects

The field property of the attachments allows for "tables" or "columns" to be displayed on messages.

Property	Example	Required	Description
short	true	Optional false	Default: false Whether this field should be a short field.
title	Status	Required	The title of this field.
value	online	Required	The value of this field, displayed underneath the title value.

Message Object Example

```
1  {
2      "message": {
3          "rid": "Xnb2kLD2Pnhwe3RH",
4          "msg": "Sample message",
5          "alias": "Gruggy",
6          "emoji": ":smirk:",
7          "avatar": "http://res.guggy.com/logo_128.png",
8          "attachments": [
9              {
10                 "color": "#ff0000",
11                 "text": "Yay for gruggy!",
12                 "ts": "2016-12-09T16:53:06.761Z",
13                 "thumb_url": "http://res.guggy.com/logo_128.png",
14                 "message_link": "https://google.com",
15                 "collapsed": false,
16                 "author_name": "Bradley Hilton",
17                 "author_link": "https://rocket.chat/",
18                 "author_icon": "https://avatars.githubusercontent.com/u/850391",
19                 "title": "Attachment Example",
20                 "title_link": "https://youtube.com",
21                 "title_link_download": true,
22                 "image_url": "http://res.guggy.com/logo_128.png",
23                 "audio_url": "http://www.w3schools.com/tags/horse.mp3",
24                 "video_url": "http://www.w3schools.com/tags/movie.mp4",
25                 "fields": [
26                     {
27                         "label": "Field 1"
28                     }
29                 ]
30             }
31         ]
32     }
33 }
```

```
25         "short": true,
26         "title": "Test",
27         "value": "Testing out something or other"
28     },{
29         "short": true,
30         "title": "Another Test",
31         "value": "[Link](https://google.com/) something and this and
32     }]
33 }
34 }
35 }
```

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2 -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3 -H "Content-type:application/json" \
4 http://localhost:3000/api/v1/chat.sendMessage \
5 -d '{"message": { "rid": "Xnb2kLD2Pnhdwe3RH", "msg": "This is a test!"}}
```

Example Result

```
1 {
2     "message": {
3         "rid": "GENERAL",
4         "msg": "123456789",
5         "ts": "2018-03-01T18:02:26.825Z",
6         "u": {
7             "_id": "i5FdM4ssFgAcQP62k",
8             "username": "rocket.cat",
9             "name": "test"
10        },
11        "unread": true,
12        "mentions": [],
13        "channels": [],
14        "_updatedAt": "2018-03-01T18:02:26.828Z",
15        "_id": "LnCSJxxNkCy6K9X8X"
```

```
16      },
17      "success": true
18  }
```

Change Log

Version	Description
2.4.0	Added validation on user's identity
0.60.0	Added

Star Message

Stars a chat message for the authenticated user.

URL	Requires Auth	HTTP Method
/api/v1/chat.starMessage	yes	POST

Payload

Argument	Example	Required	Description
messageId	7aDSXtjMA3KPLxLjt	Required	The message id to star.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/chat.starMessage \
5   -d '{ "messageId": "7aDSXtjMA3KPLxLjt" }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.59.0	Added

Sync Thread List

List all custom emojis available. It supports the [Query](#), [Fields](#) and [sort](#).

URL	Requires Auth	HTTP Method
/api/v1/chat.syncThreadsList	yes	GET

Query Parameters

One of roomId or roomName is required.

Argument	Example	Required	Description
rid	WyKnz8gDdS5jemBoK	Required	Room's ID
updatedSince	2017-11-25T15:08:17.248Z	Required	Date as ISO string

When you provide the `updatedSince` query parameter in the URL then the `update` and `remove` in the response will contain only those updated and removed since this date and time.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/chat.syncThreadsList?rid=GENERAL&updated
```

Example Result

```
1  {
2      "threads": [
3          "update": [
4              {
5                  "_id": "J8Sqj5zPw62WgE4Md",
6                  "rid": "GENERAL",
7                  "msg": "test1",
8                  "ts": "2019-04-16T18:16:00.614Z",
9                  "u": {
10                      "_id": "rocketchat.internal.admin.test",
11                      "username": "rocketchat.internal.admin.test",
12                      "name": "RocketChat Internal Admin Test"
13                  },
14                  "_updatedAt": "2019-04-16T18:17:09.749Z",
15                  "mentions": [],
16                  "channels": [],
17                  "replies": [
18                      "rocketchat.internal.admin.test"
19                  ],
20                  "tcount": 1,
21                  "tlm": "2019-04-16T18:17:09.672Z"
22              },
23              {
24                  "_id": "hbs2ZAjHH2JXrP7C7",
25                  "rid": "GENERAL",
26                  "msg": "another test",
27                  "ts": "2019-04-16T18:13:20.403Z",
28                  "u": {
29                      "_id": "rocketchat.internal.admin.test",
30                      "username": "rocketchat.internal.admin.test",
31                      "name": "RocketChat Internal Admin Test"
32                  },
33                  "_updatedAt": "2019-04-16T18:30:46.722Z",
34                  "mentions": [],
35                  "channels": [],
36                  "replies": [
37                      "rocketchat.internal.admin.test"
38                  ],
39                  "tcount": 1,
40                  "tlm": "2019-04-16T18:30:46.615Z"
41              }
42          ],
43          "remove": [
44              {
45                  "_id": "GXwCSs4cx6456WBJk",
46                  "rid": "GENERAL",
```

```
47         "msg": "test",
48         "ts": "2019-04-16T18:13:39.217Z",
49         "u": {
50             "_id": "rocketchat.internal.admin.test",
51             "username": "rocketchat.internal.admin.test",
52             "name": "RocketChat Internal Admin Test"
53         },
54         "_updatedAt": "2019-04-16T18:16:00.696Z",
55         "mentions": [],
56         "channels": [],
57         "replies": [
58             "rocketchat.internal.admin.test"
59         ],
60         "tcount": 1,
61         "tlm": "2019-04-16T18:16:00.614Z",
62         "_deletedAt": "2019-04-16T18:16:13.508Z",
63         "__collection__": "message"
64     }
65 ],
66 },
67 "success": true
68 }
```

Change Log

Version	Description
1.0.0	Added

Sync Thread Messages

It supports the [Query](#), [Fields](#) and [sort](#).

URL	Requires Auth	HTTP Method
/api/v1/chat.syncThreadMessages	yes	GET

Query Parameters

One of roomId or roomName is required.

Argument	Example	Required	Description
tmid	hbs2ZAjHH2JXrP7C7	Required	Thread's message ID
updatedSince	2017-11-25T15:08:17.248Z	Required	Date as ISO string

When you provide the `updatedSince` query parameter in the URL then the `update` and `remove` in the response will contain only those updated and removed since this date and time.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/chat.syncThreadMessages?tmid=hbs2ZAjHH2J
```

Example Result

```
1  {
2      "messages": {
3          "update": [
4              {
5                  "_id": "8BciMkvotHLpmpDEG",
6                  "rid": "GENERAL",
7                  "tmid": "hbs2ZAjHH2JXrP7C7",
8                  "msg": "test",
9                  "ts": "2019-04-16T18:30:46.615Z",
10                 "u": {
11                     "_id": "rocketchat.internal.admin.test",
12                     "username": "rocketchat.internal.admin.test",
13                     "name": "RocketChat Internal Admin Test"
14                 },
15                 "_updatedAt": "2019-04-16T18:30:46.669Z",
16                 "mentions": [],
17                 "channels": []
18             }
19         ],
20         "remove": []
21     },
22     "success": true
23 }
```

Change Log

Version	Description
1.0.0	Added

Unfollow Message

Unfollow a chat message to the message's channel.

URL	Requires Auth	HTTP Method
/api/v1/chat.unfollowMessage	yes	POST

Payload

Argument	Example	Required	Description
mid	7aDSXtjMA3KPLxLjt	Required	The message id to unfollow.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: Z3cpIYN6CNK2oXWKv" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/chat.unfollowMessage \
5   -d '{ "mid": "7aDSXtjMA3KPLxLjt" }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
1.0.0	Added

Unpins Message

Removes the pinned status of the provided chat message.

URL	Requires Auth	HTTP Method
/api/v1/chat.unPinMessage	yes	POST

Payload

Argument	Example	Required	Description
messageId	7aDSXtjMA3KPLxLjt	Required	The message id to unpin.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/chat.unPinMessage \
5   -d '{ "messageId": "7aDSXtjMA3KPLxLjt" }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.59.0	Added

Unstar Message

Unstars the chat message for the authenticated user.

URL	Requires Auth	HTTP Method
/api/v1/chat.unStarMessage	yes	POST

Payload

Argument	Example	Required	Description
messageId	7aDSXtjMA3KPLxLjt	Required	The message id to unpin.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/chat.unStarMessage \
5   -d '{ "messageId": "7aDSXtjMA3KPLxLjt" }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.59.0	Added

Custom Emoji

Url	Short Description	Details Page
/api/v1/emoji-custom.list	List all custom emojis on the server.	Link
/api/v1/emoji-custom.create	Create new custom emoji.	Link
/api/v1/emoji-custom.delete	Delete an existent custom emoji.	Link
/api/v1/emoji-custom.update	Update an existent custom emoji.	Link

Create

Create new custom emoji.

URL	Requires Auth	HTTP Method
/api/v1/emoji-custom.create	yes	POST

Uploading a File

Arguments

Argument	Example	Required	Description
emoji	Any image file	Required	The image file to use as the new custom emoji, as form data.
name	my-custom-emoji	Required	The name of the new custom emoji.
aliases	emoji-alias	Required	The alias of the new custom emoji.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2 -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3 -F "emoji=@my-custom-emoji.png" \
4 -F "name=my-custom-emoji" \
5 -F "aliases=emoji-alias" \
6 http://localhost:3000/api/v1/emoji-custom.create
```

Example Result

```
1  {
2    "success": true
3 }
```

Change Log

Version	Description
0.74.0	Added

Delete

URL	Requires Auth	HTTP Method
/api/v1/emoji-custom.delete	yes	POST

Payload

Argument	Example	Required	Description
emojiId	ByehQjC44FwMeiLbX	Required	The custom emoji id to be deleted.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -H "Content-type:application/json" \
4     http://localhost:3000/api/v1/emoji-custom.delete \
5     -d '{ "emojiId": "ByehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
2     "success": true
3 }
```

Change Log

Version	Description
0.74.0	Added

List

List all custom emojis available. It supports the [Query Parameters only](#).

URL	Requires Auth	HTTP Method
/api/v1/emoji-custom.list	yes	GET

Query Parameters

Argument	Example	Required	Description
updatedSince	2017-11-25T15:08:17.248Z	Optional	Date as ISO string

When you provide the `updatedSince` query parameter in the URL then the `update` and `remove` in the response will contain only those updated and removed since this date and time.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/emoji-custom.list
```

Example Result

```
1 {
```

```
2     "emojis": {
3         "update": [
4             {
5                 "_id": "S5XvYppoLrLd9JvQm",
6                 "name": "teste",
7                 "aliases": [],
8                 "extension": "jpg",
9                 "_updatedAt": "2019-02-18T16:48:35.119Z"
10            },
11            {
12                "_id": "Ro5HD4wKQiYnrbpbg",
13                "name": "aaaaaaaa",
14                "aliases": [
15                    "aaaaaaa"
16                ],
17                "extension": "png",
18                "_updatedAt": "2019-02-18T16:49:47.310Z"
19            }
20        ],
21        "remove": []
22    },
23    "success": true
24 }
```

Example Call with Query parameter

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/emoji-custom.list?query={"name": "teste"}
```

Example Result

```
1 [
2     "emojis": {
3         "update": [
4             {
5                 "_id": "S5XvYppoLrLd9JvQm",
```

```
6             "name": "teste",
7             "aliases": [],
8             "extension": "jpg",
9             "_updatedAt": "2019-02-18T16:48:35.119Z"
10            }
11        ],
12        "remove": []
13    },
14    "success": true
15 }
```

Example Call with `updatedSince`

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/emoji-custom.list?updatedSince=2019-02-2
```

Example Result

```
1  {
2     "emojis": {
3         "update": [
4             {
5                 "_id": "S5XvYppoLrLd9JvQm",
6                 "name": "teste",
7                 "aliases": [],
8                 "extension": "jpg",
9                 "_updatedAt": "2019-02-18T16:48:35.119Z"
10            }
11        ],
12        "remove": [
13            {
14                "_id": "2dbVBG434dnsdh23",
15                "name": "teste3",
16                "aliases": [],
17                "extension": "jpg",
18                "_updatedAt": "2019-02-18T16:48:35.119Z"
```

```
19          }
20      ]
21  },
22  "success": true
23 }
```

Change Log

Version	Description
0.75.0	Added

Update

Update an existent custom emoji.

URL	Requires Auth	HTTP Method
/api/v1/emoji-custom.update	yes	POST

Uploading a File

Arguments

Argument	Example	Required	Description
emoji	Any image file	Optional	The image file to use as custom emoji, as form data.
_id	AG7DSB2H32YHS	Required	The id of the custom emoji.
name	my-custom-emoji	Required	The name of the custom emoji.
aliases	emoji-alias	Optional	The alias of the custom emoji.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -F "emoji=@my-custom-emoji.png" \
4   -F "_id=AG7DSB2H32YHS" \
5   -F "name=my-custom-emoji" \
6   -F "aliases=emoji-alias" \
7   http://localhost:3000/api/v1/emoji-custom.update
```

Example Result

```
1  {
2      "success": true
3 }
```

Change Log

Version	Description
0.74.0	Added

Custom Sounds

Url	Short Description	Details Page
/api/v1/custom-sounds.list	Retrieves a list of custom sounds.	Link

Custom Sounds List

Retrieves all custom sounds. It supports the [Offset](#), [Count](#), and [Sort](#) Query Parameters along with [Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/custom-sounds.list	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUFD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/custom-sounds.list
```

Example Result

```
1 [
2   "sounds": [
3     {
4       "_id": "qmdQjHZYzf4Hr7LwM",
5       "name": "sound 1",
6       "extension": "mp3",
7       "_updatedAt": "2019-12-03T14:30:02.315Z"
8     },
9   ],
10  "count": 1,
11  "offset": 0,
12  "total": 29,
13  "success": true
14 }
```

Change Log

Version	Description
2.4.0	Added

Custom User Status

Url	Short Description	Details Page
/api/v1/custom-user-status.list	Lists all available custom user's status.	Link

Custom User Status List

Lists all available custom user's status.

URL	Requires Auth	HTTP Method
/api/v1/custom-user-status.list	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/custom-user-status.list
```

Example Result

```
1 {
2   "statuses": [
3     {
4       "_id": "oGxJ2irQcPkgFLGef",
5       "name": "status 1",
6       "statusType": "busy",
7       "_updatedAt": "2019-12-09T20:26:06.477Z"
8     },
9     {
10       "_id": "d3TjnQgKWttgaW9Hh",
11       "name": "status 2",
12       "statusType": "online",
13       "_updatedAt": "2019-12-09T20:25:10.884Z"
14     }
15   ],
16   "count": 2,
17   "offset": 0,
18   "total": 4,
```

```
19      "success": true  
20  }
```

Change Log

Version	Description
2.4.0	Added

Groups

Url	Short Description	Details Page
/api/v1/groups.addAll	Adds all of the users on the server to a private group.	Link
/api/v1/groups.addLeader	Gives the role of Leader for a user in the current group.	Link
/api/v1/groups.addModerator	Gives the role of moderator to a user in a group.	Link
/api/v1/groups.addOwner	Gives the role of owner to a user in a group.	Link
/api/v1/groups.archive	Archives a private group.	Link
/api/v1/groups.close	Removes a private group from the list of groups.	Link
/api/v1/groups.counters	Gets group counters.	Link
/api/v1/groups.create	Creates a new private group.	Link
/api/v1/groups.delete	Removes a private group.	Link
/api/v1/groups.files	Gets a list of files from a private group.	Link
/api/v1/groups.getIntegrations	Gets the integrations assigned to the group.	Link
/api/v1/groups.history	Retrieves the messages from a private group.	Link
/api/v1/groups.info	Gets the information about a private group.	Link
/api/v1/groups.invite	Adds a user to the private group.	Link

/api/v1/groups.kick	Removes a user from a private group.	Link
/api/v1/groups.leave	Removes the calling user from the private group.	Link
/api/v1/groups.list	List the private groups the caller is part of.	Link
/api/v1/groups.listAll	List all the private groups.	Link
/api/v1/groups.moderators	List all moderators of a group.	Link
/api/v1/groups.members	Gets the users of participants of a private group.	Link
/api/v1/groups.messages	Retrieves all group messages.	Link
/api/v1/groups.online	List all online users of a group.	Link
/api/v1/groups.open	Adds the private group back to the list of groups.	Link
/api/v1/groups.removeLeader	Removes the role of Leader for a user in the current group.	Link
/api/v1/groups.removeModerator	Removes the role of moderator from a user in a group.	Link
/api/v1/groups.removeOwner	Removes the role of owner from a user in a group.	Link
/api/v1/groups.rename	Changes the name of the private group.	Link
/api/v1/groups.roles	Gets the user's roles in the private group.	Link
/api/v1/groups.setAnnouncement	Sets a group's announcement.	Link
/api/v1/groups.setCustomFields	Sets private group's custom fields.	Link
/api/v1/groups.setDescription	Sets a private group's description.	Link

/api/v1/groups.setPurpose	Sets a private group's description.	Link
/api/v1/groups.setReadOnly	Sets whether the room is read only or not.	Link
/api/v1/groups.setTopic	Sets a private group's topic.	Link
/api/v1/groups.setType	Sets the type of room this group will be.	Link
/api/v1/groups.unarchive	Unarchives a private group.	Link

Group Online

URL	Requires Auth	HTTP Method
/api/v1/groups.online	yes	GET

Query Parameters

Argument	Example	Required	Description
query	{"_id":"5HmCfp0B7jp2uibTC"}	Optional	See Query Parameter

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/groups.online
```

Example Result

```
1 {
2     "online": [
3         {
4             "_id": "47cRd58HnWwpqxhaZ",
5             "username": "test"
6         },
7         {
8             "_id": "BsxzC22xQ43taWdff",
9             "username": "uniqueusername"
```

```
10      }
11    ],
12    "success": true
13 }
```

Query Example Call

This example shows how to filter using group's id.

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/groups.online?query={"_id": "5HmCfpoB7jp2uibT"
```

Query Example Result

```
1  {
2    "online": [
3      {
4        "_id": "47cRd58HnWwpqxhaZ",
5        "username": "test"
6      }
7    ],
8    "success": true
9  }
```

Change Log

Version	Description
0.52.0	Added

Get Integrations

Retrieves the integrations which the group has, requires at least one integration permission:

`manage-incoming-integrations` , `manage-own-incoming-integrations` ,
`manage-outgoing-integrations` or `manage-own-outgoing-integrations` . It will return
the integrations based on the user permission. It supports the [Offset, Count, and Sort Query
Parameters](#) along with [Query and Fields Query Parameters](#).

URL	Requires Auth	HTTP Method
<code>/api/v1/groups.getIntegrations</code>	yes	GET

Query Parameters

Argument	Example	Required	Description
<code>roomId</code>	<code>ByehQjC44FwMeiLbX</code>	Required	The group's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCuFD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/groups.getIntegrations?roomId=ByehQjC44F
```

Example Result

```
1 {
```

```

2     "integrations": [{  
3         "_id": "WMQDChpnYTRmFre9h",  
4         "enabled": true,  
5         "username": "rocket.cat",  
6         "alias": "Guggy",  
7         "avatar": "http://res.guggy.com/logo_128.png",  
8         "name": "Guggy",  
9         "triggerWords": [  
10            "!guggy",  
11            "guggy",  
12            "gif+"  
13        ],  
14        "urls": [  
15            "http://text2gif.guggy.com/guggify"  
16        ],  
17        "token": "8DFS89DMKLWEN",  
18        "script": "const config = {\n          color: '#ffffff'\n};\nclass Scr  
19        "scriptEnabled": true,  
20        "impersonateUser": false,  
21        "scriptCompiled": "function _classCallCheck(instance,Constructor){  
22          scriptError: null,  
23          type: \"webhook-outgoing\",  
24          userId: \"rocket.cat\",  
25          channel: [  
26            \"#private-group-name\"\n          ],  
27          _createdAt: \"2017-01-05T17:06:05.660Z\",  
28          _createdBy: {  
29            username: \"graywolf336\",  
30            _id: \"R4jgcQaQhvK6K3iY\"\n          },  
31          _updatedAt: \"2017-01-05T17:06:05.660Z\"\n        }],  
32        "success": true  
33      }  
34    }  
35  }  
36 }
```

Change Log

Version	Description
1.1.0	Separate permissions in <code>incoming</code> and <code>outgoing</code> .
0.49.0	Added

Group Add All

Adds all of the users of the Rocket.Chat server to the group.

URL	Requires Auth	HTTP Method
/api/v1/groups.addAll	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id
activeUsersOnly	true	Optional Default: false	Add active users only

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/channels.addAll \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
```

```
2     "channel": {
3         "_id": "ByehQjC44FwMeiLbX",
4         "name": "groupname",
5         "t": "p",
6         "usernames": [
7             "example",
8             "rocket.cat"
9         ],
10        "msgs": 0,
11        "u": {
12            "_id": "aobEdbYhXfu5hkeqG",
13            "username": "example"
14        },
15        "ts": "2016-05-30T13:42:25.304Z"
16    },
17    "success": true
18 }
```

Change Log

Version	Description
0.55.0	Added

Group Add Leader

Gives the role of Leader for a user in the current group.

URL	Requires Auth	HTTP Method
/api/v1/groups.addLeader	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The group's id
userId	oCHkav5Zf6vmpu2W2	Required	The user's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.addLeader \
5   -d '{"roomId": "ByehQjC44FwMeiLbX", "userId": "oCHkav5Zf6vmpu2W2"}'
```

Success Example Result

```
1 {
2   "success": true
```

```
3 }
```

Bad Request Example Result

If the user is already a leader, will return a `400 bad request` status.

```
1 {
2   "success": false,
3   "error": "User is already a leader [error-user-already-leader]",
4   "errorType": "error-user-already-leader"
5 }
```

Change Log

Version	Description
0.58.0	Added.

Group Add Moderator

Gives the role of moderator for a user in the current group.

URL	Requires Auth	HTTP Method
/api/v1/groups.addModerator	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The private group's id
userId	nSYqWzz4GsKTX4dyK	Required	The user id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.addModerator \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "userId": "nSYqWzz4GsKTX4dyK" }'
```

Example Result

```
1 {
2   "success": true
```

```
3 }
```

Change Log

Version	Description
0.49.4	Added

Group Add Owner

Gives the role of owner for a user in the current group.

URL	Requires Auth	HTTP Method
/api/v1/groups.addOwner	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The private group's id
userId	nSYqWzz4GsKTX4dyK	Required	The user id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.addOwner \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "userId": "nSYqWzz4GsKTX4dyK" }'
```

Example Result

```
1 {
2   "success": true
```

```
3 }
```

Change Log

Version	Description
0.49.4	Added

Group Archive

Archives a private group, only if you're part of the group.

URL	Requires Auth	HTTP Method
/api/v1/groups.archive	yes	POST

Payload

Argument	Example	Required	Description
roomId	HyehQjC44FwMeiLbX	Required	The private groups's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.archive \
5   -d '{ "roomId": "HyehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.48.0	Added

Group Close

Removes the private group from the user's list of groups, only if you're part of the group.

URL	Requires Auth	HTTP Method
/api/v1/groups.close	yes	POST

Payload

Argument	Example	Required	Description
roomId	HyehQjC44FwMeiLbX	Required	The group's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.close \
5   -d '{ "roomId": "HyehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.48.0	Added

Group Counters

Gets group counters.

URL	Requires Auth	HTTP Method
/api/v1/groups.counters	yes	GET

Payload

One of roomId or roomName is required.

Argument	Example	Required	Description
roomId	euzoT67Gx6nXcn66M	Required	The group's id
roomName	channel.cf.1518990680083	Required	The group's name
userId	RtycPC29hqLJfT9xj	Optional	Counters for provided user id (need to have a view-room-administration right for calling user)

Example Call

```
1 curl -H "X-Auth-Token: 8h2mKAwxB3AQrFSjLVKMooJyjdCFaA7W45sWlHP8Iz0" \
2   -H "X-User-Id: ew28FnZqipDpvKw3R" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.counters?roomId=euzoT67Gx6nXcn66M
```

Example Result

```
1  {
2    "joined":true,
3    "members":1,
4    "unread":1,
5    "unreadFrom":"2018-02-18T21:51:20.091Z",
6    "msgs":1,
7    "latest":"2018-02-23T17:20:17.345Z",
8    "userMentions":0,
9    "success":true
10 }
11
12 where:
13
14 joined - boolean flag that shows that user is joined the room or not
15 members - count of current room members
16 unread - count of unread messages for specified user (calling user or pro
17 unreadFrom - start date-time of unread interval for specified user
18 msgs - count of messages in the room
19 latest - end date-time of unread interval for specified user (or date-time
20 userMentions - count of user mentions in messages
```

Change Log

Version	Description
0.65.0	Added

Group Create

Creates a new private group, optionally including specified users. The group creator is always included.

URL	Requires Auth	HTTP Method
/api/v1/groups.create	yes	POST

Payload

Argument	Example	Required	Description
name	testing	Required	The name of the new private group
members	["rocket.cat"]	Optional Default: []	The users to add to the group when it is created.
readOnly	true	Optional Default: false	Set if the group is read only or not.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -H "Content-type: application/json" \
4     http://localhost:3000/api/v1/groups.create \
5     -d '{ "name": "testing" }'
```

Example Result

```
1  {
2    "group": {
3      "_id": "NtR6RQ7NvzA9ejecX",
4      "name": "testing",
5      "t": "p",
6      "usernames": [
7        "tester"
8      ],
9      "msgs": 0,
10     "u": {
11       "_id": "aobEdbYhXfu5hkeqG",
12       "username": "tester"
13     },
14     "ts": "2016-12-09T16:53:06.761Z",
15     "ro": false,
16     "sysMes": true,
17     "_updatedAt": "2016-12-09T16:53:06.761Z"
18   },
19   "success": true
20 }
```

Change Log

Version	Description
0.35.0	Added

Group Delete

Remove a private channel.

URL	Requires Auth	HTTP Method
/api/v1/groups.delete	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required (if no roomName)	The channel's id
roomName	general	Required (if no roomId)	The channel's name

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.delete \
5   -d '{ "roomName": "channelname" }'
```

Example Result

```
1 {
2     "success": true
```

```
3 }
```

Change Log

Version	Description
0.71.0	Removed <code>group</code> property
0.49.0	Added

Group History

Retrieves the messages from a private group, only if you're part of the group.

URL	Requires Auth	HTTP Method
/api/v1/groups.history	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The private groups's id
latest	2016-09-30T13:42:25.304Z	Optional Default: now	The end of time range of messages to retrieve
oldest	2016-05-30T13:42:25.304Z	Optional Default: n/a	The start of the time range of messages to retrieve
inclusive	true	Optional Default: false	Whether messages which land on latest and oldest should be included
offset	10	Optional Default: 0	The offset of the list messages to retrieve
count	100	Optional Default: 20	The amount of messages to retrieve
unread	false	Optional Default: false	Whether the amount of unread should be included.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/groups.history?roomId=ByehQjC44FwMeiLbX
```

Example Result

```
1 {
2   "messages": [
3     {
4       "_id": "AkzpHAvZpdnuchw2a",
5       "rid": "ByehQjC44FwMeiLbX",
6       "msg": "hi",
7       "ts": "2016-12-09T12:50:51.555Z",
8       "u": {
9         "_id": "y65tAmHs93aDChMWu",
10        "username": "testing"
11      },
12      "_updatedAt": "2016-12-09T12:50:51.562Z"
13    },
14    {
15      "_id": "vkLMxccR4MuTxreF",
16      "t": "uj",
17      "rid": "ByehQjC44FwMeiLbX",
18      "ts": "2016-12-08T15:41:37.730Z",
19      "msg": "testing2",
20      "u": {
21        "_id": "bRtgdhzM6PD9F8pSx",
22        "username": "testing2"
23      },
24      "groupable": false,
25      "_updatedAt": "2016-12-08T16:03:25.235Z"
26    },
27    {
28      "_id": "bfRW658nEyEBg75rc",
29      "t": "uj",
30      "rid": "ByehQjC44FwMeiLbX",
31      "ts": "2016-12-07T15:47:49.099Z",
32      "msg": "testing",
33      "u": {
```

```
34         "_id": "nSYqWzz4GsKTX4dyK",
35         "username": "testing1"
36     },
37     "groupable": false,
38     "_updatedAt": "2016-12-07T15:47:49.099Z"
39   },
40   {
41     "_id": "pbuFiGadhRZTKouhB",
42     "t": "uj",
43     "rid": "ByehQjC44FwMeiLbX",
44     "ts": "2016-12-06T17:57:38.635Z",
45     "msg": "testing",
46     "u": {
47       "_id": "y65tAmHs93aDChMWu",
48       "username": "testing"
49     },
50     "groupable": false,
51     "_updatedAt": "2016-12-06T17:57:38.635Z"
52   }
53 ],
54   "success": true
55 }
```

Change Log

Version	Description
0.75.0	Added <code>offset</code> property
0.48.0	Added

Group Info

Retrieves the information about the private group, only if you're part of the group.

URL	Requires Auth	HTTP Method
/api/v1/groups.info	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required (if no roomName)	The private group's id
roomName	Exclusive.Chat	Required (if no roomId)	The private group's name

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/groups.info?roomId=ByehQjC44FwMeiLbX
```

Example Result

```
1 {
2   "group": {
```

```
3      "_id": "ByehQjC44FwMeiLbX",
4      "name": "testing",
5      "fname": "testing",
6      "t": "p",
7      "msgs": 0,
8      "usersCount": 2,
9      "u": {
10        "_id": "HKKPmF8rZh45GMHWH",
11        "username": "marcos.defendi"
12      },
13      "customFields": {},
14      "broadcast": false,
15      "encrypted": false,
16      "ts": "2020-05-21T13:16:24.749Z",
17      "ro": false,
18      "default": false,
19      "sysMes": true,
20      "_updatedAt": "2020-05-21T13:16:24.772Z"
21    },
22    "success": true
23  }
```

Change Log

Version	Description
0.48.0	Added

Group Invite

Adds a user to the private group.

URL	Requires Auth	HTTP Method
/api/v1/groups.invite	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The private group's id
userId	nSYqWzz4GsKTX4dyK	Required	The user id of who to invite

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.invite \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "userId": "nSYqWzz4GsKTX4dyK" }'
```

Example Result

```
1 {
2   "group": {
```

```
3      "_id": "ByehQjC44FwMeiLbX",
4      "ts": "2016-11-30T21:23:04.737Z",
5      "t": "p",
6      "name": "testing",
7      "usernames": [
8          "testing",
9          "testing1"
10     ],
11     "u": {
12         "_id": "aobEdbYhXfu5hkeqG",
13         "username": "testing1"
14     },
15     "msgs": 1,
16     "_updatedAt": "2016-12-09T12:50:51.575Z",
17     "lm": "2016-12-09T12:50:51.555Z"
18   },
19   "success": true
20 }
```

Change Log

Version	Description
0.48.0	Added

Group Kick

Removes a user from the private group.

URL	Requires Auth	HTTP Method
/api/v1/groups.kick	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The private group's id
userId	nSYqWzz4GsKTX4dyK	Required	The user id of who to invite

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.kick \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "userId": "nSYqWzz4GsKTX4dyK" }'
```

Example Result

```
1 {
2   "group": {
```

```
3      "_id": "ByehQjC44FwMeiLbX",
4      "name": "invite-me",
5      "t": "p",
6      "usernames": [
7          "testing1"
8      ],
9      "msgs": 0,
10     "u": {
11         "_id": "aobEdbYhXfu5hkeqG",
12         "username": "testing1"
13     },
14     "ts": "2016-12-09T15:08:58.042Z",
15     "ro": false,
16     "sysMes": true,
17     "_updatedAt": "2016-12-09T15:22:40.656Z"
18 },
19     "success": true
20 }
```

Change Log

Version	Description
0.48.0	Added

Group Leave

Causes the callee to be removed from the private group, if they're part of it and are not the last owner.

URL	Requires Auth	HTTP Method
/api/v1/groups.leave	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The private group's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.leave \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
2   "group": {
3     "_id": "ByehQjC44FwMeiLbX",
```

```
4      "name": "invite-me",
5      "t": "p",
6      "usernames": [
7          "testing2"
8      ],
9      "msgs": 0,
10     "u": {
11         "_id": "aobEdbYhXfu5hkeqG",
12         "username": "testing1"
13     },
14     "ts": "2016-12-09T15:08:58.042Z",
15     "ro": false,
16     "sysMes": true,
17     "_updatedAt": "2016-12-09T15:22:40.656Z"
18 },
19     "success": true
20 }
```

Change Log

Version	Description
0.48.0	Added

Group List All

Lists all of the private groups of any users. The calling user requires to have 'view-room-administration' right. It supports the [Offset, Count, and Sort Query Parameters](#) along with just the [Fields Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/groups.listAll	yes	GET

Example Call

This example shows a list of private groups filtered by "customFields.companyId" started with "org1" using a regular expression.

```
1 curl -H "X-Auth-Token: 8-gard51USVYskZ7AAqFF3SZuwg24VIdn9-HchYersg" \
2     -H "X-User-Id: 3WpJQkDHhrWPBvXuW" \
3     http://localhost:3000/api/v1/groups.listAll?query=%7B%20%22customFiel
```

Example Result

```
1 {
2     "groups": [
3         {
4             "_id": "xA52DRDM7dqx2PfTp",
5             "name": "private1",
6             "fname": "private1",
7             "t": "p",
8             "msgs": 0,
9             "u": {
10                 "_id": "3WpJQkDHhrWPBvXuW",
11                 "username": "admin"
```

```
12     },
13     "customFields": {
14       "companyId": "org1"
15     },
16     "ts": "2018-01-21T21:05:06.729Z",
17     "ro": false,
18     "sysMes": true,
19     "_updatedAt": "2018-01-21T21:05:06.729Z"
20   }
21 ],
22 "offset": 0,
23 "count": 1,
24 "total": 1,
25 "success": true
26 }
```

Change Log

Version	Description
0.62.0	Add 'query' parameter support.
0.59.0	Added

Group List

Lists all of the private groups the calling user has joined. It supports the [Offset](#), [Count](#), and [Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/groups.list	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUFD_0YdwnNnunAJF8V47U3QHXSq" \
2       -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3       http://localhost:3000/api/v1/groups.list
```

Example Result

```
18         "_id": "t7qapfhZjANMRAi5w",
19         "name": "testing",
20         "t": "p",
21         "msgs": 0,
22         "u": {
23             "_id": "y65tAmHs93aDChMWu",
24             "username": "testing2"
25         },
26         "ts": "2016-12-01T15:08:58.042Z",
27         "ro": false,
28         "sysMes": true,
29         "_updatedAt": "2016-12-09T15:22:40.656Z"
30     }
31 ],
32 "offset": 0,
33 "count": 1,
34 "total": 1,
35 "success": true
36 }
```

Change Log

Version	Description
0.67.0	Removed <code>query</code> parameter support.
0.62.0	Add <code>query</code> parameter support.
0.49.0	Count and offset query parameters supported.
0.33.0	Added

Group Members

Lists the users of participants of a private group. It supports the [Offset, Count, and Sort Query Parameters](#).

Requires `view-broadcast-member-list` if room is a broadcast.

URL	Requires Auth	HTTP Method
<code>/api/v1/groups.members</code>	yes	GET

Query Parameters

Argument	Example	Required	Description
<code>roomId</code>	<code>ByehQjC44FwMeiLbX</code>		The room id.
<code>roomName</code>	<code>user</code>	Required	The room name.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/groups.members?roomId=ByehQjC44FwMeiLbX
```

Example Result

```
1 {
```

```
2     "members": [
3         {
4             "_id": "Q4GkX6RMepGDdQ7YJ",
5             "status": "online",
6             "name": "Marcos Defendi",
7             "utcOffset": -3,
8             "username": "marcos.defendi"
9         },
10        {
11            "_id": "rocket.cat",
12            "name": "Rocket.Cat",
13            "username": "rocket.cat",
14            "status": "online",
15            "utcOffset": 0
16        }
17    ],
18    "count": 2,
19    "offset": 0,
20    "total": 2,
21    "success": true
22 }
```

Change Log

Version	Description
0.59.0	Added

Group messages

Lists all of the specific group messages on the server. It supports the [Offset](#), [Count](#), and [Sort Query Parameters](#) along with [Query and Fields Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/groups.messages	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required(if no roomName)	The private group's id
roomName	testGroup	Required(if no roomId)	The private group's name

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/groups.messages?roomId=ByehQjC44FwMeiLbX
```

Example Result

```
1  {
2      "messages": [
3          {
4              "_id": "CeXwh5eBbdrtvnqG6",
5              "rid": "agh2Sucgb54RQ8dDo",
6              "msg": "s",
7              "ts": "2018-10-05T13:48:21.616Z",
8              "u": {
9                  "_id": "KPkEYwKKBKZnEEPpt",
10                 "username": "marcos.defendi",
11                 "name": "Marcos Defendi"
12             },
13             "_updatedAt": "2018-10-05T13:48:49.535Z",
14             "reactions": {
15                 ":frowning2:": {
16                     "usernames": [
17                         "marcos.defendi"
18                     ]
19                 }
20             },
21             "mentions": [],
22             "channels": [],
23             "starred": {
24                 "_id": "KPkEYwKKBKZnEEPpt"
25             }
26         },
27         {
28             "_id": "MrAeupRiF9TvhMesK",
29             "t": "room_changed_privacy",
30             "rid": "agh2Sucgb54RQ8dDo",
31             "ts": "2018-10-05T00:11:16.998Z",
32             "msg": "Private Group",
33             "u": {
34                 "_id": "rocketchat.internal.admin.test",
35                 "username": "rocketchat.internal.admin.test"
36             },
37             "groupable": false,
38             "_updatedAt": "2018-10-05T00:11:16.998Z"
39         }
40     ],
41     "count": 2,
42     "offset": 0,
43     "total": 2,
44     "success": true
45 }
```

Change Log

Version	Description
0.59.0	Added

Group Moderators List

Lists all group moderators.

URL	Requires Auth	HTTP Method
/api/v1/groups.moderators	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required (if no roomName)	The group's id
roomName	general	Required (if no roomId)	The group's name

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/groups.moderators?roomId=ByehQjC44FwMeiL
```

Example Result

```
1 {
2   "moderators": [
3     {
4       "_id": "rocket.cat",
```

```
5           "username": "rocket.cat",
6           "name": Rocket.Cat
7       }
8   ],
9   "success": true
10 }
```

Change Log

Version	Description
0.70.0	Added

Group Open

Adds the private group back to the user's list of private groups.

URL	Requires Auth	HTTP Method
/api/v1/groups.open	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The private group's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.open \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.48.0	Added

Group Remove Leader

Removes the role of Leader for a user in the current group.

URL	Requires Auth	HTTP Method
/api/v1/groups.removeLeader	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The group's id
userId	oCHkav5Zf6vmpu2W2	Required	The user's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.removeLeader \
5   -d '{"roomId": "ByehQjC44FwMeiLbX", "userId": "oCHkav5Zf6vmpu2W2"}'
```

Success Example Result

```
1 {
2   "success": true
```

```
3 }
```

Bad Request Example Result

If the user is not a leader, will return a `400 bad request` status.

```
1 {
2   "success": false,
3   "error": "User is not a leader [error-user-not-leader]",
4   "errorType": "error-user-not-leader"
5 }
```

Change Log

Version	Description
0.58.0	Added.

Group Remove Moderator

Removes the role of moderator from a user in the current group.

URL	Requires Auth	HTTP Method
/api/v1/groups.removeModerator	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The private group's id
userId	nSYqWzz4GsKTX4dyK	Required	The user's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.removeModerator \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "userId": "nSYqWzz4GsKTX4dyK" }'
```

Example Result

```
1 {
2   "success": true
```

3 }

Change Log

Version	Description
0.49.4	Added

Group Remove Owner

Removes the role of owner from a user in the current Group.

URL	Requires Auth	HTTP Method
/api/v1/groups.removeOwner	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The private group's id
userId	nSYqWzz4GsKTX4dyK	Required	The user's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.removeOwner \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "userId": "nSYqWzz4GsKTX4dyK" }'
```

Example Result

```
1 {
2   "success": true
```

```
3 }
```

Change Log

Version	Description
0.49.4	Added

Group Rename

Changes the name of the private group.

URL	Requires Auth	HTTP Method
/api/v1/groups.rename	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The private group's id
name	new-name	Required	The new name the private group will have, it can not be the same as another or the existing name

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.rename \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "name": "new-name" }'
```

Example Result

```
1  {
2    "group": {
3      "_id": "ByehQjC44FwMeiLbX",
4      "name": "new-name",
5      "t": "p",
6      "usernames": [
7        "testing1"
8      ],
9      "msgs": 4,
10     "u": {
11       "_id": "aobEdbYhXfu5hkeqG",
12       "username": "testing1"
13     },
14     "ts": "2016-12-09T15:08:58.042Z",
15     "ro": false,
16     "sysMes": true,
17     "_updatedAt": "2016-12-09T15:57:44.686Z"
18   },
19   "success": true
20 }
```

Change Log

Version	Description
0.48.0	Added

Group Set Announcement

Sets the announcement for the group.

URL	Requires Auth	HTTP Method
/api/v1/groups.setAnnouncement	yes	POST

Payload

Argument	Example	Required	Announcement
roomId	ByehQjC44FwMeiLbX	Required	The group's id
announcement	Test out everything.	Required	The announcement to set for the group.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.setAnnouncement \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "announcement": "Test out everyt' 
```

Example Result

```
1 { 
```

```
2   "announcement": "Test out everything.",  
3   "success": true  
4 }
```

Change Log

Version	Description
0.70.0	Added

Group Set Custom Fields

Sets the custom fields for the private group.

URL	Requires Auth	HTTP Method
/api/v1/groups.setCustomFields	yes	POST

Payload

One of roomId or roomName is required.

Argument	Example	Required	Description
roomId	euzoT67Gx6nXcn66M	Required	The private group's id
roomName	channel.cf.1518990680083	Required	The private group's name
customFields	{"company": "sell-and-more"}	Required	The custom fields to set for the private group.

Example Call

```
1 curl -H "X-Auth-Token: 8h2mKAwxB3AQrFSjLVKMooJyjdCFaA7W45sWlHP8Iz0" \
2   -H "X-User-Id: ew28FnZqipDpvKw3R" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.setCustomFields \
5   -d '{"roomId": "euzoT67Gx6nXcn66M", "customFields": {"company": "sell-
```

Example Result

```
1  {
2    "group": {
3      "_id": "euzoT67Gx6nXcn66M",
4      "name": "channel.cf.1518990680083",
5      "fname": "channel.cf.1518990680083",
6      "t": "p",
7      "msgs": 0,
8      "u": {
9        "_id": "ew28FnZqipDpvKw3R",
10       "username": "rocketchat.internal.admin.test"
11     },
12     "customFields": {
13       "company": "sell-and-more"
14     },
15     "ts": "2018-02-18T21:51:20.091Z",
16     "ro": false,
17     "sysMes": true,
18     "_updatedAt": "2018-02-23T11:27:01.411Z",
19     "username": "rocketchat.internal.admin.test"
20   },
21   "success": true,
22   "developerWarning": "[WARNING]: The \"usernames\" field has been removed"
23 }
```

Change Log

Version	Description
0.62.0	Added

Group Set Description

Sets the description for the private group.

URL	Requires Auth	HTTP Method
/api/v1/groups.setDescription	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The private group's id
description	Test out everything.	Required	The description to set for the private group.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.setDescription \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "description": "Test out everyth' 
```

Example Result

```
1 { 
```

```
2   "description": "Testing out everything.",  
3   "success": true  
4 }
```

Change Log

Version	Description
0.48.0	Added

Group Set Purpose

Sets the description for the private group (the same as groups.setDescription, obsolete).

URL	Requires Auth	HTTP Method
/api/v1/groups.setPurpose	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The private group's id
purpose	Test out everything.	Required	The description to set for the private group.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.setPurpose \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "purpose": "Test out everything"}
```

Example Result

```
1 {
```

```
2   "purpose": "Testing out everything.",  
3   "success": true  
4 }
```

Change Log

Version	Description
0.48.0	Added

Group Set Read Only

Sets whether the group is read only or not.

URL	Requires Auth	HTTP Method
/api/v1/groups.setReadOnly	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMei5LbX	Required	The group's id
readOnly	true	Required	Boolean of whether the group is read only or not.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.setReadOnly \
5   -d '{ "roomId": "ByehQjC44FwMei5LbX", "readOnly": true }'
```

Example Result

```
1 {
```

```
2     "group": {
3         "_id": "ByehQjC44FwMei5LbX",
4         "name": "testing-private",
5         "t": "p",
6         "msgs": 0,
7         "u": {
8             "_id": "aiPqNoGkjpNDiRx6d",
9             "username": "goose160"
10        },
11        "ts": "2017-01-05T18:02:50.754Z",
12        "ro": true,
13        "sysMes": true,
14        "_updatedAt": "2017-01-05T19:02:24.429Z",
15        "usernames": [
16            "goose160",
17            "graywolf336"
18        ],
19        "joinCodeRequired": true,
20        "muted": []
21    },
22    "success": true
23 }
```

Change Log

Version	Description
0.49.0	Added

Group Set Topic

Sets the topic for the private group.

URL	Requires Auth	HTTP Method
/api/v1/groups.setTopic	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The private group's id
topic	Discuss all of the testing.	Required	The private group's topic to set.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.setTopic \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "topic": "Discuss all of the tes'
```

Example Result

```
1 {
```

```
2   "topic": "Testing out everything.",  
3   "success": true  
4 }
```

Change Log

Version	Description
0.48.0	Added

Group Set Type

Sets the type of room this group should be.

URL	Requires Auth	HTTP Method
/api/v1/groups.setType	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The group's id
type	c or p	Required	The type of room this group should be, either c or p .

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.setType \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "type": "c" }'
```

Example Result

```
1 {
```

```
2     "group": {
3         "_id": "ByehQjC44FwMeiLbX",
4         "name": "testing0",
5         "t": "c",
6         "msgs": 0,
7         "u": {
8             "_id": "aiPqNoGkjpNDiRx6d",
9             "username": "goose160"
10        },
11        "ts": "2017-01-05T18:02:50.754Z",
12        "ro": false,
13        "sysMes": true,
14        "_updatedAt": "2017-01-05T19:02:24.429Z",
15        "usernames": [
16            "goose160",
17            "graywolf336"
18        ],
19        "joinCodeRequired": true,
20        "muted": []
21    },
22    "success": true
23 }
```

Change Log

Version	Description
0.49.0	Added

Group Unarchive

Unarchives a private group.

URL	Requires Auth	HTTP Method
/api/v1/groups.unarchive	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The private group's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/groups.unarchive \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.48.0	Added

Groups Files

Retrieves the files from a private group. It supports the [Offset, Count, and Sort Query Parameters](#) along with [Query and Fields Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/groups.files	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX		The room id.
roomName	test	Required	The room name.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/groups.files?roomId=ByehQjC44FwMeiLbX
```

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/groups.files?roomName=test
```

Example Result

```
1  {
2      "files": [
3          {
4              "_id": "S78TNnvaWGwdYRaCD",
5              "name": "images.jpeg",
6              "size": 9778,
7              "type": "image/jpeg",
8              "rid": "GENERAL",
9              "description": "",
10             "store": "GridFS:Uploads",
11             "complete": true,
12             "uploading": false,
13             "extension": "jpeg",
14             "progress": 1,
15             "user": {
16                 "_id": "ksKsKmrjvxzkzxkw",
17                 "username": "rocket.cat",
18                 "name": "Rocket Cat"
19             },
20             "_updatedAt": "2018-03-08T14:47:37.003Z",
21             "instanceId": "uZG54xuoKauKHykbQ",
22             "etag": "jPavis9qG22xC5sDC",
23             "path": "/ufs/GridFS:Uploads/S78TNnvaWGwdYRaCD/images.jpeg",
24             "token": "28cAb868d9",
25             "uploadedAt": "2018-03-08T14:47:37.295Z",
26             "url": "/ufs/GridFS:Uploads/S78TNnvaWGwdYRaCD/images.jpeg"
27         }
28     ],
29     "count": 1,
30     "offset": 0,
31     "total": 1,
32     "success": true
33 }
```

Change Log

Version	Description
0.64.0	Change <code>userId</code> to <code>user</code> object in response

0.59.0

Added

Groups Roles

Lists all user's roles in the private group.

URL	Requires Auth	HTTP Method
/api/v1/groups.roles	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required(if no roomName)	The group's id
roomName	general	Required(if no roomId)	The group's name

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/groups.roles?roomId=ByehQjC44FwMeiLbX
```

Example Result

```
1 {
2   "roles": [
3     {
4       "rid": "BaE62jfDLXK3Xo6BA",
```

```
5         "u": {
6             "_id": "BkNkw3iKgNyhMbPyW",
7             "username": "ronnie.dio",
8             "name": "Ronnie James Dio"
9         },
10        "roles": [
11            "moderator"
12        ],
13        "_id": "ehPuGyZBedznJsQHp"
14    }
15 ],
16 "success": true
17 }
```

Change Log

Version	Description
0.65.0	Added

Invite

Url	Short Description	Details Page
/api/v1/findOrCreateInvite	Created a new Invite or returns an existing one with the same parameters.	Link
/api/v1/listInvites	Lists all of the invite tokens.	Link
/api/v1/removeInvite	Removes an invite.	Link
/api/v1/useInviteToken	Report to the server that an invite token was used.	Link
/api/v1/validateInviteToken	Checks if an invite token is valid.	Link

Find or Create Invite

Creates or return an existing invite with the specified parameters. Requires the `create-invite-links` permission.

URL	Requires Auth	HTTP Method
/api/v1/findOrCreateInvite	yes	POST

Payload

Argument	Example	Required	Description
rid	'csFYrDeixJG7PnrAh'	Required	The id of the room that the invite is being created for.
days	0	Required	The number of days that the invite will be valid for. Send 0 for unlimited.
maxUses	20	Required	The number of times that the invite can be used. Send 0 for unlimited.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -H "Content-type: application/json" \
4     http://localhost:3000/api/v1/findOrCreateInvite \
5     -d '{ "rid": "csFYrDeixJG7PnrAh", "days": 10, "maxUses": 0 }'
```

Example Result

```
1  {
2    "_id": "kDKQ3H",
3    "days": 10,
4    "maxUses": 0,
5    "rid": "csFYrDeixJG7PnrAh",
6    "userId": "sNiDqHit5nGAGFg8X",
7    "createdAt": "2017-01-06T13:23:46.018Z",
8    "expires": "2017-01-16T13:23:46.018Z",
9    "_updatedAt": "2017-01-06T13:23:46.018Z",
10   "uses": 0,
11   "url": "http://open.rocket.chat/invite/kDKQ3H",
12   "success": true
13 }
```

List Invites

Lists all of the invites on the server. Requires the `create-invite-links` permission.

URL	Requires Auth	HTTP Method
<code>/api/v1/listInvites</code>	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -H "Content-type: application/json" \
4     http://localhost:3000/api/v1/listInvites
```

Example Result

```
1 [
2   {
3     "_id": "kDKQ3H",
4     "days": 1,
5     "maxUses": 1,
6     "rid": "s7CE842q9WtvLctD7",
7     "userId": "sNiDqHit5nGAGFg8X",
8     "createdAt": "2019-12-20T03:31:56.774Z",
9     "expires": "2019-12-21T03:31:56.774Z",
10    "uses": 1,
11    "_updatedAt": "2019-12-20T03:33:40.065Z"
12  },
13  {
14    "_id": "99ScEP",
15    "days": 1,
16    "maxUses": 0,
17    "rid": "s7CE842q9WtvLctD7",
```

```
18     "userId": "sNiDqHit5nGAGFg8X",
19     "createdAt": "2019-12-20T03:38:20.485Z",
20     "expires": "2019-12-21T03:38:20.485Z",
21     "uses": 0,
22     "_updatedAt": "2019-12-20T03:38:20.485Z"
23   },
24   {
25     "_id": "Y5JKM4",
26     "days": 0,
27     "maxUses": 1,
28     "rid": "s7CE842q9WtvLctD7",
29     "userId": "sNiDqHit5nGAGFg8X",
30     "createdAt": "2019-12-20T03:38:25.313Z",
31     "expires": null,
32     "uses": 1,
33     "_updatedAt": "2019-12-20T03:38:40.116Z"
34   }
35 ]
```

Remove Invite

Removes an invite from the server. Requires the `create-invite-links` permission.

URL	Requires Auth	HTTP Method
<code>/api/v1/removeInvite</code>	yes	DELETE

Payload

Argument	Example	Required	Description
<code>_id</code>	<code>'kDKQ3H'</code>	Required	The id of the invite to remove.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -X DELETE \
4   http://localhost:3000/api/v1/removeInvite/kDKQ3H
```

Example Result

```
true
```

Use Invite Token

Report to the server that an invite token was used. No permissions are required.

URL	Requires Auth	HTTP Method
/api/v1/useInviteToken	yes	POST

Payload

Argument	Example	Required	Description
token	iS7pza	Required	The id of the invite being used.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/useInviteToken \
5   -d '{ "token": "iS7pza" }'
```

Example Result

```
1 {
2   "room": {
3     "rid": "QqsWiqNH2TnxtvZQq",
4     "fname": "a-room-name",
5     "name": "a-room-name",
```

```
6           "t": "p"
7     },
8   "success": true
9 }
```

Validate Invite Token

Checks if an invite token is valid. No permissions are required.

URL	Requires Auth	HTTP Method
/api/v1/validateInviteToken	no	POST

Payload

Argument	Example	Required	Description
token	iS7pza	Required	The id of the invite being validated.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/validateInviteToken \
5   -d '{ "token": "iS7pza" }'
```

Example Result

```
1 {
2   "valid": true,
3   "success": true
4 }
```



Integration

Url	Short Description	Details Page
/api/v1/integrations.create	Creates an integration.	Link
/api/v1/integrations.get	Gets an integration.	Link
/api/v1/integrations.history	Lists all history of the specified integration.	Link
/api/v1/integrations.list	Lists all of the integrations.	Link
/api/v1/integrations.remove	Removes an integration.	Link

Create

Creates an integration, if the callee has the permissions. Requires `manage-incoming-integrations` AND `manage-own-incoming-integrations` permissions to be able to create incoming integrations and `manage-outgoing-integrations` AND `manage-own-outgoing-integrations` to be able to create outgoing integrations.

URL	Requires Auth	HTTP Method
/api/v1/integrations.create	yes	POST

Payload

Argument	Example	Required	Description
<code>type</code>	<code>webhook-outgoing</code>	Required	The type of integration to create, <code>webhook-outgoing</code> and <code>webhook-ingress</code> are supported
<code>name</code>	Guggy	Required	The name of the integration, or show on the Adminstration
<code>enabled</code>	true	Required	Whether this integration should be enabled or no

event	sendMessage	Required	This field is required for outgoing integration. The event, can be one of these: sendMessage, fileUpload, roomArchive, roomCreated, roomJoined, roomLeft, userCreated.
username	rocket.cat	Required	The username to post this message as.
urls	['http://text2gif.guggy.com/guggify']	Required	The urls to call whenever this integration is triggered.
scriptEnabled	false	Required	Whether the script should be enabled.
channel	#general	Required	The channel, where the @username . command should be sent. Options: all_public_channels, all_private_channels, or all_direct_channels. Comma separated if more than one.
triggerWords	!guggy	Optional Default: ''	Specific words separated by commas which should trigger this integration.

alias	Guggy	Optional Default: ''	The alias which will be applied to messages where integration is processed.
avatar	http://res.guggy.com/logo_128.png	Optional Default: ''	The logo to apply to the messages where integration is triggered.
emoji	:ghost:	Optional Default: ''	The emoji which should be displayed as the avatar for messages from the integration.
token	8DFS89DMKLWEN	Optional Default: ''	If your integration requires a specific token from the service (API key), use this.
script	Integrations	Optional	Script triggered by this integration when triggered.

Example Call

```

1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/integrations.create \
5   -d '{ "type": "webhook-outgoing", "name": "Testing via REST API", "ev

```

Example Result

```
1  {
2      "integration": {
3          "type": "webhook-outgoing",
4          "name": "Testing via REST API",
5          "enabled": false,
6          "username": "rocket.cat",
7          "event": "sendMessage",
8          "urls": [
9              "http://text2gif.guggy.com/guggify"
10         ],
11         "scriptEnabled": false,
12         "userId": "rocket.cat",
13         "channel": [],
14         "_createdAt": "2017-01-06T13:23:46.018Z",
15         "_createdBy": {
16             "username": "graywolf336",
17             "_id": "aobEdbYhXfu5hkeqG"
18         },
19         "_updatedAt": "2017-01-06T13:23:46.018Z",
20         "_id": "3aaazpZ2WzoBP8msi9"
21     },
22     "success": true
23 }
```

Change Log

Version	Description
1.1.0	Separate permissions in <code>incoming</code> and <code>outgoing</code> .
0.49.0	Added

Get

Retrieves an integration by id.

URL	Requires Auth	HTTP Method
/api/v1/integrations.get	yes	GET

Query Parameters

Argument	Example	Required	Description
integrationId	ByehQjC44FwMeiLbX	Required	The integration's id
createdBy	fn5u45ui34fvb	Optional	The user's id who created the integration

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/integrations.get?integrationId=ByehQjC44
```

Example Result

```
1 {
2   "history": [
3     {
```

```
4      "_id": "f2sd51f65sd1f65sd1fs",
5      "type": "outgoing-webhook",
6      "step": "after-http-call",
7      "integration": {
8          "event": "roomJoined",
9          "enabled": true,
10         "username": "rocket.cat",
11         "channel": [
12             "#general"
13         ],
14         "name": "testeytyrtyrtyrtyer",
15         "triggerWords": [
16             "!webhhok"
17         ],
18         "urls": [
19             "http://localhost:3000/admin/integrations/outgoing"
20         ],
21     },
22     "event": "roomJoined",
23     "_createdAt": "2019-11-29T14:15:27.518Z",
24     "_updatedAt": "2019-11-29T14:15:27.606Z",
25     "data": {
26         "token": "xMxa3o95xzJSptyEjz4Z72LJ",
27         "bot": false,
28         "timestamp": "2019-11-29T14:15:27.522Z",
29         "channel_id": "GENERAL",
30         "channel_name": "general",
31         "user_id": "8pdja8tTb2s2XbvkQ",
32         "user_name": "dsds",
33         "user": {
34             "_id": "8pdja8tTb2s2XbvkQ",
35             "createdAt": "2019-11-29T14:14:36.936Z",
36             "username": "dsds",
37             "emails": [
38                 {
39                     "address": "ddsds@d.com",
40                     "verified": false
41                 }
42             ],
43             "type": "user",
44             "status": "offline",
45             "active": true,
46             "_updatedAt": "2019-11-29T14:15:15.677Z",
47             "roles": [
48                 "user"
49             ],
50             "name": "dsdsds",
51             "requirePasswordChange": true,
52             "settings": {}
53         },
54         "room": {
```

```

55         "_id": "GENERAL",
56         "ts": "2019-10-17T22:56:31.861Z",
57         "t": "c",
58         "name": "general",
59         "usernames": [],
60         "msgs": 125,
61         "usersCount": 45,
62         "default": true,
63         "_updatedAt": "2019-11-29T14:13:14.752Z",
64         "lastMessage": {
65             "_id": "6CiNGq8FPEYCfceNL",
66             "rid": "GENERAL",
67             "msg": "!webhook",
68             "ts": "2019-11-29T14:13:14.645Z",
69             "u": {
70                 "_id": "CaevAPDbXN252kyXa",
71                 "username": "marcos.defendi",
72                 "name": "Marcos Defendi"
73             },
74             "_updatedAt": "2019-11-29T14:13:14.705Z",
75             "mentions": [],
76             "channels": []
77         },
78         "lm": "2019-11-29T14:13:14.645Z",
79         "jitsiTimeout": "2019-10-22T19:20:28.404Z"
80     }
81 },
82     "url": "http://localhost:3000/admin/integrations/outgoing",
83 },
84 ],
85     "success": true
86 }

```

Change Log

Version	Description
2.4.0	Added

History

Lists all history of the specified integration. Requires the permission

`manage-outgoing-integrations` or `manage-own-outgoing-integrations`. It will return the integrations based on user permission. It supports the [Offset](#), [Count](#), and [Sort Query Parameters](#) along with [Query](#) and [Fields Query Parameters](#).

URL	Requires Auth	HTTP Method
<code>/api/v1/integrations.history</code>	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/integrations.history?id=nvdQuJQ6tE9HRFBz
```

Example Result

```
1 {
2   "history": [],
3   "offset": 0,
4   "items": 0,
5   "total": 0,
6   "success": true
7 }
```

Change Log

Version	Description
1.1.0	Separate permissions in <code>incoming</code> and <code>outgoing</code> .
0.53.0	Added

List

Lists all of the integrations on the server. Requires at least one integration permission:

`manage-incoming-integrations` , `manage-own-incoming-integrations` ,
`manage-outgoing-integrations` or `manage-own-outgoing-integrations` . It will return
the integrations based on the user permission. It supports the [Offset](#), [Count](#), and [Sort Query Parameters](#) along with [Query and Fields Query Parameters](#).

URL	Requires Auth	HTTP Method
<code>/api/v1/integrations.list</code>	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -H "Content-type: application/json" \
4     http://localhost:3000/api/v1/integrations.list
```

Example Result

```
1 {
2     "integrations": [
3         {
4             "_id": "WMQDChpnYTRmFre9h",
5             "enabled": true,
6             "username": "rocket.cat",
7             "alias": "Guggy",
8             "avatar": "https://image.crisp.im/avatar/website/17651a90-e082",
9             "name": "Guggy",
10            "triggerWords": [
11                "!guggy",
```

```
12             "guggy",
13             "gif+"
14         ],
15         "urls": [
16             "http://text2gif.guggy.com/guggify"
17         ],
18         "token": "aobEdbYhXfu5hkeqG",
19         "script": "...",
20         "scriptEnabled": true,
21         "impersonateUser": false,
22         "scriptCompiled": "...",
23         "scriptError": null,
24         "type": "webhook-outgoing",
25         "userId": "rocket.cat",
26         "channel": [],
27         "_createdAt": "2017-01-05T17:06:05.660Z",
28         "_createdBy": {
29             "username": "graywolf336",
30             "_id": "R4jgcQaQhvvK6K3iY"
31         },
32         "_updatedAt": "2017-01-05T17:06:05.660Z"
33     },
34     {
35         "_id": "3aaazpZ2WzoBP8msi9",
36         "type": "webhook-outgoing",
37         "name": "Testing via REST API",
38         "enabled": false,
39         "username": "rocket.cat",
40         "urls": [
41             "http://text2gif.guggy.com/guggify"
42         ],
43         "scriptEnabled": false,
44         "userId": "rocket.cat",
45         "channel": [],
46         "_createdAt": "2017-01-06T13:23:46.018Z",
47         "_createdBy": {
48             "username": "graywolf336",
49             "_id": "R4jgcQaQhvvK6K3iY"
50         },
51         "_updatedAt": "2017-01-06T13:23:46.018Z"
52     }
53 ],
54 "offset": 0,
55 "items": 2,
56 "total": 2,
57 "success": true
58 }
```

Change Log

Version	Description
1.1.0	Separate permissions in <code>incoming</code> and <code>outgoing</code> .
0.49.0	Added

Remove

Removes an integration from the server. Requires `manage-incoming-integrations` or `manage-own-incoming-integrations` permissions to be able to remove incoming integrations and `manage-outgoing-integrations` or `manage-own-outgoing-integrations` to be able to remove outgoing integrations.

URL	Requires Auth	HTTP Method
<code>/api/v1/integrations.remove</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>type</code>	<code>webhook-outgoing</code>	Required	The type of integration to remove, <code>webhook-outgoing</code> and <code>webhook-incoming</code> are supported.
<code>integrationId</code>	<code>oNLthAt9RwMw39N2B</code>	Required	The id of the integration to remove.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -H "Content-type: application/json" \
4     http://localhost:3000/api/v1/integrations.remove \
5     -d '{ "type": "webhook-outgoing", "integrationId": "oNLthAt9RwMw39N2B"
```

Example Result

```
1  {
2      "integration": {
3          "_id": "oNLthAt9RwMw39N2B",
4          "type": "webhook-outgoing",
5          "name": "Testing via REST API",
6          "enabled": false,
7          "username": "rocket.cat",
8          "urls": [
9              "http://text2gif.guggy.com/guggify"
10         ],
11         "scriptEnabled": false,
12         "userId": "rocket.cat",
13         "channel": [],
14         "_createdAt": "2017-01-06T13:42:14.143Z",
15         "_createdBy": {
16             "username": "graywolf336",
17             "_id": "R4jgcQaQhvK6K3iY"
18         },
19         "_updatedAt": "2017-01-06T13:42:14.144Z"
20     },
21     "success": true
22 }
```

Change Log

Version	Description
1.1.0	Separate permissions in <code>incoming</code> and <code>outgoing</code> .
0.49.0	Added

IM

Url	Short Description	Details Page
/api/v1/im.close	Removes a direct message from the list of direct messages.	Link
/api/v1/im.counters	Gets counters of direct messages.	Link
/api/v1/im.create	Create a direct message session with another user.	Link
/api/v1/im.history	Retrieves the messages from a direct message.	Link
/api/v1/im.files	Retrieves a list of files from a direct message.	Link
/api/v1/im.members	Retrieves the users of participants of a direct message.	Link
/api/v1/im.messages	Retrieves the messages from specific direct message.	Link
/api/v1/im.messages.others	Retrieves the messages from any direct message in the server.	Link
/api/v1/im.list	List the direct messages the caller is part of.	Link
/api/v1/im.list.everyone	List all direct message the caller in the server.	Link
/api/v1/im.open	Adds the direct message back to the list of direct messages.	Link
/api/v1/im.setTopic	Sets a direct message topic.	Link

Notes

From version 0.50.0 and on you can call the methods using `dm` instead of `im`.

Close

Removes the direct message from the user's list of direct messages.

URL	Requires Auth	HTTP Method
/api/v1/im.close	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The direct message id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/im.close \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.48.0	Added

Counters

Gets counters of direct messages.

URL	Requires Auth	HTTP Method
/api/v1/im.counters	yes	GET

Payload

Argument	Example	Required	Description
roomId	RtycPC29hqLJfT9xjew28FnZqipDpvKw3R	Required	The id of Direct Messages' Room
userId	RtycPC29hqLJfT9xj	Optional	Counters for provided user id (need to have a view-room-administration right for calling user)

Example Call

```
1 curl -H "X-Auth-Token: 8h2mKAwxB3AQrFSjLVKMooJyjdCFaA7W45sWlHP8Iz0" \
2   -H "X-User-Id: ew28FnZqipDpvKw3R" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/im.counters?roomId=RtycPC29hqLJfT9xjew28
```

Example Result

```
1  {
2    "joined":true,
3    "members":2,
4    "unread":0,
5    "unreadFrom":"2018-02-21T21:08:51.026Z",
6    "msgs":0,
7    "latest":"2018-02-21T21:08:51.026Z",
8    "userMentions":0,
9    "success":true
10 }
11
12 where:
13
14 joined - boolean flag that shows that user is joined the room or not
15 members - count of current room members
16 unread - count of unread messages for specified user (calling user or pro
17 unreadFrom - start date-time of unread interval for specified user
18 msgs - count of messages in the room
19 latest - end date-time of unread interval for specified user (or date-time
20 userMentions - count of user mentions in messages
```

Change Log

Version	Description
0.65.0	Added

Create

Create a direct message session with another user.

URL	Requires Auth	HTTP Method
/api/v1/im.create	yes	POST

Payload

Argument	Example	Required	Description
username	rocket.cat	yes	The username to open a direct message session, required if <code>usernames</code> is not provided
usernames	rocket.cat, anotheruser	no	List of usernames to open a multiple direct message session, required if <code>username</code> is not provided

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/im.create \
5   -d '{ "username": "rocket.cat" }'
```

Example Result

```
1  {
2      "room": {
3          "t": "d",
4          "rid": "PMrDaS4axRqkjY7errocket.cat",
5          "usernames": [
6              "g1",
7              "rocket.cat"
8          ]
9      },
10     "success": true
11 }
```

Change Log

Version	Description
0.59.0	Added
3.1.0	Added usernames fields

Files

Retrieves the files from a direct message. It supports the [Offset, Count, and Sort Query Parameters](#) along with [Query and Fields Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/im.files	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX		The room id.
username	user	Required	The user name.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/im.files?roomId=ByehQjC44FwMeiLbX
```

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/im.files?username=user
```

Example Result

```
1  {
2      "files": [
3          {
4              "_id": "zchmBCtCEGeGRFrFw",
5              "name": "images.jpeg",
6              "size": 9778,
7              "type": "image/jpeg",
8              "rid": "ksKsKmrjvxzkzxkwzFWhJbub88A9mLrAD",
9              "description": "dafsd",
10             "store": "GridFS:Uploads",
11             "complete": true,
12             "uploading": false,
13             "extension": "jpeg",
14             "progress": 1,
15             "user": {
16                 "_id": "ksKsKmrjvxzkzxkw",
17                 "username": "rocket.cat",
18                 "name": "Rocket Cat"
19             },
20             "_updatedAt": "2018-03-08T17:36:09.846Z",
21             "instanceId": "uZG54xuoKauKHykbQ",
22             "etag": "mnqbLcmKAptJBEKrx",
23             "path": "/ufs/GridFS:Uploads/zchmBCtCEGeGRFrFw/images.jpeg",
24             "token": "787868e89B",
25             "uploadedAt": "2018-03-08T17:36:09.913Z",
26             "url": "/ufs/GridFS:Uploads/zchmBCtCEGeGRFrFw/images.jpeg"
27         }
28     ],
29     "count": 1,
30     "offset": 0,
31     "total": 1,
32     "success": true
33 }
```

Change Log

Version	Description
0.64.0	Change <code>userId</code> to <code>user</code> object in response

0.59.0

Added

History

Retrieves the messages from a direct message.

URL	Requires Auth	HTTP Method
/api/v1/im.history	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The direct message id
latest	2016-09-30T13:42:25.304Z	Optional Default: now	The end of time range of messages to retrieve
oldest	2016-05-30T13:42:25.304Z	Optional Default: n/a	The start of the time range of messages to retrieve
inclusive	true	Optional Default: false	Whether messages which land on latest and oldest should be included
offset	10	Optional Default: 0	The offset of the list messages to retrieve
count	100	Optional Default: 20	The amount of messages to retrieve
unread	false	Optional Default: false	Whether the amount of unread should be included.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/im.history?roomId=ByehQjC44FwMeiLbX
```

Example Result

```
1 {
2   "messages": [
3     {
4       "_id": "AkzpHAvZpdnuchw2a",
5       "rid": "ByehQjC44FwMeiLbX",
6       "msg": "hi",
7       "ts": "2016-12-09T12:50:51.555Z",
8       "u": {
9         "_id": "y65tAmHs93aDChMWu",
10        "username": "testing"
11      },
12      "_updatedAt": "2016-12-09T12:50:51.562Z"
13    },
14    {
15      "_id": "vkLMxccR4MuTxreF",
16      "t": "uj",
17      "rid": "ByehQjC44FwMeiLbX",
18      "ts": "2016-12-08T15:41:37.730Z",
19      "msg": "testing2",
20      "u": {
21        "_id": "bRtgdhzM6PD9F8pSx",
22        "username": "testing2"
23      },
24      "groupable": false,
25      "_updatedAt": "2016-12-08T16:03:25.235Z"
26    },
27    {
28      "_id": "bfRW658nEyEBg75rc",
29      "t": "uj",
30      "rid": "ByehQjC44FwMeiLbX",
31      "ts": "2016-12-07T15:47:49.099Z",
32      "msg": "testing",
33      "u": {
```

```
34         "_id": "nSYqWzZ4GsKTX4dyK",
35         "username": "testing1"
36     },
37     "groupable": false,
38     "_updatedAt": "2016-12-07T15:47:49.099Z"
39   },
40   {
41     "_id": "pbuFiGadhRZTKouhB",
42     "t": "uj",
43     "rid": "ByehQjC44FwMeiLbX",
44     "ts": "2016-12-06T17:57:38.635Z",
45     "msg": "testing",
46     "u": {
47       "_id": "y65tAmHs93aDChMWu",
48       "username": "testing"
49     },
50     "groupable": false,
51     "_updatedAt": "2016-12-06T17:57:38.635Z"
52   }
53 ],
54   "success": true
55 }
```

Change Log

Version	Description
0.75.0	Added <code>offset</code> property
0.48.0	Added

List Everyone

Lists all of the direct messages in the server, requires the permission

`view-room-administration` permission. It supports the [Offset](#), [Count](#), and [Sort Query](#)

Parameters along with [Query and Fields Query Parameters](#).

URL	Requires Auth	HTTP Method
<code>/api/v1/im.list.everyone</code>	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/im.list.everyone
```

Example Result

```
1 {
2     "ims": [
3         {
4             "_id": "ByehQjC44FwMeiLbX",
5             "name": "test-test",
6             "t": "p",
7             "usernames": [
8                 "testing1"
9             ],
10            "msgs": 0,
11            "u": {
12                "_id": "aobEdbYhXfu5hkeqG",
13                "username": "testing1"
14            },
15            "ts": "2016-12-09T15:08:58.042Z",
16        }
17    ]
18}
```

```
16         "ro": false,
17         "sysMes": true,
18         "_updatedAt": "2016-12-09T15:22:40.656Z"
19     },
20     {
21         "_id": "t7qapfhZjANMRAi5w",
22         "name": "testing",
23         "t": "p",
24         "usernames": [
25             "testing2"
26         ],
27         "msgs": 0,
28         "u": {
29             "_id": "y65tAmHs93aDChMwu",
30             "username": "testing2"
31         },
32         "ts": "2016-12-01T15:08:58.042Z",
33         "ro": false,
34         "sysMes": true,
35         "_updatedAt": "2016-12-09T15:22:40.656Z"
36     }
37 ],
38 "offset":0,
39 "count":2,
40 "total":2,
41 "success": true
42 }
```

Change Log

Version	Description
0.49.0	Added

List

Lists all of the direct messages the calling user has joined. It supports the [Offset](#), [Count](#), and [Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/im.list	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 8h2mKAwxB3AQrFSjLVKMooJyjdCFaA7W45sWlHP8Iz0" \
2     -H "X-User-Id: ew28FnZqipDpvKw3R" \
3     http://localhost:3000/api/v1/im.list
```

Example Result

```
1 [
2     "ims": [
3         {
4             "_id": "ew28FnZqipDpvKw3Rrocket.cat",
5             "_updatedAt": "2018-02-23T17:58:56.147Z",
6             "t": "d",
7             "msgs": 22,
8             "ts": "2018-02-18T19:51:52.557Z",
9             "lm": "2018-02-23T17:58:56.136Z",
10            "topic": "a direct message with rocket.cat"
11        },
12        {
13            "_id": "RtycPC29hqLJfT9xjew28FnZqipDpvKw3R",
14            "_updatedAt": "2018-02-23T18:14:03.510Z",
15            "t": "d",
16            "msgs": 2,
17            "ts": "2018-02-21T21:08:51.026Z",
18        }
19    ]
20]
```

```
18         "lm":"2018-02-23T18:14:03.490Z",
19         "username":"rocketchat.internal.admin.test"
20     },
21     {
22         "_id":"ew28FnZqipDpvKw3Rf2CAhYGtjS9iNZ7nd",
23         "_updatedAt":"2018-02-23T17:45:56.496Z",
24         "t":"d",
25         "msgs":1,
26         "ts":"2018-02-23T17:32:28.016Z",
27         "lm":"2018-02-23T17:45:56.475Z"
28     }
29 ],
30 "offset":0,
31 "count":3,
32 "total":3,
33 "success":true
34 }
```

Change Log

Version	Description
0.67.0	Remove 'query' parameter support.
0.62.0	Add 'query' parameter support.
0.49.0	Count and offset query parameters supported.
0.48.0	Added

Members

Lists the users of participants of a direct message. It supports the [Offset, Count, and Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/im.members	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX		The room id.
username	user	Required	The user name.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/im.members?roomId=ByehQjC44FwMeiLbX
```

Example Result

```
1 {
2     "members": [
3         {
4             "_id": "Q4GkX6RMepGDdQ7YJ",
```

```
5         "status": "online",
6         "name": "Marcos Defendi",
7         "utcOffset": -3,
8         "username": "marcos.defendi"
9     },
10    {
11        "_id": "rocket.cat",
12        "name": "Rocket.Cat",
13        "username": "rocket.cat",
14        "status": "online",
15        "utcOffset": 0
16    }
17 ],
18 "count": 2,
19 "offset": 0,
20 "total": 2,
21 "success": true
22 }
```

Change Log

Version	Description
0.59.0	Added

Messages Others

Retrieves the messages from any direct message in the server. It supports the [Offset](#), [Count](#), and [Sort Query Parameters](#) along with [Query and Fields Query Parameter](#).

For this method to work, the `Enable Direct Message History Endpoint` setting (from the admin panel: `General > REST API > Enable Direct Message History Endpoint`) must be set to true, and the user calling this method must have the `view-room-administration` permission.

URL	Requires Auth	HTTP Method
/api/v1/im.messages.others	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The direct message id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/im.messages.others?roomId=ByehQjC44FwMei
```

Example Result

```
1  {
2      "messages": [
3          {
4              "_id": "AkzpHAvZpdnuchw2a",
5              "rid": "ByehQjC44FwMeiLbX",
6              "msg": "hi",
7              "ts": "2016-12-09T12:50:51.555Z",
8              "u": {
9                  "_id": "y65tAmHs93aDChMwu",
10                 "username": "testing"
11             },
12             "_updatedAt": "2016-12-09T12:50:51.562Z"
13         },
14         {
15             "_id": "vkLMxcctR4MuTxreF",
16             "t": "uj",
17             "rid": "ByehQjC44FwMeiLbX",
18             "ts": "2016-12-08T15:41:37.730Z",
19             "msg": "testing2",
20             "u": {
21                 "_id": "bRtgdhzM6PD9F8pSx",
22                 "username": "testing2"
23             },
24             "groupable": false,
25             "_updatedAt": "2016-12-08T16:03:25.235Z"
26         },
27         {
28             "_id": "bfRW658nEyEBg75rc",
29             "t": "uj",
30             "rid": "ByehQjC44FwMeiLbX",
31             "ts": "2016-12-07T15:47:49.099Z",
32             "msg": "testing",
33             "u": {
34                 "_id": "nSYqWzZ4GsKTX4dyK",
35                 "username": "testing1"
36             },
37             "groupable": false,
38             "_updatedAt": "2016-12-07T15:47:49.099Z"
39         },
40         {
41             "_id": "pbuFiGadhRZTKouhB",
42             "t": "uj",
43             "rid": "ByehQjC44FwMeiLbX",
44             "ts": "2016-12-06T17:57:38.635Z",
45             "msg": "testing",
46             "u": {
47                 "_id": "y65tAmHs93aDChMwu",
48                 "username": "testing"
49             },
50         }
51     ]
52 }
```

```
50      "groupable": false,
51      "_updatedAt": "2016-12-06T17:57:38.635Z"
52    }
53  ],
54  "success": true
55 }
```

Change Log

Version	Description
0.50.0	Added

Messages

Lists all of the specific direct message on the server. It supports the **Offset, Count, and Sort Query Parameters** along with **Query and Fields Query Parameters**.

URL	Requires Auth	HTTP Method
/api/v1/im.messages	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required(if no username)	The channel's id
username	rocket.cat	Required(if no roomId)	The username of DM

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/im.messages?roomId=ByehQjC44FwMeiLbX
```

Example Result

```
1 {
2     "messages": [
3         {
```

```
4         "_id": "xadufzmxzYQp4H9py",
5         "alias": "test",
6         "msg": "Example message",
7         "attachments": [],
8         "parseUrls": true,
9         "bot": {
10             "i": "MnQyfhWt5LqZotyfc"
11         },
12         "groupable": false,
13         "ts": "2018-10-05T01:10:47.524Z",
14         "u": {
15             "_id": "rocket.cat",
16             "username": "rocket.cat",
17             "name": "Rocket.Cat"
18         },
19         "rid": "GENERAL",
20         "_updatedAt": "2018-10-05T13:42:51.163Z",
21         "reactions": {
22             ":grin:": {
23                 "usernames": [
24                     "marcos.defendi"
25                 ]
26             }
27         },
28         "mentions": [],
29         "channels": [],
30         "starred": {
31             "_id": "KPkEYwKKBKZnEEPpt"
32         }
33     },
34     {
35         "_id": "id-1538701845987",
36         "rid": "GENERAL",
37         "msg": "This message was edited via API",
38         "alias": "Gruggy",
39         "emoji": ":smirk:",
40         "avatar": "http://res.guggy.com/logo_128.png",
41         "attachments": [
42             {
43                 "collapsed": false,
44                 "color": "#ff0000",
45                 "text": "Yay for gruggy!",
46                 "ts": "2016-12-09T16:53:06.761Z",
47                 "message_link": "https://google.com",
48                 "thumb_url": "http://res.guggy.com/logo_128.png",
49                 "author_name": "Bradley Hilton",
50                 "author_link": "https://rocket.chat/",
51                 "author_icon": "https://avatars.githubusercontent.com/",
52                 "title": "Attachment Example",
53                 "title_link": "https://youtube.com",
54                 "title_link_download": true,
```

```
55         "image_url": "http://res.guggy.com/logo_128.png",
56         "audio_url": "http://www.w3schools.com/tags/horse.mp3"
57         "video_url": "http://www.w3schools.com/tags/movie.mp4"
58     "fields": [
59         {
60             "short": true,
61             "title": "Test",
62             "value": "Testing out something or other"
63         },
64         {
65             "short": true,
66             "title": "Another Test",
67             "value": "[Link](https://google.com/) somethin"
68         }
69     ]
70 ],
71 "ts": "2018-10-05T01:10:45.994Z",
72 "u": {
73     "_id": "rocketchat.internal.admin.test",
74     "username": "rocketchat.internal.admin.test",
75     "name": "RocketChat Internal Admin Test"
76 },
77 "_updatedAt": "2018-10-05T01:10:47.064Z",
78 "editedBy": {
79     "_id": "rocketchat.internal.admin.test",
80     "username": "rocketchat.internal.admin.test"
81 },
82 "editedAt": "2018-10-05T01:10:46.384Z",
83 "reactions": {
84     ":smile:": {
85         "usernames": [
86             "rocketchat.internal.admin.test"
87         ]
88     },
89     ":squid:": {
90         "usernames": [
91             "rocketchat.internal.admin.test"
92         ]
93     },
94     ":bee:": {
95         "usernames": [
96             "rocketchat.internal.admin.test"
97         ]
98     },
99     ":ant:": {
100         "usernames": [
101             "rocketchat.internal.admin.test"
102         ]
103     }
104 },
105 }
```

```
106         "mentions": [],
107         "channels": [],
108         "urls": []
109     },
110 ],
111     "count": 2,
112     "offset": 0,
113     "total": 2,
114     "success": true
115 }
```

Change Log

Version	Description
0.59.0	Added

Open

Adds the direct message back to the user's list of direct messages.

URL	Requires Auth	HTTP Method
/api/v1/im.open	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The direct message id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/im.open \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.48.0	Added

Set Topic

Sets the topic for the direct message.

URL	Requires Auth	HTTP Method
/api/v1/im.setTopic	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The direct message id
topic	Discuss all of the testing.	Required	The direct message topic to set.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/im.setTopic \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX", "topic": "Discuss all of the tes'
```

Example Result

```
1 {
```

```
2   "topic": "Testing out everything.",  
3   "success": true  
4 }
```

Change Log

Version	Description
0.48.0	Added

Omnichannel

Manage Livechat related data:

Url	HTTP Method	Short Description	Detail Page
/api/v1/livechat/users/:type	GET	Get a list of agents or managers.	info
/api/v1/livechat/users/:type	POST	Create a new omnichannel agent or manager.	info
/api/v1/livechat/users/:type/:_id	GET	Retrieve agent or manager data.	info
/api/v1/livechat/users/:type/:_id	DELETE	Removes an Omnichannel agent or manager.	info
/api/v1/livechat/department	GET	Get a list of Omnichannel departments.	info
/api/v1/livechat/department	POST	Creates a new Omnichannel department.	info

/api/v1/livechat/department/:_id	GET	Retrieve an Omnichannel department data.	info
/api/v1/livechat/department/:_id	PUT	Updates an Omnichannel department data.	info
/api/v1/livechat/department/:_id	DELETE	Delete an Omnichannel department.	info
/api/v1/livechat/inquiries.list	GET	Retrieves a list of open inquiries.	info
/api/v1/livechat/inquiries.take	POST	Take an open inquiry.	info
/api/v1/livechat/inquiries.getOne	GET	Get one inquiry by room id.	info
/api/v1/livechat/integrations.settings	GET	Retrieves a list of integration settings.	info
/api/v1/livechat/sms-incoming/:service	POST	Send SMS messages to Rocket.Chat.	info
/api/v1/livechat/agent.info/:rid/:token	GET	Retrieve the current Omnichannel agent data.	info

/api/v1/livechat/agent.next/:token	GET	Request the next Omnichannel agent available.
/api/v1/livechat/agents/:agentId/departments	GET	Get the agent departments.
/api/v1/livechat/config/:token	GET	Get basic Livechat widget configuration info and additional visitor data.
/api/v1/livechat/custom.field	POST	Send an Omnichannel custom field.
/api/v1/livechat/custom.fields	POST	Send an array of Omnichannel custom fields.
/api/v1/livechat/custom-fields	GET	Get a list of Omnichannel custom fields.
/api/v1/livechat/custom-fields/:_id	GET	Get an Omnichannel custom field.
/api/v1/livechat/message	POST	Send a new Omnichannel message.

/api/v1/livechat/message/:_id	PUT	Updates an Omnichannel message data.	info
/api/v1/livechat/message/:_id	DELETE	Delete an Omnichannel message.	info
/api/v1/livechat/messages.history/:rid	GET	Load Omnichannel messages history.	info
/api/v1/livechat/office-hours	GET	Get a list of office hours.	info
/api/v1/livechat/offline.message	POST	Send a new Omnichannel offline message.	info
/api/v1/livechat/rooms	GET	Retrieves a list of Omnichannel rooms.	info
/api/v1/livechat/queue	GET	Retrieves a list of queued chats.	info
/api/v1/livechat/room	GET	Get the Omnichannel room data or open a new room.	info

/api/v1/livechat/room.close	POST	Close an Omnichannel room. info
/api/v1/livechat/room.transfer	POST	Transfer an Omnichannel room to another agent or department. info
/api/v1/livechat/room.forward	POST	Allow Omnichannel Agents to forward an Omnichannel room to another agent, department, or return it back to the Queue. info
/api/v1/livechat/room.survey	POST	Send an Omnichannel survey to Rocket.Chat. info
/api/v1/livechat/transcript	POST	Request an Omnichannel transcript. info
/api/v1/livechat/triggers	GET	Get the Omnichannel triggers. info

/api/v1/livechat/triggers/:_id	GET	Get an Omnichannel info trigger.
/api/v1/livechat/video.call/:token	GET	Request a new video info call room.
/api/v1/livechat/visitor	POST	Register a new Omnichannel info visitor.
/api/v1/livechat/visitor/:token	GET	Retrieve a visitor's data. info
/api/v1/livechat/visitors.info	GET	Retrieve a visitor's info. info
/api/v1/livechat/visitors.search	GET	Search for Visitors. info
/api/v1/livechat/page.visited	POST	Send visitor navigation history to Rocket.Chat. info
/api/v1/livechat/appearance	GET	Get the settings about Livechat Widget Appearance. info
/api/v1/livechat/visitors.pagesVisited	GET	Gets the visitor navigation history. info

/api/v1/livechat/visitors.chatHistory/room/room-id/visitor/visitor-id	GET	Gets the visitor chat history.	info
---	-----	--------------------------------	----------------------

Dashboard

Agents

Url	Short Description	Details Page
/livechat/analytics/agents/average-service-time	agent average service time	Info
/livechat/analytics/agents/total-service-time	agent total service time	Info
/livechat/analytics/agents/available-for-service-history	##	Info

Departments

Url	Short Description	Details Page
/livechat/analytics/departments/ammount-of-chats	amount of chats	Info
/livechat/analytics/departments/average-service-time	average service time	[Link] (#)
/livechat/analytics/departments/average-chat-duration-time	average chats duration	Info

/livechat/analytics/departments/total-service-time	total service time	Info
/livechat/analytics/departments/average-waiting-time	avg waiting time	Info
/livechat/analytics/departments/total-transferred-chats	total transferred chats	Info
/livechat/analytics/departments/total-abandoned-chats	total abandoned chats	Info
/livechat/analytics/departments/percentage-abandoned-chats	percentage of abandoned chats	Info

Livechat

Url	Short Description	Details Page
/api/v1/livechat/inquiries.list	Retrieves a list of open inquiries.	Link
/api/v1/livechat/inquiries.take	Take an open inquiry.	Link
/api/v1/livechat/rooms	Retrieves a list of livechat rooms.	Link

Average chat duration by department

URL	Requires Auth	HTTP Method
api/v1/livechat/analytics/departments/average-chat-duration-time	YES	GET

Headers

Argument	Example	Required	Description
X-User-Id	myuser-name	Required	Your username hash (returned after you log in through the API)
X-Auth-Token	myauth-token	Required	Your token (returned after you log in through the API)

Parameters

Argument	Example	Required	Description
start	2020-09-09T00:11:22.345Z	Required	start date
end	2020-09-10T23:59:22.345Z	Required	end date

Notes

- ** The API will return a blank page if the correct headers are not sent

Example Call

```
1 curl --location --request GET 'http://localhost:3000/api/v1/livechat/analy
2 --header 'X-Auth-Token: myauth-token' \
3 --header 'X-User-Id: myuser-name'
```

Result

```
1 {
2   "departments": [],
3   "count": 0,
4   "offset": 0,
5   "total": 0,
6   "success": true
7 }
```

Change Log

Total service time by department

URL	Requires Auth	HTTP Method
api/v1/livechat/analytics/departments/total-service-time	YES	GET

Headers

Argument	Example	Required	Description
X-User-Id	myuser-name	Required	Your username hash (returned after you log in through the API)
X-Auth-Token	myauth-token	Required	Your token (returned after you log in through the API)

Parameters

Argument	Example	Required	Description
start	2020-09-09T00:11:22.345Z	Required	start date
end	2020-09-10T23:59:22.345Z	Required	end date

Notes

- ** The API will return a blank page if the correct headers are not sent

Example Call

```
1 curl --location --request GET 'http://localhost:3000/api/v1/livechat/analy
2 --header 'X-Auth-Token: myauth-token' \
3 --header 'X-User-Id: myuser-name'
```

Result

```
1 {
2   "departments": [],
3   "count": 0,
4   "offset": 0,
5   "total": 0,
6   "success": true
7 }
```

Change Log

Average waiting time by department

URL	Requires Auth	HTTP Method
api/v1/livechat/analytics/departments/average-waiting-time	YES	GET

Headers

Argument	Example	Required	Description
X-User-Id	myuser-name	Required	Your username hash (returned after you log in through the API)
X-Auth-Token	myauth-token	Required	Your token (returned after you log in through the API)

Parameters

Argument	Example	Required	Description
start	2020-09-09T00:11:22.345Z	Required	start date
end	2020-09-10T23:59:22.345Z	Required	end date

Notes

- ** The API will return a blank page if the correct headers are not sent

Example Call

```
1 curl --location --request GET 'http://localhost:3000/api/v1/livechat/analy
2 --header 'X-Auth-Token: myauth-token' \
3 --header 'X-User-Id: myuser-name'
```

Result

```
1 {
2   "departments": [],
3   "count": 0,
4   "offset": 0,
5   "total": 0,
6   "success": true
7 }
```

Change Log

Total of transferred chats by department

URL	Requires Auth	HTTP Method
api/v1/livechat/analytics/departments/total-transferred-chats	YES	GET

Headers

Argument	Example	Required	Description
X-User-Id	myuser-name	Required	Your username hash (returned after you log in through the API)
X-Auth-Token	myauth-token	Required	Your token (returned after you log in through the API)

Parameters

Argument	Example	Required	Description
start	2020-09-09T00:11:22.345Z	Required	start date
end	2020-09-10T23:59:22.345Z	Required	end date

Notes

- ** The API will return a blank page if the correct headers are not sent

Example Call

```
1 curl --location --request GET 'http://localhost:3000/api/v1/livechat/analy
2 --header 'X-Auth-Token: myauth-token' \
3 --header 'X-User-Id: myuser-name'
```

Result

```
1 {
2   "departments": [],
3   "count": 0,
4   "offset": 0,
5   "total": 0,
6   "success": true
7 }
```

Change Log

Agents total service time

URL	Requires Auth	HTTP Method
api/v1/livechat/analytics/agents/total-service-time	YES	GET

Headers

Argument	Example	Required	Description
X-User-Id	myuser-name	Required	Your username hash (returned after you log in through the API)
X-Auth-Token	myauth-token	Required	Your token (returned after you log in through the API)

Parameters

Argument	Example	Required	Description
start	2020-09-09T00:11:22.345Z	Required	start date
end	2020-09-10T23:59:22.345Z	Required	end date

Notes

- The API will return a blank page if the correct headers are not sent

Example Call

```
1 curl --location --request GET 'http://localhost:3000/api/v1/livechat/analy
2 --header 'X-Auth-Token: myauth-token' \
3 --header 'X-User-Id: myuser-name'
```

Result

```
1 {
2   "agents": [],
3   "count": 0,
4   "offset": 0,
5   "total": 0,
6   "success": true
7 }
```

Change Log

Agents average service time

URL	Requires Auth	HTTP Method
api/v1/livechat/analytics/agents/average-service-time	YES	GET

Headers

Argument	Example	Required	Description
X-User-Id	myuser-name	Required	Your username hash (returned after you log in through the API)
X-Auth-Token	myauth-token	Required	Your token (returned after you log in through the API)

Parameters

Argument	Example	Required	Description
start	2020-09-09T00:11:22.345Z	Required	start date
end	2020-09-10T23:59:22.345Z	Required	end date

Notes

- ** The API will return a blank page if the correct headers are not sent

Example Call

```
1 curl --location --request GET 'http://localhost:3000/api/v1/livechat/analy
2 --header 'X-Auth-Token: myauth-token' \
3 --header 'X-User-Id: myuser-name'
```

Result

```
1 {
2   "agents": [],
3   "count": 0,
4   "offset": 0,
5   "total": 0,
6   "success": true
7 }
```

Change Log

Total abandoned chats by department

URL	Requires Auth	HTTP Method
api/v1/livechat/analytics/departments/total-abandoned-chats	YES	GET

Headers

Argument	Example	Required	Description
X-User-Id	myuser-name	Required	Your username hash (returned after you log in through the API)
X-Auth-Token	myauth-token	Required	Your token (returned after you log in through the API)

Parameters

Argument	Example	Required	Description
start	2020-09-09T00:11:22.345Z	Required	start date
end	2020-09-10T23:59:22.345Z	Required	end date

Notes

- ** The API will return a blank page if the correct headers are not sent

Example Call

```
1 curl --location --request GET 'http://localhost:3000/api/v1/livechat/analy
2 --header 'X-Auth-Token: myauth-token' \
3 --header 'X-User-Id: myuser-name'
```

Result

```
1 {
2   "departments": [],
3   "count": 0,
4   "offset": 0,
5   "total": 0,
6   "success": true
7 }
```

Change Log

Percentage of abandoned chats by department

URL	Requires Auth	HTTP Method
api/v1/livechat/analytics/departments/percentage-abandoned-chats	YES	GET

Headers

Argument	Example	Required	Description
X-User-Id	myuser-name	Required	Your username hash (returned after you log in through the API)
X-Auth-Token	myauth-token	Required	Your token (returned after you log in through the API)

Parameters

Argument	Example	Required	Description
start	2020-09-09T00:11:22.345Z	Required	start date
end	2020-09-10T23:59:22.345Z	Required	end date

Notes

- ** The API will return a blank page if the correct headers are not sent

Example Call

```
1 curl --location --request GET 'http://localhost:3000//api/v1/livechat/anal
2 --header 'X-Auth-Token: myauth-token' \
3 --header 'X-User-Id: myuser-name'
```

Result

```
1 {
2   "departments": [],
3   "count": 0,
4   "offset": 0,
5   "total": 0,
6   "success": true
7 }
```

Change Log

Agents available for service time

URL	Requires Auth	HTTP Method
api/v1/livechat/analytics/agents/available-for-service-history	YES	GET

Headers

Argument	Example	Required	Description
X-User-Id	myuser-name	Required	Your username hash (returned after you log in through the API)
X-Auth-Token	myauth-token	Required	Your token (returned after you log in through the API)

Parameters

Argument	Example	Required	Description
start	2020-09-09T00:11:22.345Z	Required	start date
end	2020-09-10T23:59:22.345Z	Required	end date

Notes

- ** The API will return a blank page if the correct headers are not sent

Example Call

```
1 curl --location --request GET 'http://localhost:3000/api/v1/livechat/analy
2 --header 'X-Auth-Token: myauth-token' \
3 --header 'X-User-Id: myuser-name'
```

Result

```
1  {
2     "agents": [
3         {
4             "availableTimeInSeconds": 102117,
5             "username": "andres.mauricio"
6         },
7         {
8             "availableTimeInSeconds": 280,
9             "username": "bruna.martins"
10        },
11        {
12            "availableTimeInSeconds": 40242,
13            "username": "marina"
14        },
15        {
16            "availableTimeInSeconds": 1,
17            "username": "rogerio"
18        }
19    ],
20    "count": 4,
21    "offset": 0,
22    "total": 4,
23    "success": true
24 }
```

Change Log

Agent Info

Retrieve the current Livechat agent data.

URL	Requires Auth	HTTP Method
/api/v1/livechat/agent.info/:rid/:token	no	GET

Payload

Argument	Example	Required	Description
rid	zRAeTszXor8CCPceB	Required	The room <code>_id</code> .
token	iNKE8a6k6cjbjqWhWd	Required	The visitor <code>token</code> .

Example Call

```
curl http://localhost:3000/api/v1/livechat/agent.info/zRAeTszXor8CCPceB/iNKE
```

Example Result

```
1  {
2    "agent": {
3      "_id": "7Gm3PoFCJWTCJ68XR",
4      "emails": [
5        {
6          "address": "agent@rocket.chat",
```

```
7         "verified": true
8     }
9 ],
10    "name": "Livechat Agent",
11    "username": "livechat.agent"
12 },
13    "success": true
14 }
```

Change Log

Version	Description
0.70.0	Added

Request the next Livechat agent available.

URL	Requires Auth	HTTP Method
/api/v1/livechat/agent.next/:token	no	GET

Payload

Argument	Example	Required	Description
token	iNKE8a6k6cjbjqWhWd	Required	The visitor token .

Example Call

```
curl http://localhost:3000/api/v1/livechat/agent.next/iNKE8a6k6cjBqWhWd
```

Example Result

```
1  {
2    "agent": {
3      "_id": "7Gm3PoFCJWTCJ68XR",
4      "emails": [
5        {
6          "address": "agent@rocket.chat",
7          "verified": true
8        }
9      ],
10     "name": "Livechat Agent",
11     "username": "livechat.agent"
12   },
13   "success": true
14 }
```

Change Log

Version	Description
0.70.0	Added

Get agent departments.

URL	Requires Auth	HTTP Method
/api/v1/livechat/agents/:agentId/departments	yes	GET

Url params

Argument	Example	Required	Description
agentId	iNKE8a6k6cjbjqWhWd	Required	The agent Id.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/livechat/agents/iNKE8a6k6cjbjqWhWd/depart
```

Example Result

```
1 {
2     "departments": [
3         {
4             "_id": "7FKDYAFFQ42tBEizA",
5             "agentId": "bp9s6p76ocZ9QCkbz",
6             "departmentId": "KGvTnGdv7jZYWDK5D",
7             "_updatedAt": "2019-10-01T18:15:38.319Z",
8             "count": 0,
9             "order": 0,
10            "username": "marcos"
11        },
12        {
13            "_id": "hTY4cR3RrkZgZBX4T",
14            "agentId": "bp9s6p76ocZ9QCkbz",
15            "departmentId": "E3vjWSY3wNkW3unBo",
16            "_updatedAt": "2019-10-04T21:46:59.781Z",
17            "count": 0,
18            "order": 0,
19            "username": "marcos"
20        }
21    ],
22}
```

```
22      "success": true
23  }
```

Change Log

Version	Description
2.4.0	Added

Appearance

Get Livechat widget settings about appearance

URL	Requires Auth	HTTP Method
/api/v1/livechat/appearance	yes	GET

Example Call

```
curl http://localhost:3000/api/v1/livechat/appearance
```

Example Result

```
1  {
2      "appearance": [
3          {
4              "_id": "Livechat_conversation_finished_message",
5              "_updatedAt": "2019-10-04T17:39:10.786Z",
6              "autocomplete": true,
7              "blocked": false,
8              "createdAt": "2019-10-01T16:52:37.687Z",
9              "group": "Livechat",
10             "hidden": false,
11             "i18nDescription": "Livechat_conversation_finished_message_Des",
12             "i18nLabel": "Conversation_finished_message",
13             "packageValue": "",
14             "public": true,
15             "secret": false,
16             "sorter": 15,
17             "ts": "2019-10-01T17:09:23.894Z",
18             "type": "string",
19             "value": "",
20             "valueSource": "meteorSettingsValue",
```

```
21           "meteorSettingsValue": ""  
22       }  
23     // ...  
24   ],  
25   "success": true  
26 }
```

Change Log

Version	Description
2.2.0	Added

Canned Responses

Get a list of canned responses. It supports the [Offset, Count, and Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/canned-responses	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/canned-responses
```

Example Result

```
1 {
2   "cannedResponses": [
3     {
4       "_id": "QZXvWyo3niwWREZ4g",
5       "shortcut": "my-new-canned",
6       "text": "This is an example",
7       "scope": "global",
8       "tags": [
9         "tag1",
10        "tag2",
11      ],
12       "createdBy": {
13         "_id": "9L3wuBXKpbAwdfgx8",
14         "username": "rafael.ferreira"
15       },
16       "_createdAt": "2021-05-26T18:58:48.855Z",
17       "_updatedAt": "2021-05-26T18:58:48.855Z"
18     }
```

```
19  ],
20  "count": 1,
21  "offset": 0,
22  "total": 1,
23  "success": true
24 }
```

Change Log

Version	Description
1.0.0	Added

Get the canned response info

Retrieve the visitor info

URL	Requires Auth	HTTP Method
/api/v1/canned-responses.getOne	yes	GET

Query Parameter

Argument	Example	Required	Description
_id	EwmbZ9nLSx7kFamYB	Required	The canned response's id.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/canned-responses.getOne?_id=EwmbZ9nLSx7k
```

Example Result

```
1  {
2    "cannedResponse": {
3      "_id": "EwmbZ9nLSx7kFamYB",
4      "shortcut": "user",
5      "text": "Meu 32423423423423423",
6      "scope": "user",
7      "userId": "9L3wuBXKpbAwdfgx8",
8      "createdBy": {
9        "_id": "9L3wuBXKpbAwdfgx8",
10       "username": "rafael.ferreira"
11     },
12     "_createdAt": "2021-05-26T18:59:17.929Z",
13     "_updatedAt": "2021-05-26T18:59:17.929Z"
14   },
15   "success": true
16 }
```

Change Log

Version	Description
1.0.0	Added

Register a new canned response

URL	Requires Auth	HTTP Method
/api/v1/canned-responses	yes	POST

Example payload

```
1  {
2    "shortcut": "my-new-canned",
3    "text": "This is an example",
4    "scope": "global",
5    "tags": ["tag1", "tag2"]
6 }
```

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -X POST \
4   -H "Content-type:application/json" \
5   http://localhost:3000/api/v1/canned-responses \
6   -d '{    "shortcut": "my-new-canned", "text": "This is an example", "s'
```

Example Result

```
1  {
2    "success": true
3 }
```

Change Log

Version	Description
1.0.0	Added

Update a canned response

URL	Requires Auth	HTTP Method
/api/v1/canned-responses	yes	POST

Example payload

```
1  {
2    "_id": "N5D2posu4fCqamjae",
3    "shortcut": "my-new-canned",
4    "text": "This is an example",
5    "scope": "global",
6    "tags": ["tag1", "tag2"]
7 }
```

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -X POST \
4   -H "Content-type:application/json" \
5   http://localhost:3000/api/v1/canned-responses \
6   -d '{      "_id": "N5D2posu4fCqamjae", "shortcut": "my-new-canned", "tex
```



Example Result

```
1  {
2    "success": true
3 }
```

Change Log

Version	Description
1.0.0	Added

Delete a canned response

Retrieve the visitor info

URL	Requires Auth	HTTP Method
/api/v1/canned-responses	yes	DELETE

Query Parameter

Argument	Example	Required	Description
_id	EwmbZ9nLSx7kFamYB	Required	The canned response's id.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -X DELETE \
4     http://localhost:3000/api/v1/canned-responses?_id=N5D2posu4fCqamjae
```

Example Result

```
1 {
2     "success": true
3 }
```

Change Log

Version	Description
1.0.0	Added

Config

Get Livechat widget configuration info and additional visitor data

URL	Requires Auth	HTTP Method
/api/v1/livechat/config	no	GET

Query Parameters

Argument	Example	Required	Description
token	iNKE8a6k6cjbjqWhWd	Optional	The visitor's token .
department	department	Optional	The visitor's department .

Example Calls

```
1 curl http://localhost:3000/api/v1/livechat/config?token=iNKE8a6k6cjbjqWhWd
2 curl http://localhost:3000/api/v1/livechat/config?department=department
```

Example Result

```
1 {
2   "config": {
3     "enabled": true,
4     "settings": {
5       "registrationForm": true,
```

```
6      "allowSwitchingDepartments": false,
7      "nameFieldRegistrationForm": true,
8      "emailFieldRegistrationForm": true,
9      "displayOfflineForm": true,
10     "videoCall": false,
11     "fileUpload": true,
12     "language": "",
13     "transcript": false,
14     "historyMonitorType": "url"
15   },
16   "theme": {
17     "title": "Rocket.Chat",
18     "color": "#C1272D",
19     "offlineTitle": "Leave a message",
20     "offlineColor": "#666666",
21     "actionLinks": [
22       {
23         "icon": "icon-videoocam",
24         "i18nLabel": "Accept",
25         "method_id": "createLivechatCall",
26         "params": ""
27       },
28       {
29         "icon": "icon-cancel",
30         "i18nLabel": "Decline",
31         "method_id": "denyLivechatCall",
32         "params": ""
33       }
34     ],
35   },
36   "messages": {
37     "offlineMessage": "",
38     "offlineSuccessMessage": "",
39     "offlineUnavailableMessage": "",
40     "conversationFinishedMessage": "",
41     "transcriptMessage": ""
42   },
43   "survey": {
44     "items": [
45       "satisfaction",
46       "agentKnowledge",
47       "agentResposiveness",
48       "agentFriendliness"
49     ],
50     "values": [
51       "1",
52       "2",
53       "3",
54       "4",
55       "5"
56     ],
57   },
58 }
```

```
57     "online": true,
58     "guest": {
59       "_id": "2KNu66RPCwxA4ncy7",
60       "username": "guest-3",
61       "token": "iNKE8a6k6cjbjqWhWd",
62       "name": "Livechat Visitor"
63     },
64     "room": {
65       "_id": "zRAeTszXor8CCPceB",
66       "servedBy": {
67         "_id": "7Gm3PoFCJWTCJ68XR",
68         "username": "livechat.agent"
69       },
70       "open": true
71     },
72     "agent": {
73       "_id": "7Gm3PoFCJWTCJ68XR",
74       "emails": [
75         {
76           "address": "agent@rocket.chat",
77           "verified": true
78         },
79         "name": "Livechat Agent",
80         "username": "livechat.agent"
81       }
82     },
83     "success": true
84   }
```

Change Log

Version	Description
0.70.0	Added

Custom Field

Send a Livechat custom field

URL	Requires Auth	HTTP Method
/api/v1/livechat/custom.field	no	POST

Example payload

```
1  {
2    "token": "iNKE8a6k6cjBqWhWd",
3    "key": "address",
4    "value": "Rocket.Chat Avenue",
5    "overwrite": true
6 }
```

Example Call

```
1 curl -X POST \
2   -H "Content-type:application/json" \
3   http://localhost:3000/api/v1/livechat/custom.field \
4   -d '{
      "token": "iNKE8a6k6cjBqWhWd",
      "key": "address",
      "value": "Rocket.
```

Example Result

```
1  {
```

```
2   "field": {
3     "key": "address",
4     "value": "Rocket.Chat Avenue",
5     "overwrite": true
6   },
7   "success": true
8 }
```

Change Log

Version	Description
0.70.0	Added

Send an array of Livechat custom fields

URL	Requires Auth	HTTP Method
/api/v1/livechat/custom.fields	no	POST

Example payload

```
1  {
2    "token": "iNKE8a6k6cjBqWhWd",
3    "customFields": [
4      {
5        "key": "address",
6        "value": "Rocket.Chat Avenue",
7        "overwrite": true
8      }
9    }
10 }
```

Example Call

```
1 curl -X POST \
2     -H "Content-type:application/json" \
3     http://localhost:3000/api/v1/livechat/custom.fields \
4     -d '{"token": "iNKE8a6k6cjBqWhWd", "customFields": [{"key": "address"}]}
```

Example Result

```
1 {
2     "fields": [
3         {
4             "Key": "address",
5             "value": "Rocket.Chat Avenue",
6             "overwrite": true
7         }
8     ],
9     "success": true
10 }
```

Change Log

Version	Description
0.70.0	Added

List livechat custom fields

Get a list of Livechat custom fields. It supports the [Offset, Count, and Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/livechat/custom-fields	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/livechat/custom-fields
```

Example Result

```
1  {
2      "customFields": [
3          {
4              "_id": "address",
5              "label": "address",
6              "scope": "visitor",
7              "visibility": "visible",
8              "_updatedAt": "2019-10-03T14:12:47.595Z"
9          }
10     ],
11     "count": 1,
12     "offset": 0,
13     "total": 1,
14     "success": true
15 }
```

Change Log

Version	Description
2.2.0	Added

Get info about a custom field

Get info about a custom field.

URL	Requires Auth	HTTP Method
/api/v1/livechat/custom-fields/:_id	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/livechat/custom-fields/address
```

Example Result

```
1 {
2   "customField": {
3     "_id": "address",
4     "label": "address",
5     "scope": "visitor",
6     "visibility": "visible",
7     "_updatedAt": "2019-10-03T14:12:47.595Z"
```

```
8      },
9      "success": true
10 }
```

Change Log

Version	Description
2.4.0	Added

Department

Get a list of departments. It supports the [Offset, Count, and Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/livechat/department	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/livechat/department
```

Example Result

```
1 {
2     "departments": [
3         {
4             "_id": "4swtja84kmn5WCdwL",
5             "enabled": true,
6             "name": "My Department",
7             "description": "I have no description for this department",
8             "numAgents": 1,
9             "showOnRegistration": true,
10            "_updatedAt": "2016-12-06T17:19:18.138Z"
11        }
12    ],
13    "success": true
14 }
```

Change Log

Version	Description
2.2.0	Added support to pagination
0.42.0	Added

Register a new department

URL	Requires Auth	HTTP Method
/api/v1/livechat/department	yes	POST

Example payload

```
 1  {
 2    "department": {
 3      "enabled": false,
 4      "showOnRegistration": true,
 5      "email": "email@email.com",
 6      "showOnRegistration": true,
 7      "name": "new from api",
 8      "description": "created from api"
 9    },
10    "agents": [
11      {"agentId": "SQafHvoFPub57NmBD",
12       "username": "john.doe",
13       "count": 0,
14       "order": 0
15     }
16  }
```

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -X POST \
4     -H "Content-type:application/json" \
5     http://localhost:3000/api/v1/livechat/department \
6     -d '{"department": {"enabled": false, "showOnRegistration": true, "name": "new from api", "numAgents": 1, "agents": [{"agentId": "SQafHvoFPuB57NmBD", "username": "john.doe", "departmentId": "iTfLCX3qqwKgf5uqg", "count": 0, "order": 0, "updatedAt": "2016-12-13T17:22:19.169Z"}, {"agentId": "DDjZbhTF74n3NBuWK", "username": "john.doe", "departmentId": "iTfLCX3qqwKgf5uqg", "count": 0, "order": 1, "updatedAt": "2016-12-13T17:22:19.169Z"}], "success": true}}
```

Example Result

```
1 {
2     "department": {
3         "enabled": false,
4         "name": "new from api",
5         "numAgents": 1,
6         "showOnRegistration": true,
7         "_updatedAt": "2016-12-13T17:22:19.109Z",
8         "_id": "iTfLCX3qqwKgf5uqg"
9     },
10    "agents": [
11        {
12            "_id": "DDjZbhTF74n3NBuWK",
13            "agentId": "SQafHvoFPuB57NmBD",
14            "departmentId": "iTfLCX3qqwKgf5uqg",
15            "username": "john.doe",
16            "count": 0,
17            "order": 0,
18            "_updatedAt": "2016-12-13T17:22:19.169Z"
19        }
20    ],
21    "success": true
22 }
```

Change Log

Version	Description
1.0.0	New fields for department updated
0.42.0	Added

Get info about a department

URL	Requires Auth	HTTP Method
/api/v1/livechat/department/:_id	yes	GET

Payload

Argument	Example	Required	Description
_id	SQafHvoFPuB57NmBD	Required	The department _id .

Query Parameter

Argument	Example	Required	Description
includeAgents	true	Optional(default true)	If agents should be included.

Note: The `agents` field will only be returned if the user has the `view-livechat-departments` permission.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/livechat/department/iTfLCX3qqwKgf5uqg
```

Example Result

```
1  {
2     "department": {
3         "_id": "iTfLCX3qqwKgf5uqg",
4         "enabled": false,
5         "name": "new from api",
6         "description": null,
7         "numAgents": 1,
8         "showOnRegistration": true,
9         "_updatedAt": "2016-12-13T17:22:19.109Z"
10    },
11    "agents": [
12        {
13            "_id": "DDjZbhTF74n3NBuWK",
14            "agentId": "SQafHvoFPuB57NmBD",
15            "departmentId": "iTfLCX3qqwKgf5uqg",
16            "username": "john.doe",
17            "count": 0,
18            "order": 0,
19            "_updatedAt": "2016-12-13T17:22:19.169Z"
20        }
21    ],
22    "success": true
23 }
```

Change Log

Version	Description
---------	-------------

2.2.0	Added <code>includeAgents</code> query parameter
0.42.0	Added

Update a department

URL	Requires Auth	HTTP Method
/api/v1/livechat/department/:_id	yes	PUT

Payload

Argument	Example	Required	Description
_id	iTfLCX3qqwKgf5uqg	Required	Department <code>_id</code> .

Example payload

```
1  {
2    "department": {
3      "enabled": true,
4      "showOnRegistration": true,
5      "name": "new from api - live"
6    },
7    "agents": [
8      {"agentId": "SQafHvoFPuB57NmBD",
9       "username": "john.doe",
10      "count": 0,
11      "order": 0
12    ]
13  }
```

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -X PUT \
4     -H "Content-type:application/json" \
5     http://localhost:3000/api/v1/livechat/department/iTfLCX3qqwKgf5uqg \
6     -d '{"department": {"enabled": true, "name": "new from api - live", "s'`
```

Example Result

```
1  {
2     "department": {
3         "_id": "iTfLCX3qqwKgf5uqg",
4         "enabled": true,
5         "name": "new from api - live",
6         "description": null,
7         "numAgents": 1,
8         "showOnRegistration": true,
9         "_updatedAt": "2016-12-13T17:30:02.643Z"
10    },
11    "agents": [
12      {
13          "_id": "DDjZbhTF74n3NBuWK",
14          "agentId": "SQafHvoFPuB57NmBD",
15          "departmentId": "iTfLCX3qqwKgf5uqg",
16          "username": "john.doe",
17          "count": 0,
18          "order": 0,
19          "_updatedAt": "2016-12-13T17:30:02.656Z"
20      }
21    ],
22    "success": true
23 }
```

Change Log

Version	Description
0.42.0	Added

Removes a department

URL	Requires Auth	HTTP Method
/api/v1/livechat/department/:_id	yes	DELETE

Payload

Argument	Example	Required	Description
_id	iTfLCX3qqwKgf5uqg	Required	The department _id .

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -X DELETE \
4   http://localhost:3000/api/v1/livechat/department/iTfLCX3qqwKgf5uqg
```

Example Result

```
1  {
2    "success": true
3 }
```

Change Log

Version	Description
0.42.0	Added

Inquiries List

Lists all of the open livechat inquiries. It supports the [Offset, Count, and Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/livechat/inquiries.list	yes	GET

Query Parameters

Argument	Example	Required	Description
department	ByehQjC44FwMeiLbx	Optional	The department's id or name

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/livechat/inquiries.list
```

Example Result

```
1 {
2   "inquiries": [
3     {
4       "_id": "GpxfRo8TaPHfsnnC5",
5       "rid": "EbQjtCosHJWLQmQYT",
```

```

6           "name": "Marcos Defendi",
7           "ts": "2019-06-11T19:01:57.424Z",
8           "department": "jHFgdJyJWstB9M2ik",
9           "status": "open"
10      },
11      {
12          "_id": "9dCi64GkwWE389xCm",
13          "rid": "ZpjCcbRwPPdnhKWW5",
14          "name": "Marcos Defendi",
15          "ts": "2019-06-11T19:00:43.124Z",
16          "department": "gDDAjeJb7BhHzzEQ8",
17          "status": "open"
18      }
19  ],
20  "offset": 0,
21  "count": 1,
22  "total": 1,
23  "success": true
24 }

```

Change Log

Version	Description
1.2.0	Added

Livechat take inquiry

Takes an open inquiry.

URL	Requires Auth	HTTP Method
/api/v1/livechat/inquiries.take	yes	POST

Payload

Argument	Example	Required	Description
inquiryId	ByehQjC44FwMeiLbX	Required	The inquiry's id
userId	ByehQjCfsd876sfd	Optional	The user's (agent) id to take the inquiry.

Note: if the user id is provided, the user must have the `view-l-room` permission.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/livechat/inquiries.take \
5   -d '{ "inquiryId": "ByehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
2   "inquiry": {
3     "_id": "wbKmn6pAZ8jyJuANG",
4     "rid": "S4wwL9WNY98uoHgJg",
5     "message": "test",
6     "name": "teste",
7     "ts": "2019-06-10T23:09:06.482Z",
8     "agents": [
9       "hjwGZafNqExtFNmN7",
10      "26KdXgrQXhddy2MfQ"
11    ],
12    "status": "open",
13    "v": {
14      "_id": "2iZSexGXjW7EJnRwM",
```

```

15         "username": "guest-3",
16         "token": "RtQzkfQYKG4WpNMEW",
17         "status": "online"
18     },
19     "t": "l",
20     "_updatedAt": "2019-06-10T23:09:07.480Z"
21 },
22   "success": true
23 }
```

Change Log

Version	Description
1.2.0	Added

Livechat Get one inquiry by room id

Get one inquiry by room id.

URL	Requires Auth	HTTP Method
/api/v1/livechat/inquiries.getOne	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The room's id

Note: if the user id is provided, the user must have the `view-l-room` permission.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -H "Content-type: application/json" \
4     http://localhost:3000/api/v1/livechat/inquiries.getOne?roomId=ByehQjC
```

Example Result

```
1  {
2     "inquiry": {
3         "_id": "EMXtMA7mPPNkQtWWq",
4         "rid": "MA6HRhkEdRGATfEbh",
5         "name": "inquiry test",
6         "ts": "2019-12-13T18:54:25.352Z",
7         "message": "",
8         "status": "queued",
9         "v": {
10             "_id": "rLitbjv7e9WyLBCR4",
11             "username": "guest-14",
12             "token": "btzu0qdm0sjsn7aqm78kzr",
13             "status": "offline"
14         },
15         "t": "l",
16         "_updatedAt": "2019-12-13T19:33:13.255Z"
17     },
18     "success": true
19 }
```

Change Log

Version	Description
2.4.0	Added

Integrations Settings

Retrieves a list of integration settings.

URL	Requires Auth	HTTP Method
/api/v1/livechat/integrations.settings	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/livechat/integrations.settings
```

Example Result

```
1 {
2   "settings": [
3     {
4       "_id": "Livechat_secret_token",
5       "_updatedAt": "2019-10-17T13:56:07.285Z",
6       "autocomplete": true,
7       "blocked": false,
8       "createdAt": "2019-10-07T18:52:20.484Z",
9       "group": "Livechat",
10      "hidden": false,
11      "i18nDescription": "Livechat_secret_token_Description",
12      "i18nLabel": "Secret_token",
13      "packageValue": "",
14      "secret": true,
15      "section": "CRM_Integration",
16      "sorter": 25,
17      "ts": "2019-10-07T21:00:01.287Z",
18      "type": "string",
```

```
19         "value": "",  
20         "valueSource": "meteorSettingsValue",  
21         "meteorSettingsValue": ""  
22     }  
23 ],  
24 "success": true  
25 }
```

Change Log

Version	Description
2.2.0	Added

Message

Send a new Livechat message

URL	Requires Auth	HTTP Method
/api/v1/livechat/message	no	POST

Payload

Argument	Example	Required	Description
token	iNKE8a6k6cjbjqWhWd	Required	Visitor token .
rid	zRAeTszXor8CCPceB	Required	Room _id .
msg	sending livechat message..	Required	Message text .
_id		Optional	Message _id .
agent		Optional	Room agent .

Example payload

```
1  {
2    "token": "iNKE8a6k6cjbjqWhWd",
3    "rid": "zRAeTszXor8CCPceB",
4    "msg": "sending livechat message.."
5 }
```

Example Call

```
1 curl -X POST \
2     -H "Content-type:application/json" \
3     http://localhost:3000/api/v1/livechat/message \
4     -d '{"token": "iNKE8a6k6cjBqWhWd", "rid": "zRAeTszXor8CCPceB", "msg":
```

Example Result

```
1  {
2     "message": {
3         "_id": "ZKWP8LfGnRHQ3ozWa",
4         "msg": "sending livechat message..",
5         "u": {
6             "_id": "iNKE8a6k6cjBqWhWd",
7             "username": "guest-4",
8             "name": "Livechat Visitor"
9         },
10        "ls": "2018-09-14T13:31:33.201Z"
11    },
12    "success": true
13 }
```

Change Log

Version	Description
0.70.0	Added

Updates a Livechat message

URL	Requires Auth	HTTP Method
/api/v1/livechat/message/:_id	no	PUT

Payload

Argument	Example	Required	Description
_id	ZKWP8LfGnRHQ3ozWa	Required	Message _id .
token	iNKE8a6k6cjbjqWhWd	Required	Visitor token .
rid	zRAeTszXor8CCPceB	Required	Room _id .
msg	editing a livechat message..	Required	Message text .

Example payload

```
1  {
2    "token": "iNKE8a6k6cjbjqWhWd",
3    "rid": "zRAeTszXor8CCPceB",
4    "msg": "editing a livechat message.."
5 }
```

Example Call

```
1 curl -X PUT \
2   -H "Content-type:application/json" \
3   http://localhost:3000/api/v1/livechat/message/ZKWP8LfGnRHQ3ozWa \
4   -d '{"token": "iNKE8a6k6cjbjqWhWd", "rid": "zRAeTszXor8CCPceB", "msg": "
```

Example Result

```
1  {
2    "message": {
3      "_id": "ZKWP8LfGnRHQ3ozWa",
4      "msg": "editing livechat message..",
5      "u": {
6        "_id": "YgEoq2djbGdjZnsL",
7        "username": "guest-4",
8        "name": "Livechat Visitor"
9      },
10     "ls": "2018-09-14T13:31:33.201Z"
11   },
12   "success": true
13 }
```

Change Log

Version	Description
0.70.0	Added

Removes a Livechat message

URL	Requires Auth	HTTP Method
/api/v1/livechat/message/:_id	no	DELETE

Payload

Argument	Example	Required	Description
_id	ZKWP8LfGnRHQ3ozWa	Required	Message _id .

Example Call

```
1 curl -X DELETE \
2      http://localhost:3000/api/v1/livechat/message/ZKWP8LfGnRHQ3ozWa
```

Example Result

```
1 {
2   "message": {
3     "_id": "ZKWP8LfGnRHQ3ozWa",
4     "ls": "2018-09-14T13:31:33.279Z"
5   },
6   "success": true
7 }
```

Change Log

Version	Description
0.70.0	Added

Load Livechat messages history

URL	Requires Auth	HTTP Method
/api/v1/livechat/messages.history/:rid	no	GET

Payload

Argument	Example	Required	Description
rid	KuACMJ5MpN6SfAFWg	Required	The room _id .

Query Parameters

Argument	Example	Required	Description
token	iNKE8a6k6cjbjqWhWd	Required	The visitor token .
ls		Optional	The timestamp to start loading.
end		Optional	The timestamp limit to load.
limit		Optional	The number of messages to load.

Example Call

```
curl http://localhost:3000/api/v1/livechat/messages.history/KuACMJ5MpN6SfAFWg
```

Example Result

```
1  {
2    "messages": [
3      {
4        "_id": "ZKWP8LfGnRHQ3ozWa",
5        "rid": "KuACMJ5MpN6SfAFWg",
6        "msg": "editing livechat message..",
7        "token": "iNKE8a6k6cjbqWhWd",
8        "alias": "Livechat Visitor",
9        "ls": "2018-09-14T13:31:33.201Z",
10       "u": {
11         "_id": "YgEoq2djbGdjZnsL",
12         "username": "guest-4",
13         "name": "Livechat Visitor"
14       },
15       "mentions": [],
16       "channels": [],
17       "_updatedAt": "2018-09-14T13:31:33.222Z",
18       "editedAt": "2018-09-14T13:31:33.219Z",
19       "editedBy": {
20         "_id": "YgEoq2djbGdjZnsL",
21         "username": "guest-4"
22       },
23       "urls": []
24     }],
25     "unreadNotLoaded": 0,
26     "success": true
27   }
```

Change Log

Version	Description
0.70.0	Added

Send a new Livechat offline message.

URL	Requires Auth	HTTP Method
/api/v1/livechat/offline.message	no	POST

Payload

Argument	Example	Required	Description
name	Livechat Visitor	Required	Message name .
email	visitor@rocket.chat	Required	Message email .
message	I need help	Required	Message text .

Example payload

```
1  {
2    "name": "Livechat Visitor",
3    "email": "visitor@rocket.chat",
4    "message": "I need help"
5 }
```

Example Call

```
1 curl -X POST \
2   -H "Content-type:application/json" \
3   http://localhost:3000/api/v1/livechat/offline.message \
4   -d '{"name": "Livechat Visitor", "email": "visitor@rocket.chat", "mes'`
```

Example Result

```
1  {
2    "message": "Livechat offline message sent",
3    "success": true
4 }
```

Change Log

Version	Description
0.70.0	Added

Office Hours

Get Livechat office hours

URL	Requires Auth	HTTP Method
/api/v1/livechat/office-hours	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/livechat/office-hours
```

Example Result

```
1 {
2   "officeHours": [
3     {
4       "_id": "A7gaBDnbGen3qFqtx",
5       "day": "Monday",
6       "start": "08:00",
7       "finish": "20:00",
8       "code": 1,
9       "open": true,
10      "_updatedAt": "2019-10-04T13:50:04.010Z"
11    },
12    {
13      "_id": "DjjRM49xwZ3LMraSC",
14      "day": "Tuesday",
15      "start": "08:00",
16      "finish": "20:00",
17      "code": 2,
18      "open": true,
```

```
19         "_updatedAt": "2019-10-04T13:50:04.023Z"
20     },
21     {
22         "_id": "zx6vTXginatNnbWc2",
23         "day": "Wednesday",
24         "start": "08:00",
25         "finish": "20:00",
26         "code": 3,
27         "open": true,
28         "_updatedAt": "2019-10-04T13:50:04.038Z"
29     },
30     {
31         "_id": "YWx7a3ebA5KeB9uNW",
32         "day": "Thursday",
33         "start": "08:00",
34         "finish": "20:00",
35         "code": 4,
36         "open": true,
37         "_updatedAt": "2019-10-04T13:50:04.048Z"
38     },
39     {
40         "_id": "jw25hqfWnH3XvvxPm",
41         "day": "Friday",
42         "start": "08:00",
43         "finish": "20:00",
44         "code": 5,
45         "open": true,
46         "_updatedAt": "2019-10-04T13:50:04.060Z"
47     },
48     {
49         "_id": "hLvF69wFf49AYYF6D",
50         "day": "Saturday",
51         "start": "08:00",
52         "finish": "20:00",
53         "code": 6,
54         "open": false,
55         "_updatedAt": "2019-10-04T13:50:04.071Z"
56     },
57     {
58         "_id": "QHRZeuwXNPYXdoaJN",
59         "day": "Sunday",
60         "start": "08:00",
61         "finish": "20:00",
62         "code": 0,
63         "open": false,
64         "_updatedAt": "2019-10-04T13:50:04.093Z"
65     }
66 ],
67 "success": true
68 }
```

Change Log

Version	Description
2.2.0	Added

Queue

List queued chats

Get the queued chats. It supports the [Offset, Count, and Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/livechat/queue	yes	GET

Query params

Argument	Example	Required	Description
includeOfflineAgents	true	Optional	Whether include offline agents or not.
agentId	aobEdbYhXfu5dsd9	Optional	The agent's id.
departmentId	3n2kj3n2j3fu5hkeqG	Optional	The department's id.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/livechat/queue?includeOfflineAgents=true
```

Example Result

```
1  {
2      "queue": [
3      {
4          "chats": 1,
5          "user": {
6              "_id": "CaevAPDbXN252kyXa",
7              "username": "marcos.defendi",
8              "status": "online"
9          },
10         "department": {
11             "_id": "457diLwcQMmdpaTjo",
12             "name": "Department 1"
13         }
14     },
15     {
16         "chats": 2,
17         "user": {
18             "_id": "CaevAPDbXN252kyXa",
19             "username": "marcos.defendi",
20             "status": "online"
21         },
22         "department": {
23             "_id": "5mt9oEtQxuMMH23Co",
24             "name": "Department 2"
25         }
26     }
27 ],
28 "count": 2,
29 "offset": 0,
30 "total": 2,
31 "success": true
32 }
```

Change Log

Version	Description
2.4.0	Added

Room

Get the Livechat room data or open a new room.

URL	Requires Auth	HTTP Method
/api/v1/livechat/room	no	GET

Query Parameters

Argument	Example	Required	Description
token	iNKE8a6k6cjbjqWhWd	Required	The visitor token .
rid		Optional	The room _id .

Example Call

```
curl http://localhost:3000/api/v1/livechat/room?token=iNKE8a6k6cjbjqWhWd
```

Example Result

```
1  {
2    "room": {
3      "_id": "onrMgdKbpX9Qqtvoi",
4      "msgs": 0,
5      "usersCount": 1,
6      "lm": "2018-09-14T17:36:48.276Z",
```

```
7      "fname": "Livechat Visitor",
8      "t": "l",
9      "ts": "2018-09-14T17:36:48.276Z",
10     "v": {
11       "_id": "YgEoq2djbGdjZnsL",
12       "username": "guest-4",
13       "token": "iNKE8a6k6cjbqWhWd",
14       "status": "online"
15     },
16     "servedBy": {
17       "_id": "7Gm3PoFCJWTCJ68XR",
18       "username": "livechat.agent"
19     },
20     "cl": false,
21     "open": true,
22     "waitingResponse": true,
23     "jitsiTimeout": "2018-09-14T18:36:48.273Z",
24     "_updatedAt": "2018-09-14T17:36:48.276Z"
25   },
26   "newRoom": true,
27   "success": true
28 }
```

Change Log

Version	Description
0.70.0	Added

Close Livechat room

URL	Requires Auth	HTTP Method
/api/v1/livechat/room.close	no	POST

Payload

Argument	Example	Required	Description
rid	XFzMqgn33DcsQkpJp	Required	The room <code>_id</code> .
token	iNKE8a6k6cjbjqWhWd	Required	The visitor <code>token</code> .

Example payload

```
1  {
2    "rid": "XFzMqgn33DcsQkpJp",
3    "token": "iNKE8a6k6cjbjqWhWd"
4 }
```

Example Call

```
1 curl -X POST \
2   -H "Content-type:application/json" \
3   http://localhost:3000/api/v1/livechat/room.close \
4   -d '{"rid": "XFzMqgn33DcsQkpJp", "token": "iNKE8a6k6cjbjqWhWd"}'
```

Example Result

```
1  {
2    "rid": "XFzMqgn33DcsQkpJp",
3    "comment": "Closed by visitor",
4    "success": true
```

```
5 }
```

Change Log

Version	Description
0.70.0	Added

Transfer Livechat room

URL	Requires Auth	HTTP Method
/api/v1/livechat/room.transfer	no	POST

Payload

Argument	Example	Required	Description
rid	XFzMqgn33DcsQkpJp	Required	The room <code>_id</code> .
token	iNKE8a6k6cjbjqWhWd	Required	The visitor <code>token</code> .
department	wXpPLofkffqWAwDNF	Required	The new <code>department</code> .

Example payload

```
1  {
2    "rid": "XFzMqgn33DcsQkpJp",
3    "token": "iNKE8a6k6cjBqWhWd",
4    "department": "wXpPLofkffqWAwDNF"
5 }
```

Example Call

```
1 curl -X POST \
2   -H "Content-type:application/json" \
3   http://localhost:3000/api/v1/livechat/room.transfer \
4   -d '{"rid": "XFzMqgn33DcsQkpJp", "token": "iNKE8a6k6cjBqWhWd", "depar'}
```

Example Result

```
1  {
2    "room": {
3      "_id": "XFzMqgn33DcsQkpJp",
4      "servedBy": {
5        "_id": "wiyTfFKXr5GhgRu9A",
6        "username": "livechat.agent"
7      },
8      "open": true,
9      "departmentId": "wXpPLofkffqWAwDNF"
10    },
11    "success": true
12  }
```

Change Log

Version	Description
0.70.0	Added

Forward Livechat room

URL	Requires Auth	HTTP Method
/api/v1/livechat/room.forward	yes	POST

Payload

Argument	Example	Required	Description
roomId	XFzMqgn33DcsQkpJp	Required	The room <code>_id</code> .
userId	iNKE8a6k6cjbjqWhWd	Optional	The new <code>agent ID</code> .
departmentId	wXpPLofkffqWAwDNF	Optional	The new <code>department ID</code> .

Example payload

```
1  {
2    "roomId": "XFzMqgn33DcsQkpJp",
3    "userId": "iNKE8a6k6cjbjqWhWd"
4 }
```

```
1 {
```

```
2   "roomId": "XFzMqgn33DcsQkpJp",
3   "departmentId": "wXpPLofkffqWAwDNF"
4 }
```

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/livechat/room.forward \
5   -d '{"roomId": "XFzMqgn33DcsQkpJp", "userId": "iNKE8a6k6cjBqWhWd"}'
```

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/livechat/room.forward \
5   -d '{"roomId": "XFzMqgn33DcsQkpJp", "departmentId": "wXpPLofkffqWAwDNF"}'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
---------	-------------

0.74.1	Added
<hr/>	
<h2>Send the Livechat survey</h2>	
URL	Requires Auth
/api/v1/livechat/room.survey	no
HTTP Method	POST

Payload

Argument	Example	Required	Description
rid	XFzMqgn33DcsQkpJp	Required	The room <code>_id</code> .
token	iNKE8a6k6cjBqWhWd	Required	The visitor <code>token</code> .
data	[{"name": "satisfaction", "value": "3"}]	Required	The survey <code>data</code> .

Example payload

```
1  {
2      "rid": "XFzMqgn33DcsQkpJp",
3      "token": "iNKE8a6k6cj bqWhWd",
4      "data": [
5          {
6              "name": "satisfaction",
7              "value": "3"
8          }
9      ]
}
```

```
10 }
```

Example Call

```
1 curl -X POST \
2     -H "Content-type:application/json" \
3     http://localhost:3000/api/v1/livechat/room.survey \
4     -d '{"rid":"XFzMqgn33DcsQkpJp", "token": "iNKE8a6k6cjBqWhWd", "data":}
```

Example Result

```
1 {
2     "rid": "XFzMqgn33DcsQkpJp",
3     "data": {
4         "satisfaction": "3",
5         "agentResponiveness": "5"
6     },
7     "success": true
8 }
```

Change Log

Version	Description
0.70.0	Added

Change Room Visitor

URL	Requires Auth	HTTP Method
/api/v1/livechat/room.visitor	yes	PUT

Payload

Argument	Example	Required	Description
rid	XFzMqgn33DcsQkpJp	Required	The room <code>_id</code> .
visitorId	iNKE8a6k6cjBqWhWd	Required	The visitor <code>_id_</code> .

Example payload

```
1  {
2    "rid": "2xC3rGZRijztFYP5t",
3    "visitorId": "qpdxquZygF4j9aFTR",
4 }
```

Example Call

```
1 curl "http://localhost:3000/api/v1/livechat/room.visitor" \
2   -X PUT \
3   -d "{\"rid\": \"2xC3rGZRijztFYP5t\", \"visitorId\": \"qpdxquZygF4j9aFTR\"}"
4   -H "X-User-Id: YE3xHYifqk4ic3e3B" \
5   -H "X-Auth-Token: DVeIk8owiiudy0_lP981GkTWXLТАRcCNkef5XuIDVY5" \
6   -H "Content-Type: application/json"
```

Example Result

```
1  {
2    "rid": "XFzMqgn33DcsQkpJp",
3    "room": {
4      "rid": "XFzMqgn33DcsQkpJp",
5      "v": {
6        "_id": "qpdxquZygF4j9aFTR",
7        "username": "guest-2",
8        "token": "gn33Dcsgn33DcsQk"
9      }
10    },
11    "success": true
12 }
```

Change Log

Version	Description
3.6.0	Added

Rooms

Retrieves a list of livechat rooms. It supports the [Offset, Count, and Sort Query Parameters](#) along with [Fields Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/livechat/rooms	yes	GET

Query Parameters

Argument	Example	Required	Description
agents	['b32h3b2hhwb25d']	Optional	A list of agents, can be a list of ids or usernames
departmentId	AkzpHAvZpdnuchw2a	Optional	The department's id
open	true	Optional	If it should filter by open/closed rooms
createdAt	{"start": "2018-01-26T00:11:22.345Z", "end": "2018-01-26T00:11:22.345Z"}	Optional	An object representing when the room was created(Can also be filtered only with start or end)
closedAt	{"start": "2018-01-26T00:11:22.345Z", "end": "2018-01-26T00:11:22.345Z"}	Optional	An object representing when the room was closed(Can also be filtered only with start or end)
tags	['rocket', 'chat']	Optional	A list of tags

customFields	{"docId": "aobEdbYhXfu5hkeqG"}	Optional	An object with custom fields to be filtered (previously created and populated at custom fields endpoints)
roomName	room name	Optional	The room's name

Example Call

```

1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/livechat/rooms?agents[]="teste&departamen

```

Example Result

```

1 {
2   "rooms": [
3     {
4       "_id": "9dQMveZhqxidgCM3B",
5       "msgs": 1,
6       "usersCount": 1,
7       "lm": "2019-08-09T20:18:02.350Z",
8       "fname": "Marcos Defendi",
9       "t": "l",
10      "ts": "2019-08-09T20:18:02.108Z",
11      "v": {
12        "_id": "xh45w4y7oPW5SiMJU",
13        "username": "guest-4",
14        "token": "xn5hdkypteil1m07f3me",
15        "status": "online"
16      },
17      "servedBy": {
18        "_id": "3kyWm8PsmwhC7xgwg",
19        "username": "marcos",
20        "ts": "2019-08-09T20:18:02.109Z"

```

```

21     },
22     "cl": false,
23     "open": true,
24     "waitingResponse": true,
25     "departmentId": "xnmXQmytuvxp2TuSC",
26     "_updatedAt": "2019-08-09T20:19:34.856Z",
27     "lastMessage": {
28         "_id": "TBEz7zSCBaKsfyrc",
29         "rid": "9dQMveZhqxiDgCM3B",
30         "msg": "Hello",
31         "token": "xn5hdkyptei11m07f3me",
32         "alias": "Marcos Defendi",
33         "ts": "2019-08-09T20:18:02.350Z",
34         "u": {
35             "_id": "xh45w4y7oPW5SiMJU",
36             "username": "guest-4",
37             "name": "Marcos Defendi"
38         },
39         "_updatedAt": "2019-08-09T20:18:02.383Z",
40         "mentions": [],
41         "channels": [],
42         "newRoom": false,
43         "showConnecting": false
44     },
45     "metrics": {
46         "v": {
47             "lq": "2019-08-09T20:18:02.350Z"
48         }
49     },
50     "livechatData": {
51         "docId": "0310584343112"
52     },
53     "tags": [
54         "rocket",
55         "chat"
56     ],
57     "name": "Marcos Defendi"
58 }
59 ],
60 "count": 1,
61 "offset": 1,
62 "total": 7,
63 "success": true
64 }

```

Change Log

Version	Description
2.4.0	Added support to the parameter <code>roomName</code>
2.0.0	Added

SMS Incoming

URL	Requires Auth	HTTP Method
/api/v1/livechat/sms-incoming/:service	no	POST

Payload

Argument	Example	Required	Description
service	twilio	Required	Only the server <code>twilio</code> is accepted at moment.

Twilio payload

```
1  {
2    "From": "5551123456789",
3    "To": "5551987654321",
4    "Body": "SMS message",
5    "ToCountry": "Brazil",
6    "ToState": "RS",
7    "ToCity": "Porto Alegre",
8    "ToZip": "",
9    "FromCountry": "Brazil",
10   "FromState": "RS",
11   "FromCity": "Porto Alegre",
12   "FromZip": ""
```

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -X POST \
4   -H "Content-type:application/json" \
5   http://localhost:3000/api/v1/livechat/sms-incoming/twilio \
6   -d '{"From": "5551123456789", "To": "5551987654321", "Body": "SMS mess
```

Result to Twilio

```
<Response></Response>
```

Change Log

Version	Description
0.42.0	Added

Transcript

Request a Livechat transcript

URL	Requires Auth	HTTP Method
/api/v1/livechat/transcript	no	POST

Payload

Argument	Example	Required	Description
rid	XFzMqgn33DcsQkpJp	Required	The room <code>_id</code> .
token	iNKE8a6k6cjBqWhWd	Required	The visitor <code>token</code> .
email	visitor@rocket.chat	Required	The visitor <code>email</code> .

Example payload

```
1  {
2    "rid": "XFzMqgn33DcsQkpJp",
3    "token": "iNKE8a6k6cjBqWhWd",
4    "email": "visitor@rocket.chat"
5 }
```

Example Call

```
1 curl -X POST \
2     -H "Content-type:application/json" \
3     http://localhost:3000/api/v1/livechat/transcript \
4     -d '{"rid":"XFzMqgn33DcsQkpJp", "token": "iNKE8a6k6cjBqWhWd", "email"
```

Example Result

```
1 {
2     "message": "Livechat transcript sent",
3     "success": true
4 }
```

Change Log

Version	Description
0.70.0	Added

Triggers

Get the Livechat Triggers

Lists all Livechat triggers. It supports the [Offset, Count, and Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/livechat/triggers	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/livechat/triggers
```

Example Result

```
1 {
2     "triggers": [
3         {
4             "_id": "Lk52shJFYyb55trw8",
5             "name": "test",
6             "description": "test",
7             "enabled": true,
8             "runOnce": true,
9             "conditions": [
10                 {
11                     "name": "page-url",
12                     "value": ""
13                 }
14             ],
15             "actions": [
```

```
16          {
17              "name": "send-message",
18              "params": {
19                  "sender": "",
20                  "msg": ""
21              }
22          }
23      ],
24      "_updatedAt": "2019-10-04T15:36:29.695Z"
25  }
26 ],
27 "count": 1,
28 "offset": 0,
29 "total": 1,
30 "success": true
31 }
```

Change Log

Version	Description
2.2.0	Added

Get a Livechat Trigger

Get a Livechat Trigger by id.

URL	Requires Auth	HTTP Method
/api/v1/livechat/triggers/:_id	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/livechat/triggers/Lk52shJFYyb55trw8
```

Example Result

```
1  [
2    "trigger": {
3      "_id": "Lk52shJFYyb55trw8",
4      "name": "test",
5      "description": "test",
6      "enabled": true,
7      "runOnce": true,
8      "conditions": [
9        {
10          "name": "page-url",
11          "value": ""
12        }
13      ],
14      "actions": [
15        {
16          "name": "send-message",
17          "params": {
18            "sender": "",
19            "msg": ""
20          }
21        }
22      ],
23      "_updatedAt": "2019-10-04T15:36:29.695Z"
24    },
25    "success": true
26  }
```

Change Log

Version	Description

2.2.0

Added

Users

URL	Requires Auth	HTTP Method
/api/v1/livechat/users/:type	yes	GET

Payload

Argument	Example	Required	Description
type	agent	Required	Can be either agent or manager .

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2 -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3 http://localhost:3000/api/v1/livechat/users/agent
```

Example Result

```
1 {
2   "users": [
3     {
4       "_id": "aobEdbYhXfu5hkeqG",
5       "username": "john.doe"
6     },
7     {
8       "_id": "SQafHvoFPuB57NmBD",
9       "username": "doe.john"
```

```
10      }
11    ],
12    "success": true
13 }
```

Change Log

Version	Description
2.2.0	Added support to pagination
0.42.0	Added

Register new agent or manager

Register a new agent or manager.

URL	Requires Auth	HTTP Method
/api/v1/livechat/users/:type	yes	POST

Payload

Argument	Example	Required	Description
type	agent	Required	Can be either agent or manager .

Example payload

```
1  {
2    "username": "john.doe"
3 }
```

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -X POST \
4   -H "Content-type:application/json" \
5   http://localhost:3000/api/v1/livechat/users/agent \
6   -d '[{"username": "john.doe"}]
```

Example Result

```
1  {
2    "user": {
3      "_id": "SQafHvoFPuB57NmBD",
4      "username": "john.doe"
5    },
6    "success": true
7 }
```

Change Log

Version	Description
0.42.0	Added

Get info about an agent or manager

URL	Requires Auth	HTTP Method
/api/v1/livechat/users/:type/:_id	yes	GET

Payload

Argument	Example	Required	Description
type	agent	Required	Can be either <code>agent</code> or <code>manager</code> .
_id	SQafHvoFPuB57NmBD	Required	The user <code>_id</code> .

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/livechat/users/agent/SQafHvoFPuB57NmBD
```

Example Result

```
1 {
2   "user": {
3     "_id": "SQafHvoFPuB57NmBD",
4     "username": "john.doe"
5   },
6   "success": true
7 }
```

Change Log

Version	Description
0.42.0	Added

Removes an agent or manager

URL	Requires Auth	HTTP Method
/api/v1/livechat/users/:type/:_id	yes	DELETE

Payload

Argument	Example	Required	Description
type	agent	Required	Can be either <code>agent</code> or <code>manager</code> .
_id	SQafHvoFPuB57NmBD	Required	The user <code>_id</code> .

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -X DELETE \
4     http://localhost:3000/api/v1/livechat/users/agent/SQafHvoFPuB57NmBD
```

Example Result

```
1  {
2    "success": true
3 }
```

Change Log

Version	Description
0.42.0	Added

Video Call

Request a Livechat transcript

URL	Requires Auth	HTTP Method
/api/v1/livechat/video.call/:token	no	GET

Payload

Argument	Example	Required	Description
token	iNKE8a6k6cjbjqWhWd	Required	The visitor token

Query Parameters

Argument	Example	Required	Description
rid		Optional	The room _id .

Example Call

```
curl http://localhost:3000/api/v1/livechat/video.call?token=iNKE8a6k6cjbjqWhWd
```

Example Result

```
1  {
2    "videoCall": {
3      "rid": "vp3D9H8ud6HYvRpvq",
4      "domain": "meet.jit.si",
5      "provider": "jitsi",
6      "room": "RocketChatFsitsR8HptRxCmE4Qvp3D9H8ud6HYvRpvq",
7      "timeout": "2018-09-21T15:26:13.308Z"
8    },
9    "success": true
10 }
```

Change Log

Version	Description
0.70.0	Added

Visitors

Retrieve a visitor data

URL	Requires Auth	HTTP Method
/api/v1/livechat/visitor/:token	no	GET

Payload

Argument	Example	Required	Description
token	iNKE8a6k6cjbjqWhWd	Required	The visitor token .

Example Call

```
curl http://localhost:3000/api/v1/livechat/visitor/iNKE8a6k6cjbjqWhWd
```

Example Result

```
1  {
2    "visitor": {
3      "_id": "sGtcfEYz852uguxaS",
4      "username": "guest-7",
5      "_updatedAt": "2018-09-21T14:10:56.529Z",
6      "token": "iNKE8a6k6cjbjqWhWd",
7      "phone": [
8        {
9          "phoneNumber": "55 51 5555-5555"
```

```
10      }
11    ],
12    "visitorEmails": [
13      {
14        "address": "visitor@rocket.chat"
15      }
16    ],
17    "name": "Livechat Visitor",
18    "livechatData": {
19      "address": "Rocket.Chat street"
20    }
21  },
22  "success": true
23 }
```

Change Log

Version	Description
0.70.0	Added

Register a new Livechat visitor

URL	Requires Auth	HTTP Method
/api/v1/livechat/visitor	no	POST

Example payload

```
1  {
2    "visitor": {
3      "name": "Livechat Visitor",
4      "email": "visitor@rocket.chat",
```

```
5     "token": "iNKE8a6k6cjcbqWhWd",
6     "phone": "55 51 5555-5555",
7     "customFields": [
8       {
9         "key": "address",
10        "value": "Rocket.Chat street",
11        "overwrite": true
12      }
13    }
```

Example Call

```
1 curl -X POST \
2   -H "Content-type:application/json" \
3   http://localhost:3000/api/v1/livechat/visitor \
4   -d '{"visitor": {"name": "Livechat Visitor", "email": "visitor@rocket.chat"}' --insecure
```

Example Result

```
1  {
2    "visitor": {
3      "_id": "sGtcfEYz852uguxaS",
4      "username": "guest-7",
5      "_updatedAt": "2018-09-21T16:12:32.808Z",
6      "token": "iNKE8a6k6cjcbqWhWd",
7      "phone": [
8        {
9          "phoneNumber": "55 51 5555-5555"
10        }
11      ],
12      "visitorEmails": [
13        {
14          "address": "visitor@rocket.chat"
15        }
16      ],
17      "name": "Livechat Visitor",
18      "livechatData": {
```

```
19      "address": "Rocket.Chat street"
20    }
21  },
22  "success": true
23 }
```

Change Log

Version	Description
0.70.0	Added

Send Visitor navigation history

URL	Requires Auth	HTTP Method
/api/v1/livechat/page.visited	no	POST

Example payload

```
1  {
2    "token": "iNKE8a6k6cjBqWhWd",
3    "rid": "vp3D9H8ud6HYvRpVq",
4    "pageInfo": {
5      "change": "url",
6      "title": "",
7      "location": {
8        "href": "http://localhost:3000/packages/rocketchat_livechat/assets/d
9      }
10    }
11  }
```

Example Call

```
1 curl -X POST \
2     -H "Content-type:application/json" \
3     http://localhost:3000/api/v1/livechat/page.visited \
4     -d '{"token": "iNKE8a6k6cjBqWhWd", "rid": "vp3D9H8ud6HYvRpVq", "pageI}
```

Example Result

```
1 {
2     "page": {
3         "msg": " - http://localhost:3000/packages/rocketchat_livechat/assets/d
4         "navigation": {
5             "page": {
6                 "change": "url",
7                 "title": "",
8                 "location": {
9                     "href": "http://localhost:3000/packages/rocketchat_livechat/asse
10                }
11            },
12            "token": "iNKE8a6k6cjBqWhWd"
13        }
14    },
15    "success": true
16 }
```

Change Log

Version	Description
0.70.0	Added

Get the visitor info

Retrieve the visitor info

URL	Requires Auth	HTTP Method
/api/v1/livechat/visitors.info	yes	GET

Query Parameter

Argument	Example	Required	Description
visitorId	iNKE8a6k6cjbjqWhWd	Required	The visitor's id.

Example Call

```
curl http://localhost:3000/api/v1/livechat/visitors.info?visitorId=iNKE8a6k6
```

Example Result

```
1  {
2    "visitor": {
3      "_id": "L64iXAKtGaPfusnem",
4      "username": "guest-4",
5      "_updatedAt": "2019-10-21T19:36:47.960Z",
6      "token": "hzf48867bv9lwjzigk2tk"
7    },
8    "success": true
9  }
```

Change Log

Version	Description
2.3.0	Added

Get the visitor navigation history

Retrieve the visitor navigation history

URL	Requires Auth	HTTP Method
/api/v1/livechat/visitors.pagesVisited	yes	GET

Query Parameter

Argument	Example	Required	Description
roomId	iNKE8a6k6cjbjqWhWd	Required	The room's id.

Example Call

```
curl http://localhost:3000/api/v1/livechat/visitors.pagesVisited?roomId=iNKE
```

Example Result

```
1  {
2      "pages": [
3          {
4              "_id": "7sixPtdqiCcfSRrWJ",
5              "t": "livechat_navigation_history",
6              "rid": "wBCg9ZgXT7Mvqz5Xu",
7              "ts": "2019-10-21T19:51:22.298Z",
8              "msg": "App test - http://localhost:8601/",
9              "u": {
10                  "_id": "rocket.cat",
11                  "username": "rocket.cat"
12              },
13              "groupable": false,
14              "navigation": {
15                  "page": {
16                      "change": "url",
17                      "title": "Test",
18                      "location": {
19                          "href": "http://localhost:8601/"
20                      }
21                  },
22                  "token": "9vcfr6wb3sllv5t14d9x3"
23              },
24              "_hidden": true,
25              "_updatedAt": "2019-10-21T19:51:22.298Z"
26          }
27      ],
28      "success": true
29 }
```

Change Log

Version	Description
2.3.0	Added

Get the visitor chat history

Retrieve the visitor chat history. It supports the [Offset, Count, and Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/livechat/visitors.chatHistory/room/room-id/visitor/visitor-id	yes	GET

Query Parameter

Argument	Example	Required	Description
roomId	iNKE8a6k6cjbjqWhWd	Required	The room's id.
visitorId	iNKE8a6k6cjbjqWhWd	Required	The visitor's id.

Example Call

```
curl http://localhost:3000/api/v1/livechat/visitors.chatHistory/room/room-id/visitor/visitor-id
```

Example Result

```
1  {
2    "history": [
3      {
4        "_id": "Yjc62ZSoWiHB8Q6xr",
```

```

5      "msgs": 1,
6      "usersCount": 2,
7      "lm": "2019-12-09T20:30:32.058Z",
8      "fname": "Visitor 1575554334960",
9      "t": "l",
10     "ts": "2019-12-09T20:30:32.058Z",
11     "v": {
12       "_id": "sAvPE8bjCJP5YwPnN",
13       "username": "guest-2",
14       "token": "iNKE8a6k6cjBqWhWd",
15       "status": "online"
16     },
17     "cl": false,
18     "open": true,
19     "waitingResponse": true,
20     "_updatedAt": "2019-12-09T20:30:32.808Z",
21     "servedBy": {
22       "_id": "w6YjbdZHjghjfNBAG",
23       "username": "marcos.defendi",
24       "ts": "2019-12-09T20:30:32.166Z"
25     }
26   }
27 ],
28   "count": 1,
29   "offset": 0,
30   "total": 14,
31   "success": true
32 }
```

Change Log

Version	Description
2.4.0	Added

Search for Visitors

Search for a Visitor using their name, username, email or phone. It supports the [Offset](#), [Count](#), and [Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/livechat/visitors.search	yes	GET

Query Parameter

Argument	Example	Required	Description
term	guest-1@company.com	Optional	The visitor's name, username, email or phone

Example Call

```
curl http://localhost:3000/api/v1/livechat/visitors.search?term=guest-1@comp
```

Example Result

```
1  [
2    "visitors": [
3      {
4        "_id": "KQv3cHgvW7CDQtGap",
5        "username": "guest-1",
6        "visitorEmails": [
7          {
8            "address": "guest-1@company.com"
9          }
10        ],
11        "phone": [
12          {
13            "phoneNumber": "912235665456"
14          }
15        ]
16      }
17    ]
18  ]
```

```
15      ],
16      "name": "joey"
17    }
18  ],
19  "count": 1,
20  "offset": 0,
21  "total": 1,
22  "success": true
23 }
```

Change Log

Version	Description
3.6.0	Added

OAuthApps

Url	Short Description	Details Page
/api/v1/oauth-apps.get	Retrieves an OAuth App by id or client id.	Link
/api/v1/oauth-apps.list	Retrieves a list of OAuth Apps.	Link

Get

Retrieves an OAuth App by id or client id

URL	Requires Auth	HTTP Method
/api/v1/oauth-apps.get	yes	GET

Query Parameters

Argument	Example	Required	Description
appId	jn32kjnnnd0943j4njk	Required (if the clientId is not sent)	The app's id.
clientId	nkn8jk67b8b99b78	Required (if the appId is not sent)	The client's id.

Example

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/oauth-apps.get?appId=jn32kjnnnd0943j4njk
```

Example Result

```
1 {
2   "oauthApp": {
```

```
3     "_id": "zapier",
4     "name": "Zapier",
5     "active": true,
6     "clientId": "zapier",
7     "clientSecret": "RTK6TlndaCIolhQhZ7_KHIGOKj41RnlaOq_o-7JKwLr",
8     "redirectUri": "https://zapier.com/dashboard/auth/oauth/return/RocketC
9     "_createdAt": "2019-10-17T22:55:32.787Z",
10    "_createdBy": {
11        "_id": "system",
12        "username": "system"
13    },
14    "_updatedAt": "2019-10-17T22:55:32.787Z"
15 },
16    "success": true
17 }
```

Change Log

Version	Description
2.4.0	Added

List

Retrieves all OAuth Apps

URL	Requires Auth	HTTP Method
/api/v1/oauth-apps.list	yes	GET

Example

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/oauth-apps.list
```

Example Result

```
1 {
2   "oauthApps": [
3     {
4       "_id": "zapier",
5       "name": "Zapier",
6       "active": true,
7       "clientId": "zapier",
8       "clientSecret": "RTK6TlndaCIolhQhZ7_KHIGOKj41Rnla0q_o-7JKwLr",
9       "redirectUri": "https://zapier.com/dashboard/auth/oauth/return/Rocke",
10      "_createdAt": "2019-10-17T22:55:32.787Z",
11      "_createdBy": {
12        "_id": "system",
13        "username": "system"
14      },
15      "_updatedAt": "2019-10-17T22:55:32.787Z"
16    }
17  ],
18  "success": true
```

```
19 }
```

Change Log

Version	Description
2.4.0	Added

Miscellaneous

Just some generic information, such as information about the server and authenticated user.

Url	Short Description	Details Page
/api/v1/directory	Search by all users and channels available on server.	Link
/api/info	Information about the Rocket.Chat server.	Link
/api/v1/shield.svg	Gets the shield SVG (badge) to add in your website.	Link
/api/v1/spotlight	Searches for users or rooms that are visible to the user.	Link
/api/v1/statistics	Statistics about the Rocket.Chat server.	Link
/api/v1/statistics.list	Selectable statistics about the Rocket.Chat server.	Link
/api/v1/instances.get	Gets all running instances.	Link

Directory

A method, that searches by users or channels on all users and channels available on server. It supports the [Offset, Count, and Sort Query Parameters](#) along with [Query and Fields Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/directory	yes	GET

Payload

Argument	Example	Required	Description
query	{"text": "rocket", "type": "users", "workspace": "local"}	Required	When type is users you can send an additional workspace field, that can be local (default) or all . workspace=all will work only if Federation is enabled.

Example Call

```
1 curl -G -H "X-Auth-Token: ijFlJ1yfidXhwEYY284Anoq_iEs0eMMVCupzNhX22tB" \
2           -H "X-User-Id: hw5DThnhQmxDWnavu" \
3           -H "Content-type: application/json" \
4           http://localhost:3000/api/v1/directory \
5           --data-urlencode 'query={"text": "rocket", "type": "users", "wor'
```

Example Result

```
1  {
2      "result": [
3          {
4              "_id": "jRca8kibJx8NkLJxt",
5              "createdAt": "2018-04-13T12:46:26.517Z",
6              "emails": [
7                  {
8                      "address": "user.test.1523623548558@rocket.chat",
9                      "verified": false
10                 }
11             ],
12             "name": "EditedRealNameuser.test.1523623548558",
13             "username": "editedusernameuser.test.1523623548558",
14             "avatarETag": "6YbLtc4v9b4conXon"
15         }
16     ],
17     "count": 1,
18     "offset": 0,
19     "total": 1,
20     "success": true
21 }
```

Example Call

```
1 curl -H "X-Auth-Token: ijFlJ1yfidXhwEYY284Anoq_iEs0eMMVCupzNhX22tB" \
2           -H "X-User-Id: hw5DThnhQmxDWnavu" \
3           -H "Content-type: application/json" \
4           http://localhost:3000/api/v1/directory?query={"text": "gene", "type": "ch
```

Example Result

```
1  {
```

```
2     "result": [
3         {
4             "_id": "GENERAL",
5             "ts": "2018-05-15T19:10:54.689Z",
6             "name": "general",
7             "usernames": [
8                 "rocketchat.internal.admin.test",
9                 "editedusernameuser.test.1526941091574"
10            ]
11        }
12    ],
13    "count": 1,
14    "offset": 2,
15    "total": 4,
16    "success": true
17 }
```

Change Log

Version	Description
1.0.0	Added <code>workspace</code> query param
0.65.0	Added Pagination fields: <code>count</code> , <code>total</code> , <code>offset</code>
0.64.0	Added

Info

A simple method, requires no authentication, that returns information about the server including version information.

URL	Requires Auth	HTTP Method
/api/info	no	GET

Example Call

```
curl http://localhost:3000/api/info
```

Example Result

```
1  {
2    "success": true,
3    "info": {
4      "version": "0.47.0-develop",
5      "build": {
6        "nodeVersion": "v4.6.2",
7        "arch": "x64",
8        "platform": "linux",
9        "cpus": 4
10       },
11      "commit": {
12        "hash": "5901cc7270e3587101631ee222def950d705c611",
13        "date": "Thu Dec 1 19:08:01 2016 -0200",
14        "author": "Gabriel Engel",
15        "subject": "Merge branch 'develop' into experimental",
16        "tag": "0.46.0",
17        "branch": "experimental"
18      }
19    }
```

```
20  }
```

Change Log

Version	Description
0.13.0	Added

Shield SVG

A simple method, requires no authentication, that returns the shield svg(badge) to add in your website. It must have `API_Enable_Shields` enabled.

URL	Requires Auth	HTTP Method
/api/v1/shield.svg	no	GET

Query Parameters

Argument	Example	Required	Description
type	online	Optional	Can be one of online , user , channel
icon	false	Optional	
channel	general	Optional	Name of the channel
name	Rocket.Chat	Optional	Name to show

Example Call

```
curl http://localhost:3000/api/v1/shield.svg
```

Example Result

Headers

Content-Type : image/svg+xml; charset=utf-8

Body

```
1  <svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" style="width:100%; height:100%;">  
2      <linearGradient id="b" x2="0" y2="100%">  
3          <stop offset="0" stop-color="#bbb" stop-opacity=".1"/>  
4          <stop offset="1" stop-color="#fff" stop-opacity=".1"/>  
5      </linearGradient>  
6      <mask id="a">  
7          <rect width="141" height="20" rx="3" fill="#fff"/>  
8      </mask>  
9      <g mask=url(#a)>  
10         <path fill="#555" d="M0 0h67v20H0z"/>  
11         <path fill="#4c1" d="M67 0h74v20H67z"/>  
12         <path fill="url(#b)" d="M0 0h141v20H0z"/>  
13     </g>  
14     <image x="5" y="3" width="14" height="14" xlink:href="/assets/favicon.svg">  
15         <g fill="#fff" font-family="DejaVu Sans, Verdana, Geneva, sans-serif">  
16             <text x="24" y="15" fill="#010101" fill-opacity=".3">Rocket</text>  
17             <text x="24" y="14">Rocket</text>  
18             <text x="74" y="15" fill="#010101" fill-opacity=".3">JOIN CHAT</text>  
19             <text x="74" y="14">JOIN CHAT</text>  
20         </g>  
21     </svg>
```

Change Log

Version	Description
0.55.0	Added

Spotlight

Searches for users or rooms that are visible to the user.

WARNING: It will only return rooms that user didn't join yet.

URL	Requires Auth	HTTP Method
/api/v1/spotlight	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 6eWs4ECxUITYYoZ_bAYK50E2srkxAepQqQA1cGGv3jK" \
2     -H "X-User-Id: hvzu8z6mHFigiXy6Y" \
3     http://localhost:3000/api/v1/spotlight?query=foobar
```

Example Call

```
1 curl -H "X-Auth-Token: 6eWs4ECxUITYYoZ_bAYK50E2srkxAepQqQA1cGGv3jK" \
2     -H "X-User-Id: hvzu8z6mHFigiXy6Y" \
3     http://localhost:3000/api/v1/spotlight?query=@foobar
```

Example Call

```
1 curl -H "X-Auth-Token: 6eWs4ECxUITYYoZ_bAYK50E2srkxAepQqQA1cGGv3jK" \
2     -H "X-User-Id: hvzu8z6mHFigiXy6Y" \
3     http://localhost:3000/api/v1/spotlight?query=#foobar
```

Query Parameters

Argument	Example	Required	Description
query	john	Required	The term to be searched. Supports '#' for channels and '@' for users

Example Result

```
1  {
2    "users": [
3      {
4        "_id": "rocket.cat",
5        "name": "Rocket.Cat",
6        "username": "rocket.cat",
7        "status": "online",
8        "avatarETag": "5BB9B5ny5DkKdrwkq"
9      }
10     ],
11     "rooms": [],
12     "success": true
13   }
```

Change Log

Version	Description
0.64.0	Added support to '#' and '@' searches, for channels and users respectively.
0.61.0	Added.

Statistics

Statistics about the Rocket.Chat server. Requires `view-statistics` permission.

URL	Requires Auth	HTTP Method
<code>/api/v1/statistics</code>	yes	GET

Query Parameters

Argument	Example	Required	Description
<code>refresh</code>	<code>true</code>	Optional	Need to refresh statistics

Example Call

```
1 curl -H "X-Auth-Token: 8h2mKAwxB3AQrFSjLVKMooJyjdCFaA7W45sWlHP8Iz0" \
2   -H "X-User-Id: ew28FnZqipDpvKw3R" \
3   http://localhost:3000/api/v1/statistics
```

Example Result

```
1 {
2   "_id": "wufRdmSrjmSMhBdTN",
3   "uniqueId": "wD4EP3M7FeFzJZgk9",
4   "installedAt": "2018-02-18T19:40:45.369Z",
5   "version": "0.61.0-develop",
6   "totalUsers": 88,
```

```
7   "activeUsers":88,
8   "nonActiveUsers":0,
9   "onlineUsers":0,
10  "awayUsers":1,
11  "offlineUsers":87,
12  "totalRooms":81,
13  "totalChannels":41,
14  "totalPrivateGroups":37,
15  "totalDirect":3,
16  "totalLivechat":0,
17  "totalMessages":2408,
18  "totalChannelMessages":730,
19  "totalPrivateGroupMessages":1869,
20  "totalDirectMessages":25,
21  "totalLivechatMessages":0,
22  "lastLogin":"2018-02-24T12:44:45.045Z",
23  "lastMessageSentAt":"2018-02-23T18:14:03.490Z",
24  "lastSeenSubscription":"2018-02-23T17:58:54.779Z",
25  "os": {
26    "type":"Linux",
27    "platform":"linux",
28    "arch":"x64",
29    "release":"4.13.0-32-generic",
30    "uptime":76242,
31    "loadavg": [
32      0.0576171875,0.04638671875,0.00439453125
33    ],
34    "totalmem":5787901952,
35    "freemem":1151168512,
36    "cpus": [
37      {
38        "model":"Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
39        "speed":2405,
40        "times": {
41          "user":6437000,
42          "nice":586500,
43          "sys":1432200,
44          "idle":750117500,
45          "irq":0
46        }
47      },
48      {
49        "model":"Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
50        "speed":2405,
51        "times": {
52          "user":7319700,
53          "nice":268800,
54          "sys":1823600,
55          "idle":747642700,
56          "irq":0
57        }
58      }
59    ]
60  }
61  "cpu": [
62    {
63      "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
64      "speed": 2405,
65      "times": {
66        "user": 6437000,
67        "nice": 586500,
68        "sys": 1432200,
69        "idle": 750117500,
70        "irq": 0
71      }
72    }
73  ]
74  "memory": [
75    {
76      "type": "total",
77      "value": 5787901952
78    },
79    {
80      "type": "free",
81      "value": 1151168512
82    }
83  ]
84  "disk": [
85    {
86      "type": "total",
87      "value": 1000000000000
88    },
89    {
90      "type": "free",
91      "value": 100000000000
92    }
93  ]
94  "swap": [
95    {
96      "type": "total",
97      "value": 0
98    },
99    {
100      "type": "free",
101      "value": 0
102    }
103  ]
104  "processes": [
105    {
106      "name": "nodejs",
107      "count": 1
108    }
109  ]
110  "users": [
111    {
112      "name": "root",
113      "id": 0
114    }
115  ]
116  "groups": [
117    {
118      "name": "root"
119    }
120  ]
121  "files": [
122    {
123      "path": "/etc/hosts"
124    }
125  ]
126  "services": [
127    {
128      "name": "nodejs"
129    }
130  ]
131  "ports": [
132    {
133      "port": 3001
134    }
135  ]
136  "interfaces": [
137    {
138      "name": "eth0",
139      "ip": "192.168.1.100",
140      "mac": "00:0C:29:1A:0D:0A"
141    }
142  ]
143  "loadavg": [
144    0.0576171875,0.04638671875,0.00439453125
145  ]
146  "lastSeen": "2018-02-23T17:58:54.779Z"
147  "lastLogin": "2018-02-24T12:44:45.045Z"
148  "lastMessageSentAt": "2018-02-23T18:14:03.490Z"
149  "lastSeenSubscription": "2018-02-23T17:58:54.779Z"
150  "os": {
151    "type": "Linux",
152    "platform": "linux",
153    "arch": "x64",
154    "release": "4.13.0-32-generic",
155    "uptime": 76242,
156    "loadavg": [
157      0.0576171875,0.04638671875,0.00439453125
158    ],
159    "totalmem": 5787901952,
160    "freemem": 1151168512,
161    "cpus": [
162      {
163        "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
164        "speed": 2405,
165        "times": {
166          "user": 6437000,
167          "nice": 586500,
168          "sys": 1432200,
169          "idle": 750117500,
170          "irq": 0
171        }
172      }
173    ]
174  }
175  "cpu": [
176    {
177      "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
178      "speed": 2405,
179      "times": {
180        "user": 6437000,
181        "nice": 586500,
182        "sys": 1432200,
183        "idle": 750117500,
184        "irq": 0
185      }
186    }
187  ]
188  "memory": [
189    {
190      "type": "total",
191      "value": 5787901952
192    },
193    {
194      "type": "free",
195      "value": 1151168512
196    }
197  ]
198  "disk": [
199    {
200      "type": "total",
201      "value": 1000000000000
202    },
203    {
204      "type": "free",
205      "value": 100000000000
206    }
207  ]
208  "swap": [
209    {
210      "type": "total",
211      "value": 0
212    },
213    {
214      "type": "free",
215      "value": 0
216    }
217  ]
218  "processes": [
219    {
220      "name": "nodejs",
221      "count": 1
222    }
223  ]
224  "users": [
225    {
226      "name": "root",
227      "id": 0
228    }
229  ]
230  "groups": [
231    {
232      "name": "root"
233    }
234  ]
235  "files": [
236    {
237      "path": "/etc/hosts"
238    }
239  ]
240  "services": [
241    {
242      "name": "nodejs"
243    }
244  ]
245  "ports": [
246    {
247      "port": 3001
248    }
249  ]
250  "interfaces": [
251    {
252      "name": "eth0",
253      "ip": "192.168.1.100",
254      "mac": "00:0C:29:1A:0D:0A"
255    }
256  ]
257  "loadavg": [
258    0.0576171875,0.04638671875,0.00439453125
259  ]
260  "lastSeen": "2018-02-23T17:58:54.779Z"
261  "lastLogin": "2018-02-24T12:44:45.045Z"
262  "lastMessageSentAt": "2018-02-23T18:14:03.490Z"
263  "lastSeenSubscription": "2018-02-23T17:58:54.779Z"
264  "os": {
265    "type": "Linux",
266    "platform": "linux",
267    "arch": "x64",
268    "release": "4.13.0-32-generic",
269    "uptime": 76242,
270    "loadavg": [
271      0.0576171875,0.04638671875,0.00439453125
272    ],
273    "totalmem": 5787901952,
274    "freemem": 1151168512,
275    "cpus": [
276      {
277        "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
278        "speed": 2405,
279        "times": {
280          "user": 6437000,
281          "nice": 586500,
282          "sys": 1432200,
283          "idle": 750117500,
284          "irq": 0
285        }
286      }
287    ]
288  }
289  "cpu": [
290    {
291      "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
292      "speed": 2405,
293      "times": {
294        "user": 6437000,
295        "nice": 586500,
296        "sys": 1432200,
297        "idle": 750117500,
298        "irq": 0
299      }
300    }
301  ]
302  "memory": [
303    {
304      "type": "total",
305      "value": 5787901952
306    },
307    {
308      "type": "free",
309      "value": 1151168512
310    }
311  ]
312  "disk": [
313    {
314      "type": "total",
315      "value": 1000000000000
316    },
317    {
318      "type": "free",
319      "value": 100000000000
320    }
321  ]
322  "swap": [
323    {
324      "type": "total",
325      "value": 0
326    },
327    {
328      "type": "free",
329      "value": 0
330    }
331  ]
332  "processes": [
333    {
334      "name": "nodejs",
335      "count": 1
336    }
337  ]
338  "users": [
339    {
340      "name": "root",
341      "id": 0
342    }
343  ]
344  "groups": [
345    {
346      "name": "root"
347    }
348  ]
349  "files": [
350    {
351      "path": "/etc/hosts"
352    }
353  ]
354  "services": [
355    {
356      "name": "nodejs"
357    }
358  ]
359  "ports": [
360    {
361      "port": 3001
362    }
363  ]
364  "interfaces": [
365    {
366      "name": "eth0",
367      "ip": "192.168.1.100",
368      "mac": "00:0C:29:1A:0D:0A"
369    }
370  ]
371  "loadavg": [
372    0.0576171875,0.04638671875,0.00439453125
373  ]
374  "lastSeen": "2018-02-23T17:58:54.779Z"
375  "lastLogin": "2018-02-24T12:44:45.045Z"
376  "lastMessageSentAt": "2018-02-23T18:14:03.490Z"
377  "lastSeenSubscription": "2018-02-23T17:58:54.779Z"
378  "os": {
379    "type": "Linux",
380    "platform": "linux",
381    "arch": "x64",
382    "release": "4.13.0-32-generic",
383    "uptime": 76242,
384    "loadavg": [
385      0.0576171875,0.04638671875,0.00439453125
386    ],
387    "totalmem": 5787901952,
388    "freemem": 1151168512,
389    "cpus": [
390      {
391        "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
392        "speed": 2405,
393        "times": {
394          "user": 6437000,
395          "nice": 586500,
396          "sys": 1432200,
397          "idle": 750117500,
398          "irq": 0
399        }
400      }
401    ]
402  }
403  "cpu": [
404    {
405      "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
406      "speed": 2405,
407      "times": {
408        "user": 6437000,
409        "nice": 586500,
410        "sys": 1432200,
411        "idle": 750117500,
412        "irq": 0
413      }
414    }
415  ]
416  "memory": [
417    {
418      "type": "total",
419      "value": 5787901952
420    },
421    {
422      "type": "free",
423      "value": 1151168512
424    }
425  ]
426  "disk": [
427    {
428      "type": "total",
429      "value": 1000000000000
430    },
431    {
432      "type": "free",
433      "value": 100000000000
434    }
435  ]
436  "swap": [
437    {
438      "type": "total",
439      "value": 0
440    },
441    {
442      "type": "free",
443      "value": 0
444    }
445  ]
446  "processes": [
447    {
448      "name": "nodejs",
449      "count": 1
450    }
451  ]
452  "users": [
453    {
454      "name": "root",
455      "id": 0
456    }
457  ]
458  "groups": [
459    {
460      "name": "root"
461    }
462  ]
463  "files": [
464    {
465      "path": "/etc/hosts"
466    }
467  ]
468  "services": [
469    {
470      "name": "nodejs"
471    }
472  ]
473  "ports": [
474    {
475      "port": 3001
476    }
477  ]
478  "interfaces": [
479    {
480      "name": "eth0",
481      "ip": "192.168.1.100",
482      "mac": "00:0C:29:1A:0D:0A"
483    }
484  ]
485  "loadavg": [
486    0.0576171875,0.04638671875,0.00439453125
487  ]
488  "lastSeen": "2018-02-23T17:58:54.779Z"
489  "lastLogin": "2018-02-24T12:44:45.045Z"
490  "lastMessageSentAt": "2018-02-23T18:14:03.490Z"
491  "lastSeenSubscription": "2018-02-23T17:58:54.779Z"
492  "os": {
493    "type": "Linux",
494    "platform": "linux",
495    "arch": "x64",
496    "release": "4.13.0-32-generic",
497    "uptime": 76242,
498    "loadavg": [
499      0.0576171875,0.04638671875,0.00439453125
500    ],
501    "totalmem": 5787901952,
502    "freemem": 1151168512,
503    "cpus": [
504      {
505        "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
506        "speed": 2405,
507        "times": {
508          "user": 6437000,
509          "nice": 586500,
510          "sys": 1432200,
511          "idle": 750117500,
512          "irq": 0
513        }
514      }
515    ]
516  }
517  "cpu": [
518    {
519      "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
520      "speed": 2405,
521      "times": {
522        "user": 6437000,
523        "nice": 586500,
524        "sys": 1432200,
525        "idle": 750117500,
526        "irq": 0
527      }
528    }
529  ]
530  "memory": [
531    {
532      "type": "total",
533      "value": 5787901952
534    },
535    {
536      "type": "free",
537      "value": 1151168512
538    }
539  ]
540  "disk": [
541    {
542      "type": "total",
543      "value": 1000000000000
544    },
545    {
546      "type": "free",
547      "value": 100000000000
548    }
549  ]
550  "swap": [
551    {
552      "type": "total",
553      "value": 0
554    },
555    {
556      "type": "free",
557      "value": 0
558    }
559  ]
560  "processes": [
561    {
562      "name": "nodejs",
563      "count": 1
564    }
565  ]
566  "users": [
567    {
568      "name": "root",
569      "id": 0
570    }
571  ]
572  "groups": [
573    {
574      "name": "root"
575    }
576  ]
577  "files": [
578    {
579      "path": "/etc/hosts"
580    }
581  ]
582  "services": [
583    {
584      "name": "nodejs"
585    }
586  ]
587  "ports": [
588    {
589      "port": 3001
590    }
591  ]
592  "interfaces": [
593    {
594      "name": "eth0",
595      "ip": "192.168.1.100",
596      "mac": "00:0C:29:1A:0D:0A"
597    }
598  ]
599  "loadavg": [
600    0.0576171875,0.04638671875,0.00439453125
601  ]
602  "lastSeen": "2018-02-23T17:58:54.779Z"
603  "lastLogin": "2018-02-24T12:44:45.045Z"
604  "lastMessageSentAt": "2018-02-23T18:14:03.490Z"
605  "lastSeenSubscription": "2018-02-23T17:58:54.779Z"
606  "os": {
607    "type": "Linux",
608    "platform": "linux",
609    "arch": "x64",
610    "release": "4.13.0-32-generic",
611    "uptime": 76242,
612    "loadavg": [
613      0.0576171875,0.04638671875,0.00439453125
614    ],
615    "totalmem": 5787901952,
616    "freemem": 1151168512,
617    "cpus": [
618      {
619        "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
620        "speed": 2405,
621        "times": {
622          "user": 6437000,
623          "nice": 586500,
624          "sys": 1432200,
625          "idle": 750117500,
626          "irq": 0
627        }
628      }
629    ]
630  }
631  "cpu": [
632    {
633      "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
634      "speed": 2405,
635      "times": {
636        "user": 6437000,
637        "nice": 586500,
638        "sys": 1432200,
639        "idle": 750117500,
640        "irq": 0
641      }
642    }
643  ]
644  "memory": [
645    {
646      "type": "total",
647      "value": 5787901952
648    },
649    {
650      "type": "free",
651      "value": 1151168512
652    }
653  ]
654  "disk": [
655    {
656      "type": "total",
657      "value": 1000000000000
658    },
659    {
660      "type": "free",
661      "value": 100000000000
662    }
663  ]
664  "swap": [
665    {
666      "type": "total",
667      "value": 0
668    },
669    {
670      "type": "free",
671      "value": 0
672    }
673  ]
674  "processes": [
675    {
676      "name": "nodejs",
677      "count": 1
678    }
679  ]
680  "users": [
681    {
682      "name": "root",
683      "id": 0
684    }
685  ]
686  "groups": [
687    {
688      "name": "root"
689    }
690  ]
691  "files": [
692    {
693      "path": "/etc/hosts"
694    }
695  ]
696  "services": [
697    {
698      "name": "nodejs"
699    }
700  ]
701  "ports": [
702    {
703      "port": 3001
704    }
705  ]
706  "interfaces": [
707    {
708      "name": "eth0",
709      "ip": "192.168.1.100",
710      "mac": "00:0C:29:1A:0D:0A"
711    }
712  ]
713  "loadavg": [
714    0.0576171875,0.04638671875,0.00439453125
715  ]
716  "lastSeen": "2018-02-23T17:58:54.779Z"
717  "lastLogin": "2018-02-24T12:44:45.045Z"
718  "lastMessageSentAt": "2018-02-23T18:14:03.490Z"
719  "lastSeenSubscription": "2018-02-23T17:58:54.779Z"
720  "os": {
721    "type": "Linux",
722    "platform": "linux",
723    "arch": "x64",
724    "release": "4.13.0-32-generic",
725    "uptime": 76242,
726    "loadavg": [
727      0.0576171875,0.04638671875,0.00439453125
728    ],
729    "totalmem": 5787901952,
730    "freemem": 1151168512,
731    "cpus": [
732      {
733        "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
734        "speed": 2405,
735        "times": {
736          "user": 6437000,
737          "nice": 586500,
738          "sys": 1432200,
739          "idle": 750117500,
740          "irq": 0
741        }
742      }
743    ]
744  }
745  "cpu": [
746    {
747      "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
748      "speed": 2405,
749      "times": {
750        "user": 6437000,
751        "nice": 586500,
752        "sys": 1432200,
753        "idle": 750117500,
754        "irq": 0
755      }
756    }
757  ]
758  "memory": [
759    {
760      "type": "total",
761      "value": 5787901952
762    },
763    {
764      "type": "free",
765      "value": 1151168512
766    }
767  ]
768  "disk": [
769    {
770      "type": "total",
771      "value": 1000000000000
772    },
773    {
774      "type": "free",
775      "value": 100000000000
776    }
777  ]
778  "swap": [
779    {
780      "type": "total",
781      "value": 0
782    },
783    {
784      "type": "free",
785      "value": 0
786    }
787  ]
788  "processes": [
789    {
790      "name": "nodejs",
791      "count": 1
792    }
793  ]
794  "users": [
795    {
796      "name": "root",
797      "id": 0
798    }
799  ]
800  "groups": [
801    {
802      "name": "root"
803    }
804  ]
805  "files": [
806    {
807      "path": "/etc/hosts"
808    }
809  ]
810  "services": [
811    {
812      "name": "nodejs"
813    }
814  ]
815  "ports": [
816    {
817      "port": 3001
818    }
819  ]
820  "interfaces": [
821    {
822      "name": "eth0",
823      "ip": "192.168.1.100",
824      "mac": "00:0C:29:1A:0D:0A"
825    }
826  ]
827  "loadavg": [
828    0.0576171875,0.04638671875,0.00439453125
829  ]
830  "lastSeen": "2018-02-23T17:58:54.779Z"
831  "lastLogin": "2018-02-24T12:44:45.045Z"
832  "lastMessageSentAt": "2018-02-23T18:14:03.490Z"
833  "lastSeenSubscription": "2018-02-23T17:58:54.779Z"
834  "os": {
835    "type": "Linux",
836    "platform": "linux",
837    "arch": "x64",
838    "release": "4.13.0-32-generic",
839    "uptime": 76242,
840    "loadavg": [
841      0.0576171875,0.04638671875,0.00439453125
842    ],
843    "totalmem": 5787901952,
844    "freemem": 1151168512,
845    "cpus": [
846      {
847        "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
848        "speed": 2405,
849        "times": {
850          "user": 6437000,
851          "nice": 586500,
852          "sys": 1432200,
853          "idle": 750117500,
854          "irq": 0
855        }
856      }
857    ]
858  }
859  "cpu": [
860    {
861      "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
862      "speed": 2405,
863      "times": {
864        "user": 6437000,
865        "nice": 586500,
866        "sys": 1432200,
867        "idle": 750117500,
868        "irq": 0
869      }
870    }
871  ]
872  "memory": [
873    {
874      "type": "total",
875      "value": 5787901952
876    },
877    {
878      "type": "free",
879      "value": 1151168512
880    }
881  ]
882  "disk": [
883    {
884      "type": "total",
885      "value": 1000000000000
886    },
887    {
888      "type": "free",
889      "value": 100000000000
890    }
891  ]
892  "swap": [
893    {
894      "type": "total",
895      "value": 0
896    },
897    {
898      "type": "free",
899      "value": 0
900    }
901  ]
902  "processes": [
903    {
904      "name": "nodejs",
905      "count": 1
906    }
907  ]
908  "users": [
909    {
910      "name": "root",
911      "id": 0
912    }
913  ]
914  "groups": [
915    {
916      "name": "root"
917    }
918  ]
919  "files": [
920    {
921      "path": "/etc/hosts"
922    }
923  ]
924  "services": [
925    {
926      "name": "nodejs"
927    }
928  ]
929  "ports": [
930    {
931      "port": 3001
932    }
933  ]
934  "interfaces": [
935    {
936      "name": "eth0",
937      "ip": "192.168.1.100",
938      "mac": "00:0C:29:1A:0D:0A"
939    }
940  ]
941  "loadavg": [
942    0.0576171875,0.04638671875,0.00439453125
943  ]
944  "lastSeen": "2018-02-23T17:58:54.779Z"
945  "lastLogin": "2018-02-24T12:44:45.045Z"
946  "lastMessageSentAt": "2018-02-23T18:14:03.490Z"
947  "lastSeenSubscription": "2018-02-23T17:58:54.779Z"
948  "os": {
949    "type": "Linux",
950    "platform": "linux",
951    "arch": "x64",
952    "release": "4.13.0-32-generic",
953    "uptime": 76242,
954    "loadavg": [
955      0.0576171875,0.04638671875,0.00439453125
956    ],
957    "totalmem": 5787901952,
958    "freemem": 1151168512,
959    "cpus": [
960      {
961        "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
962        "speed": 2405,
963        "times": {
964          "user": 6437000,
965          "nice": 586500,
966          "sys": 1432200,
967          "idle": 750117500,
968          "irq": 0
969        }
970      }
971    ]
972  }
973  "cpu": [
974    {
975      "model": "Intel(R) Xeon(R) CPU E5620 @ 2.40GHz",
976      "speed": 2405,
977      "times": {
978        "user": 6437000,
979        "nice": 586500,
980        "sys": 1432200,
981        "idle": 750117500,
982        "irq": 0
983      }
984    }
985  ]
986  "memory": [
987    {
988      "type": "total",
989      "value": 5787901952
990    },
991    {
992      "type": "free",
993      "value": 1151168512
994    }
995  ]
996  "disk": [
997    {
998      "type": "total",
999      "value": 1000000000000
1000    },
1001    {
1002      "type": "free",
1003      "value": 100000000000
1004    }
1005  ]
1006  "swap": [
1007    {
1008      "type": "total",
1009      "value": 0
1010    },
1011    {
1012      "type": "free",
1013      "value": 0
1014    }
1015  ]
1016  "processes": [
1017    {
1018      "name": "nodejs",
1019      "count": 1
1020    }
1021  ]
1022  "users": [
1023    {
1024      "
```

```
58     },
59     {
60         "model":"Intel(R) Xeon(R) CPU           E5620 @ 2.40GHz",
61         "speed":2405,
62         "times": {
63             "user":7484600,
64             "nice":1003500,
65             "sys":1446000,
66             "idle":748873400,
67             "irq":0
68         }
69     },
70     {
71         "model":"Intel(R) Xeon(R) CPU           E5620 @ 2.40GHz",
72         "speed":2405,
73         "times": {
74             "user":8378200,
75             "nice":548500,
76             "sys":1443200,
77             "idle":747053300,
78             "irq":0
79         }
80     }
81 ],
82 },
83 "process": {
84     "nodeVersion":"v8.9.4",
85     "pid":11736,
86     "uptime":16265.506
87 },
88 "deploy": {
89     "method":"tar",
90     "platform":"selfinstall"
91 },
92 "migration": {
93     "_id":"control",
94     "version":106,
95     "locked":false,
96     "lockedAt":"2018-02-23T18:13:13.948Z",
97     "buildAt":"2018-02-18T17:22:51.212Z"
98 },
99 "instanceCount":1,
100 "createdAt":"2018-02-24T13:13:00.236Z",
101 "_updatedAt":"2018-02-24T13:13:00.236Z",
102 "success":true
103 }
```

Change Log

Version	Description
0.51.0	Added

Statistics List

Selectable statistics about the Rocket.Chat server. It supports the [Offset, Count and Sort Query Parameters](#) along with just the [Fields and Query Parameters](#). Requires `view-statistics` permission.

URL	Requires Auth	HTTP Method
<code>/api/v1/statistics.list</code>	yes	GET

Query Parameters

Argument	Example	Required	Description
query	<code>{"_id":"v3D4mvobwfznKozH8"}</code>	Required	See Query Parameter
offset	0	Optional	See Offset Parameter
count	1	Optional	See Count Parameter
fields	<code>{"os":0,"migration":0,"deploy":0,"process":0}</code>	Optional	See Fields Parameters
sort	<code>{"_id":1}</code>	Optional	See Sort Query Parameters

Example Call

A request below performs a statistic search for a server instance with

```
_id: v3D4mvobwfznKozH8 except response fields like os, migration, deploy, proces
```

```
1 curl -H "X-Auth-Token: 8h2mKAwxB3AQrFSjLVKMooJyjdCFaA7W45sWlHP8Iz0" \
2   -H "X-User-Id: ew28FnZqipDpvKw3R" \
3   "http://localhost:3000/api/v1/statistics.list?query=%7B%22_id%22%3A%2
```

Example Result

```
1 {
2   "statistics": [
3     {
4       "_id": "v3D4mvobwfznKozH8",
5       "uniqueId": "wD4EP3M7FeFzJZgk9",
6       "installedAt": "2018-02-18T19:40:45.369Z",
7       "version": "0.61.0-develop",
8       "totalUsers": 88,
9       "activeUsers": 88,
10      "nonActiveUsers": 0,
11      "onlineUsers": 0,
12      "awayUsers": 1,
13      "offlineUsers": 87,
14      "totalRooms": 81,
15      "totalChannels": 41,
16      "totalPrivateGroups": 37,
17      "totalDirect": 3,
18      "totalAllLivechat": 0,
19      "totalMessages": 2408,
20      "totalChannelMessages": 730,
21      "totalPrivateGroupMessages": 1869,
22      "totalDirectMessages": 25,
23      "totalAllLivechatMessages": 0,
24      "lastLogin": "2018-02-24T12:44:45.045Z",
25      "lastMessageSentAt": "2018-02-23T18:14:03.490Z",
26      "lastSeenSubscription": "2018-02-23T17:58:54.779Z",
27      "instanceCount": 1,
28      "createdAt": "2018-02-24T15:13:00.312Z",
29      "_updatedAt": "2018-02-24T15:13:00.312Z"
30     }
31   ],
32 }
```

```
32      "count":1,  
33      "offset":0,  
34      "total":1,  
35      "success":true  
36  }
```

Change Log

Version	Description
0.51.0	Added

Permissions

Url	Short Description	Details Page
/api/v1/permissions.listAll	Lists permissions on the server.	Link
/api/v1/permissions.update	Edits permissions on the server.	Link

List All

Returns all permissions from the server.

URL	Requires Auth	HTTP Method
/api/v1/permissions.listAll	yes	GET

Query Parameters

Argument	Example	Required	Description
updatedSince	2017-11-25T15:08:17.248Z	Optional	Date as ISO string

When you provide the `updatedSince` query parameter in the URL then the `update` and `remove` in the response will contain only those updated and removed since this date and time.

Example Call to get all of the rooms

```
1 curl -H "X-Auth-Token: ijFlJ1yfidXhwEYY284Anoq_iEs0eMMVCupzNhX22tB" \
2     -H "X-User-Id: hw5DThnhQmxDWnavu" \
3     -H "Content-type: application/json" \
4     http://localhost:3000/api/v1/permissions.listAll?updatedSince=2017-11
```

Example Result

```
1  {
2    "update": [
3      {
4        "_id": "access-permissions",
5        "_updatedAt": "2018-11-28T11:55:49.106Z",
6        "roles": [
7          "admin"
8        ]
9      },
10     {
11       "_id": "add-oauth-service",
12       "_updatedAt": "2018-11-28T12:59:51.974Z",
13       "roles": [
14         "admin",
15         "user"
16       ]
17     }
18   ],
19   "remove": [],
20   "success": true
21 }
```

Change Log

Version	Description
0.73.0	Added

Update

Edits permissions on the server.

URL	Requires Auth	HTTP Method
/api/v1/permissions.update	yes	POST

Payload

Argument	Example	Required	Description
permissions	[{"_id": "access-permissions", "roles": ["admin"]}]	Required	The array of roles to overwrite.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/permissions.update \
5   -d '{"permissions": [{"_id": "access-permissions", "roles": ["admin"]}]}
```

Example Result

```
1 {
2   "permissions": [
```

```
3      {
4          "_id": "access-permissions",
5          "roles": [
6              "admin"
7          ],
8          "_updatedAt": "2018-01-25T13:03:20.879Z",
9          "meta": {
10              "revision": 0,
11              "created": 1516980515815,
12              "version": 0
13          },
14          "$loki": 1
15      },{
16          ...
17      }
18  ],
19  "success": true
20 }
```

Change Log

Version	Description
0.66.0	Added.

Push Token

Url	Method	Short Description	Details Page
/api/v1/push.token	POST	Saves push token.	Link
/api/v1/push.token	DELETE	Removes push token.	Link

Delete

Removes push token on the server.

URL	Requires Auth	HTTP Method
/api/v1/push.token	yes	DELETE

Payload

Argument	Example	Required	Description
token	TOKEN	Required	The Token

Example Call

```
1 curl -H "X-Auth-Token: wQcX06fwN0w_syxZl40S67afcFJkKI6Ajz0JGA1-0-J" \
2   -H "X-User-Id: 52d5Rw8LT3TeDa59Z" \
3   -H "Content-type:application/json" \
4   -X "DELETE" \
5   http://localhost:3000/api/v1/push.token \
6   -d '{"token": "TOKEN"}'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.60.0	Added.

Post

Save new push token on the server.

URL	Requires Auth	HTTP Method
/api/v1/push.token	yes	POST

Payload

Argument	Example	Required	Description
id	id	Optional	A record id for the Application/Token document to update. If this does not exist, will return 404.
type	gcm	Required	Can be apn or gcm
value	TOKEN	Required	The Token
appName	com.example.rocketchat	Required	The name of the application to associate the token with

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/push.token \
5   -d '{"type": "gcm", "value": "TOKEN", "appName": "MyApp"}'
```

Example Result

```
1  {
2      "result": {
3          "token": {
4              "gcm": "TOKEN"
5          },
6          "appName": "RocketAPP",
7          "userId": "52d5Rw8LT3TeDa59Z",
8          "enabled": true,
9          "createdAt": "2018-09-11T18:22:55.006Z",
10         "updatedAt": "2018-09-11T18:22:55.006Z",
11         "_id": "DMMjKG2hFoLvM2hk7"
12     },
13     "success": true
14 }
```

Change Log

Version	Description
0.60.0	Added.

Roles

Url	Short Description	Details Page
/api/v1/roles.list	List all roles on the server.	Link
/api/v1/roles.sync	List all roles on the server which are updated after given date.	Link
/api/v1/roles.create	Create a new role.	Link
/api/v1/roles.addUserToRole	Edits permissions on the server.	Link
/api/v1/roles.getUsersInRole	Gets the users that belongs to a role.	Link

Role AddUserToRole

Assign a role to an user. Optionally, you can set this role to a room.

URL	Requires Auth	HTTP Method
/api/v1/roles.addUserToRole	yes	POST

Payload

Argument	Example	Required	Description
roleName	guest	Required	The role name.
username	rocket.chat	Required	The user name.
roomId	dK7vNYXMdHGLdukpL	Optional	If the role scope be Subscriptions and assign it to a room, you need to pass the roomId as parameter.

Example Call

```
1 curl -H "Content-type:application/json" \
2   -H "X-Auth-Token: 9HqLlyZ0ugoStsXCuF_0YdwnNnunAJF8V47U3QHXSq" \
3   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
4   http://localhost:3000/api/v1/roles.addUserToRole \
5   -d '{ "roleName": "guest" }' \
6   -d '{ "username": "rocket.chat" }' \
7   -d '{ "roomId": "dK7vNYXMdHGLdukpL" }'
```

Example Result

```
1  {
2      "role": {
3          "_id": "guest",
4          "name": "guest",
5          "scope": "Subscriptions",
6          "description": "",
7          "_updatedAt": "2018-08-23T03:57:54.603Z"
8      },
9      "success": true
10 }
```

Change Log

Version	Description
0.70.0	Added

Role Create

Create a new role in the system.

URL	Requires Auth	HTTP Method
/api/v1/roles.create	yes	POST

Payload

Argument	Example	Required	Description
name	newRole	Required	The name of the new role.
scope	Subscriptions	Optional Default: Users	The scope of the new role.
description	Role description	Optional	A description for the new role.

Example Call

```
1 curl -H "Content-type:application/json" \
2   -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
3   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
4   http://localhost:3000/api/v1/roles.create \
5   -d '{ "name": "newRole" }' \
6   -d '{ "scope": "Subscriptions" }' \
7   -d '{ "description": "Role description" }' \
```

Example Result

```
1  {
2      "role": {
3          "_id": "newRole",
4          "name": "newRole",
5          "scope": "Subscriptions",
6          "description": "Role description",
7          "_updatedAt": "2018-08-23T03:57:54.603Z"
8      },
9      "success": true
10 }
```

Change Log

Version	Description
0.70.0	Added

Role Get Users In Role

Gets the users that belongs to a role. It supports the [Offset and Count Only](#).

URL	Requires Auth	HTTP Method
/api/v1/roles.getUsersInRole	yes	GET

Query Parameters

Argument	Example	Required	Description
role	admin	Required	The role.
roomId	GENERAL	Optional	The room's id.

Example call to get all the roles

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/roles.getUsersInRole
```

Example Result

```
1 {
2   "users": [
3     {
4       "_id": "voakrL3cHjYBwwRPq",
```

```
5      "username": "a",
6      "type": "user",
7      "status": "offline",
8      "active": true,
9      "name": "a"
10     },
11     {
12       "_id": "N2s7KG6YkzgJfXbyn",
13       "username": "b",
14       "type": "user",
15       "status": "offline",
16       "active": true,
17       "name": "b"
18     },
19   ],
20   "success": true
21 }
```

Change Log

Version	Description
1.3.0	Added

Role List

Gets all the roles in the system.

URL	Requires Auth	HTTP Method
/api/v1/roles.list	yes	GET

Example call to get all the roles

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/roles.list
```

Example Result

```
1 {
2   "roles": [
3     {
4       "_id": "admin",
5       "description": "Admin",
6       "mandatory2fa": false,
7       "protected": true,
8       "scope": "Users"
9     },
10    {
11      "_id": "moderator",
12      "description": "Moderator",
13      "mandatory2fa": false,
14      "protected": true,
15      "scope": "Subscriptions"
16    },
17    {
18      "_id": "leader",
```

```
19     "description": "Leader",
20     "mandatory2fa": false,
21     "protected": true,
22     "scope": "Subscriptions"
23   },
24   {
25     "_id": "owner",
26     "description": "Owner",
27     "mandatory2fa": false,
28     "protected": true,
29     "scope": "Subscriptions"
30   },
31   {
32     "_id": "user",
33     "description": "",
34     "mandatory2fa": false,
35     "protected": true,
36     "scope": "Users"
37   },
38   {
39     "_id": "bot",
40     "description": "",
41     "mandatory2fa": false,
42     "protected": true,
43     "scope": "Users"
44   },
45   {
46     "_id": "guest",
47     "description": "",
48     "mandatory2fa": false,
49     "protected": true,
50     "scope": "Users"
51   },
52   {
53     "_id": "anonymous",
54     "description": "",
55     "mandatory2fa": false,
56     "protected": true,
57     "scope": "Users"
58   },
59   {
60     "_id": "livechat-agent",
61     "name": "livechat-agent",
62     "scope": "Users"
63   },
64   {
65     "_id": "livechat-manager",
66     "name": "livechat-manager",
67     "scope": "Users"
68   },
69   {
```

```
70      "_id": "livechat-guest",
71      "name": "livechat-guest",
72      "scope": "Users"
73    }
74  ],
75  "success": true
76 }
```

Change Log

Version	Description
0.73.0	Added mandatory2fa property
0.70.0	Added

Role Sync

Gets all the roles in the system which are updated after a given date.

URL	Requires Auth	HTTP Method
/api/v1/roles.sync	yes	GET

Query Parameters

Argument	Example	Required	Description
updatedSince	2017-11-25T15:08:17.248Z	Required	Date as ISO string

Example call to get all the roles after a given date

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/roles.sync?updatedSince=2017-11-25T15:08
```

Example Result

```
1 {
2   "roles": [
3     "update": [
4       {
5         "_id": "admin",
6         "description": "Admin",
7         "mandatory2fa": false,
```

```
8         "protected": true,
9         "scope": "Users"
10        }
11    ],
12    "remove" : [
13      {
14        "_id": "user",
15        "description": "User",
16        "mandatory2fa": false,
17        "protected": true,
18        "scope": "Users"
19      }
20    ]
21  },
22  "success": true
23 }
```

Change Log

Version	Description
3.1.0	Added

Rooms

Url	Method	Short Description	Details Page
/api/v1/rooms.adminRooms	GET	Retrieve all rooms (requires special permission).	Link
/api/v1/rooms.cleanHistory	POST	Cleans up a room's history, requires special permission.	Link
/api/v1/rooms.createDiscussion	POST	Creates a new discussion.	Link
/api/v1/rooms.favorite	POST	Favorite/Unfavorite room.	Link
/api/v1/rooms.get	GET	Gets rooms.	Link
/api/v1/rooms.getDiscussions	GET	Gets room's discussions.	Link
/api/v1/rooms.info	GET	Gets info from a room.	Link
/api/v1/rooms.leave	POST	Leaves a room.	Link
/api/v1/rooms.saveNotification	POST	Sets the notification settings of specific channel.	Link
/api/v1/rooms.upload/:rid	POST	Upload a message with attached file.	Link

Save Notification

Sets the notifications settings of specific channel.

URL	Requires Auth	HTTP Method
/api/v1/rooms.saveNotification	yes	POST

Payload

Argument	Example	Required	Description
roomId	5of4weEXaH7yncxz9	Required	The room id
notifications.desktopNotifications	nothing	Optional: 'nothing', 'all', 'mentions', 'default'	The desktop notifications
notifications.disableNotifications	'0'	Optional: '0' or '1'	The disable notifications
notifications.emailNotifications	nothing	Optional: 'nothing', 'all', 'mentions', 'default'	The email notifications

notifications.audioNotifications	nothing	Optional: 'nothing', 'all', 'mentions', 'default'
notifications.mobilePushNotifications	nothing	Optional: 'nothing', 'all', 'mentions', 'default'
notifications.audioNotificationValue	beep	Optional
notifications.desktopNotificationDuration	'0'	Optional
notifications.unreadAlert	nothing	Optional: 'nothing', 'all', 'mentions', 'default'
notifications.hideUnreadStatus	'0'	Optional: '0' or '1'

Example payload

```
1  {
2      "roomId": "5of4weEXaH7yncxz9",
3      "notifications" :{
4          "desktopNotifications": "all",
5          "disableNotifications": "0",
6          "emailNotifications": "nothing",
7          "audioNotificationValue": "beep",
8          "desktopNotificationDuration": "2",
9          "audioNotifications": "all",
10         "unreadAlert": "nothing",
11         "hideUnreadStatus": "all",
12         "mobilePushNotifications": "mentions"
13     }
14 }
```

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -H "Content-type: application/json" \
4     http://localhost:3000/api/v1/rooms.saveNotification \
5     -d '{"roomId": "5of4weEXaH7yncxz9", "notifications" :{ "desktopNotifi
6     "emailNotifications": "nothing", "audioNotificationValue": "beep", "d
7         "unreadAlert": "nothing", "hideUnreadStatus": "all", "mobilePushNot
```

Example Result

```
1  {
2      "success": true
3  }
```

Change Log

Version	Description
0.63.0	Added

Admin Rooms

Retrieves all rooms (requires the `view-room-administration` permission). It supports the **Offset, Count, and Sort Query Parameters**.

URL	Requires Auth	HTTP Method
<code>/api/v1/rooms.adminRooms</code>	yes	GET

Query Parameters

Argument	Example	Required	Description
<code>types</code>	<code>['c']</code>	Optional (must be an array)	The rooms types
<code>filter</code>	<code>general</code>	Optional	The rooms's name

Example

```
1 curl -H "X-Auth-Token: ijFlJ1yfidXhwEYY284Anoq_iEs0eMMVCupzNhX22tB" \
2     -H "X-User-Id: hw5DThnhQmxDwnavu" \
3     -H "Content-type: application/json" \
4     http://localhost:3000/api/v1/rooms.adminRooms?types[]=c&types[]=p&fil
```

Example Result

```
1 {
2     "rooms": [
```

```
3      {
4          "_id": "GENERAL",
5          "t": "c",
6          "name": "general",
7          "usernames": [],
8          "msgs": 98,
9          "usersCount": 38,
10         "default": true
11     },
12 ],
13 "count": 1,
14 "offset": 0,
15 "total": 88,
16 "success": true
17 }
```

Change Log

Version	Description
2.4.0	Added

Clean History

Cleans up a room, removing messages from the provided time range.

URL	Requires Auth	HTTP Method
/api/v1/rooms.cleanHistory	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The room's id
latest	2016-09-30T13:42:25.304Z	Required	The end of time range of messages to clean
oldest	2016-05-30T13:42:25.304Z	Required	The start of the time range of messages to clean
inclusive	true	Optional Default: false	Whether messages which land on latest and oldest should be included
excludePinned	true	Optional Default: false	Whether pinned messages should be deleted
filesOnly	true	Optional Default: false	Whether to only delete files and keep messages intact
users	["vynmera", "ggazzo"]	Optional Default: [] (everyone)	Specific set of users whose content to delete

limit	10	Optional Default: 0 (all)	The amount of messages to prune
ignoreDiscussion	true	Optional Default: true	Determines if messages from discussions should be pruned
ignoreThreads	true	Optional Default: true	Determines if messages from threads should be pruned

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/rooms.cleanHistory \
5   -d '{ "roomId": "roomId", "latest": "2016-12-09T13:42:25.304Z", "olde
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.64.0	Added
0.67.0	Added fields <code>limit</code> , <code>excludePinned</code> , <code>filesOnly</code> and <code>users</code>

Rooms Info

Retrieves the information about the room. It supports **Fields Query Parameters only.**

URL	Requires Auth	HTTP Method
/api/v1/rooms.info	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required (if no roomName)	The rooms's id
roomName	general	Required (if no roomId)	The rooms's name

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/rooms.info?roomId=ByehQjC44FwMeiLbX
```

Example Result

```
1 {
2   "room": {
3     "_id": "ByehQjC44FwMeiLbX",
4     "name": "testing",
```

```
5     "fname": "testing",
6     "t": "c",
7     "msgs": 0,
8     "usersCount": 2,
9     "u": {
10        "_id": "HKKPmF8rZh45GMHWH",
11        "username": "marcos.defendi"
12    },
13    "customFields": {},
14    "broadcast": false,
15    "encrypted": false,
16    "ts": "2020-05-21T13:14:07.070Z",
17    "ro": false,
18    "default": false,
19    "sysMes": true,
20    "_updatedAt": "2020-05-21T13:14:07.096Z"
21  },
22  "success": true
23 }
```

Example Call with fields query parameters

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/rooms.info?roomId=ByehQjC44FwMeiLbX&fiel
```

Example Result

```
1  {
2    "room": {
3      "_id": "ByehQjC44FwMeiLbX",
4      "name": "testing"
5    },
6    "success": true
7  }
```

Change Log

Version	Description
0.72.0	Added

Get Discussions

Get all the discussions of room. It supports the [Offset, Count, and Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/rooms.getDiscussions	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: ijFlJ1yfidXhwEYY284Anoq_iEs0eMMVCupzNhX22tB" \
2     -H "X-User-Id: hw5DThnhQmxDWnavu" \
3     -H "Content-type: application/json" \
4     http://localhost:3000/api/v1/rooms.getDiscussions?roomId=GENERAL
```

Example Result

```
1  {
2      "discussions": [
3          {
4              "_id": "hthLkno5RqTyNna4H",
5              "name": "8BRnQEJQM8MjJWxkJ",
6              "fname": "general discussion",
7              "t": "p",
8              "msgs": 1,
9              "usersCount": 1,
10             "u": {
11                 "_id": "rocketchat.internal.admin.test",
12                 "username": "rocketchat.internal.admin.test"
13             },
14             "topic": "general",
15             "prid": "GENERAL",
16             "ts": "2019-04-03T01:01:36.286Z",
17             "ro": false,
```

```
18         "sysMes": true,
19         "default": false,
20         "_updatedAt": "2019-04-03T01:01:36.402Z",
21         "lastMessage": {
22             "msg": "test",
23             "ts": "2019-04-03T01:01:36.339Z",
24             "u": {
25                 "_id": "rocketchat.internal.admin.test",
26                 "username": "rocketchat.internal.admin.test",
27                 "name": "RocketChat Internal Admin Test"
28             },
29             "rid": "hthLkno5RqTyNna4H",
30             "_id": "LLoaxBR8A7M24Qiji",
31             "_updatedAt": "2019-04-03T01:01:36.368Z",
32             "mentions": [],
33             "channels": []
34         },
35         "lm": "2019-04-03T01:01:36.339Z"
36     }
37 ],
38     "count": 1,
39     "offset": 0,
40     "total": 1,
41     "success": true
42 }
```

Change Log

Version	Description
1.0.0	Added

Get

Get all opened rooms (all joined public & private channels and all DMs) for this user.

URL	Requires Auth	HTTP Method
/api/v1/rooms.get	yes	GET

Query Parameters

Argument	Example	Required	Description
updatedSince	2017-11-25T15:08:17.248Z	Optional	Date as ISO string

When you provide the `updatedSince` query parameter in the URL then the `update` and `remove` in the response will contain only those updated and removed since this date and time.

Example Call to get all of the rooms

```
1 curl -H "X-Auth-Token: ijFlJ1yfidXhwEYY284Anoq_iEs0eMMVCupzNhX22tB" \
2     -H "X-User-Id: hw5DThnhQmxDWnavu" \
3     -H "Content-type: application/json" \
4     http://localhost:3000/api/v1/rooms.get
```

Example Result

```

1  {
2      "update": [
3          {
4              "_id": "GENERAL",
5              "name": "general",
6              "t": "c",
7              "_updatedAt": "2018-01-21T21:03:43.736Z",
8              "default": true
9          },
10         {
11             "_id": "3WpJQkDHhrWPBvXuWhw5DThnhQmxDWnavu",
12             "t": "d",
13             "_updatedAt": "2018-01-21T21:07:16.123Z"
14         },
15         {
16             "_id": "hw5DThnhQmxDWnavurocket.cat",
17             "t": "d",
18             "_updatedAt": "2018-01-21T21:07:18.510Z"
19         },
20         {
21             "_id": "hw5DThnhQmxDWnavuhw5DThnhQmxDWnavu",
22             "t": "d",
23             "_updatedAt": "2018-01-21T21:07:20.324Z"
24         },
25         {
26             "_id": "EAemRsyeykhfr9Stt",
27             "name": "123",
28             "fname": "123",
29             "t": "p",
30             "u": {
31                 "_id": "hw5DThnhQmxDWnavu",
32                 "username": "user2"
33             },
34             "_updatedAt": "2018-01-24T21:02:04.318Z",
35             "customFields": {},
36             "ro": false
37         }
38     ],
39     "remove": [],
40     "success": true
41 }

```

Example Call to get the rooms which have changed after the provided date

```
1 curl -H "X-Auth-Token: ijFlJ1yfidXhwEYY284Anoq_iEs0eMMVCupzNhX22tB" \
2   -H "X-User-Id: hw5DThnhQmxDWnavu" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/rooms.get?updatedSince=2018-01-26T00:11:
```

Example Result

```
1  {
2    "update": [
3      {
4        "_id": "GENERAL",
5        "name": "general",
6        "t": "c",
7        "_updatedAt": "2018-01-27T16:14:15.379Z",
8        "default": true
9      }
10    ],
11    "remove": [],
12    "success": true
13  }
```

Change Log

Version	Description
0.62.0	Added

Leave

Causes the caller to be removed from the room.

URL	Requires Auth	HTTP Method
/api/v1/rooms.leave	yes	POST

Payload

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The room's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/rooms.leave \
5   -d '{ "roomId": "ByehQjC44FwMeiLbX" }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.72.0	Added

Favorite

Favorite or unfavorite room.

URL	Requires Auth	HTTP Method
/api/v1/rooms.favorite	yes	POST

Payload

| roomId | GENERAL | Required | Room Id where to favorite or unfavorite. || roomName | general | Optional | Room Name where to favorite or unfavorite. || favorite | true/false | Required | The value to favorite(true)/unfavorite(false) room. |

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-Type: application/json" \
4   http://localhost:3000/api/v1/rooms.favorite \
5   -d '{ "roomId": "GENERAL", "favorite": true }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.64.0	Added

Upload

Post a message with attached file to a dedicated room.

URL	Requires Auth	HTTP Method
/api/v1/rooms.upload/:rid	yes	POST

Payload

Argument	Example	Required	Description
file	\$HOME/example.txt	Required	A file name to upload
msg	This is a message with a file	Optional	A message text
description	Simple text file	Optional	A description of the file
tmid	vg3h21v31v23	Optional	The thread message id (if you want upload a file to a thread)

Example Call

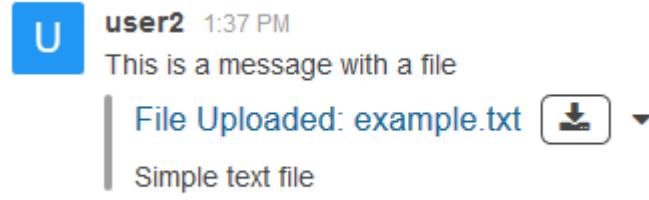
```
1 curl "http://localhost:3000/api/v1/rooms.upload/GENERAL" \
2     -F file=@$HOME/example.txt \
3     -F "msg=This is a message with a file" \
4     -F "description=Simple text file" \
5     -H "X-Auth-Token: ijFlJ1yfidXhwEYY284Anoq_iEs0eMMVCupzNhX22tB" \
6     -H "X-User-Id: hw5DThnhQmxDWnavu"
```

Note: For some file types if uploading via curl you may need to set the mime type.

With some file types, curl will upload the file as `application/octet-stream` . You can pass a custom mime type like this: `-F "file=@file.wav;type=audio/wav"`

Example Result

```
1  {
2      "message": {
3          "_id": "HxX7qtrm6azSaMChG",
4          "rid": "GENERAL",
5          "ts": "2019-03-24T20:28:02.501Z",
6          "msg": "This is a message with a file and I want to test it",
7          "file": {
8              "_id": "RwiWv3waLHfXLW3mj",
9              "name": "temp.txt",
10             "type": "text/plain"
11         },
12         "attachments": [
13             {
14                 "ts": "1970-01-01T00:00:00.000Z",
15                 "title": "temp.txt",
16                 "title_link": "/file-upload/RwiWv3waLHfXLW3mj/temp.txt",
17                 "title_link_download": true,
18                 "type": "file",
19                 "description": "Simple text file"
20             }
21         ],
22         "u": {
23             "_id": "abJaoNiibGs7gP9fe",
24             "username": "knrt10",
25             "name": "knrt10"
26         },
27         "_updatedAt": "2019-03-24T20:28:02.558Z"
28     },
29     "success": true
30 }
```



A picture with a message

Example HTTP dump

```
1 POST /api/v1/rooms.upload/GENERAL HTTP/1.1
2 Host: 192.168.2.71:3000
3 User-Agent: curl/7.45.0
4 Accept: */*
5 X-Auth-Token: ijFlJ1yfidXhwEYY284Anoq_iEs0eMMVCupzNhX22tB
6 X-User-Id: hw5DThnhQmxDWnavu
7 Content-Length: 457
8 Expect: 100-continue
9 Content-Type: multipart/form-data; boundary=-----f2ee26
10
11 -----f2ee26fd9d793455
12 Content-Disposition: form-data; name="file"; filename="example.txt"
13 Content-Type: text/plain
14
15 This text file contains a text.
16 -----f2ee26fd9d793455
17 Content-Disposition: form-data; name="msg"
18
19 This is a message with a file
20 -----f2ee26fd9d793455
21 Content-Disposition: form-data; name="description"
22
23 Simple text file
24 -----f2ee26fd9d793455--
25
26 HTTP/1.1 200 OK
27 Access-Control-Allow-Origin: *
28 X-Instance-ID: 6vZ3BLNJ8jnPZfYYx
29 Cache-Control: no-store
30 Pragma: no-cache
31 content-type: application/json
32 Vary: Accept-Encoding
```

```

33 Date: Sun, 28 Jan 2018 11:08:18 GMT
34 Connection: keep-alive
35 Transfer-Encoding: chunked
36
37 {
38     "message": {
39         "_id": "HxX7qtrm6azSaMChG",
40         "rid": "GENERAL",
41         "ts": "2019-03-24T20:28:02.501Z",
42         "msg": "This is a message with a file and I want to test it",
43         "file": {
44             "_id": "RwiWv3waLHfXLW3mj",
45             "name": "temp.txt",
46             "type": "text/plain"
47         },
48         "attachments": [
49             {
50                 "ts": "1970-01-01T00:00:00.000Z",
51                 "title": "temp.txt",
52                 "title_link": "/file-upload/RwiWv3waLHfXLW3mj/temp.txt",
53                 "title_link_download": true,
54                 "type": "file",
55                 "description": "Simple text file"
56             }
57         ],
58         "u": {
59             "_id": "abJaoNiibGs7gP9fe",
60             "username": "knrt10",
61             "name": "knrt10"
62         },
63         "_updatedAt": "2019-03-24T20:28:02.558Z"
64     },
65     "success": true
66 }
```

Change Log

Version	Description
1.0.0	Return Message object on file upload
0.62.0	Added

Create Discussion

Creates a new discussion for room. It requires at least one of the following permissions:

`start-discussion` OR `start-discussion-other-user` , AND must be with the following setting enabled: `Discussion_enabled` .

URL	Requires Auth	HTTP Method
<code>/api/v1/rooms.createDiscussion</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>prid</code>	<code>GENERAL</code>	Required	Parent room id of the discussion.
<code>t_name</code>	<code>discussion name</code>	Required	Discussion name.
<code>users</code>	<code>['rocket.cat']</code>	Optional	Array of users to join in the discussion, if not provide will be an empty array (Note: if provided, it must be an array).
<code>pmid</code>	<code>aobEgbghXfe543keqG</code>	Optional	Parent message id (if the discussion comes from a message).
<code>reply</code>	<code>reply of this discussion</code>	Optional	The reply of the discussion.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
```

```
3      -H "Content-Type: application/json" \
4      http://localhost:3000/api/v1/rooms.createDiscussion \
5      -d '{ "prid": "GENERAL", "t_name": "Discussion Name"}'
```

Example Result

```
1  {
2    "discussion": {
3      "rid": "cgk88DHLHexwMaFWh",
4      "name": "WJNEAM7W45wRYitHo",
5      "fname": "Discussion Name",
6      "t": "p",
7      "msgs": 0,
8      "usersCount": 0,
9      "u": {
10        "_id": "rocketchat.internal.admin.test",
11        "username": "rocketchat.internal.admin.test"
12      },
13      "topic": "general",
14      "prid": "GENERAL",
15      "ts": "2019-04-03T01:35:32.271Z",
16      "ro": false,
17      "sysMes": true,
18      "default": false,
19      "_updatedAt": "2019-04-03T01:35:32.280Z",
20      "_id": "cgk88DHLHexwMaFWh"
21    },
22    "success": true
23 }
```

Change Log

Version	Description
1.0.0	Added

Settings

You can get and update the settings via the REST API, only if you have permission to.

Url	Method	Short Description	Details Page
/api/v1/settings	GET	Lists all private settings.	Link
/api/v1/settings.public	GET	Lists all public settings.	Link
/api/v1/settings.oauth	GET	Return list of all available oauth services.	Link
/api/v1/service.configurations	GET	Lists all service configurations.	Link
/api/v1/settings/:_id	GET	Gets a setting.	Link
/api/v1/settings/:_id	POST	Updates a setting.	Link

oAuth

A simple method, requires no authentication, that returns list of all available oauth services.

URL	Requires Auth	HTTP Method
/api/v1/settings.oauth	no	GET

Example Call

```
curl http://localhost:3000/api/v1/settings.oauth
```

Example Result

```
1  {
2    "services": [
3      {
4        "_id": "iJeY7x4wxnh2p3pLr",
5        "name": "facebook",
6        "clientId": "test",
7        "buttonLabelText": "test",
8        "buttonColor": "#13679A",
9        "buttonLabelColor": "#FFFFFF",
10       "custom": false
11     },
12     {
13       "_id": "iJeY7x4wxnh2p3pLr",
14       "name": "twitter",
15       "clientId": "test",
16       "buttonLabelText": "",
17       "buttonColor": "",
18       "buttonLabelColor": "",
19       "custom": false
20     },
21   ]
```

```
21      {
22          "_id": "5RQ4SBL3NuZKsqxaF",
23          "service": "test",
24          "clientId": "test",
25          "custom": true,
26          "serverURL": "/test/test",
27          "tokenPath": "/oauth/token",
28          "identityPath": "/me",
29          "authorizePath": "/oauth/authorize",
30          "scope": "openid",
31          "buttonLabelText": "test",
32          "buttonLabelColor": "#FFFFFF",
33          "loginStyle": "popup",
34          "buttonColor": "#13679A",
35          "tokenSentVia": "payload",
36          "identityTokenSentVia": "default",
37          "usernameField": "dfsgdfgdfgdfgsd",
38          "mergeUsers": true
39      }
40  ],
41  "success": true
42 }
```

Change Log

Version	Description
0.64.0	Renamed field <code>appId</code> to <code>clientId</code> and added flag <code>custom</code> to indicate whether the oauth service is customized and fix <code>_id</code> inconsistence(set all cases to <code>_id</code>)
0.63.0	Added

0.64.0	Renamed field <code>appId</code> to <code>clientId</code> and added flag <code>custom</code> to indicate whether the oauth service is customized and fix <code>_id</code> inconsistence(set all cases to <code>_id</code>)
0.63.0	Added

Private Settings

List all private settings.

URL	Requires Auth	HTTP Method
/api/v1/settings	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/settings
```

Example Result

```
1 {
2   "settings": [
3     { "_id": "API_Allow_Infinite_Count", "value": true },
4     { "_id": "API_CORS_Origin", "value": "*" }
5   ],
6   "count": 50,
7   "offset": 0,
8   "total": 299,
9   "success": true
10 }
```

Public Settings Get

List all public settings.

It supports the [Offset, Count, and Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/settings.public	no	GET

Query Parameters

Argument	Example	Required	Description	Format
query	{"_id": {"\$in": ["LDAP_Enable", "Jitsi_Enabled"]}}	Optional	Query for named properties	URL encoded JSON object

Example Call

```
curl http://localhost:3000/api/v1/settings.public
```

Example Result

```
1  {
2    "settings": [
3      { "_id": "API_Drupal_URL", "value": "" },
```

```
4   { "_id": "API_EMBED", "value": true },  
5   "count": 50,  
6   "offset": 0,  
7   "total": 299,  
8   "success": true  
9 }
```

Service Configurations Get

List all service configurations.

URL	Requires Auth	HTTP Method
/api/v1/service.configurations	no	GET

Example Call

```
curl http://localhost:3000/api/v1/service.configurations
```

Example Result

```
1  {
2    "configurations": [],
3    "success": true
4 }
```

Settings Get

Gets the setting for the provided `_id`.

URL	Requires Auth	HTTP Method
/api/v1/settings/:_id	yes	GET

Payload

Argument	Example	Required	Description
_id	Livechat_enabled	Required	The setting's <code>_id</code> field

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/settings/Livechat_enabled
```

Example Result

```
1 {
2   "_id": "Livechat_enabled",
3   "value": false,
4   "success": true
5 }
```

Change Log

Version	Description
0.42.0	Added

Settings Update

Updates the setting for the provided `_id`.

The `_id` of a setting is the first argument of the `RocketChat.settings.add` method used in `Rocket.Chat/packages/rocketchat-lib/server/startup/settings.js` (among other files). For example, the following code in that `settings.js` file

```
1  this.add('Accounts_AllowAnonymousRead', false, {  
2    type: 'boolean',  
3    public: true  
4  });
```

means that the setting labeled "Allow anonymous read" in the section "Accounts" has `_id` equal to `'Accounts_AllowAnonymousRead'`. The second argument is the default value (`false`). The third argument specifies the variable's type and whether it is public, hidden, and so on. To set a color, you can send

```
1  {  
2    value: '<color-code>',  
3    editor: 'color'  
4  }
```

And also to trigger a action-button use:

```
1  {  
2    execute: true  
3  }
```

URL

Requires Auth

HTTP Method

`/api/v1/settings/:_id`

yes

POST

Payload

Argument	Example	Required	Description
_id	Livechat_enabled	Required	The setting's <code>_id</code> field
value	true	Required	The value to update the setting.
color	<color-code>	Optional	The color code to update the setting.
editor	true	Required if color was set	The editor key, with <code>color</code> value.
execute	true	Optional	To trigger a action-button

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/settings/Livechat_enabled \
5   -d '{ "value": true }'
```

Example Result

```
1  {
2    "success": true
3 }
```

Example Call

Setting a color

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/settings/Livechat_enabled \
5   -d '{ "value": "#ffffff", "editor": "color"}'
```

Example Result

```
1  {
2    "success": true
3 }
```

Example Call

Trigger an action

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
```

```
4      http://localhost:3000/api/v1/settings/Livechat_enabled \
5      -d '{ "value": "method", "execute": true}'
```

Example Result

```
1  {
2    "success": true
3 }
```

Change Log

Version	Description
0.65.0	Added option to set a color and trigger an action
0.42.0	Added

Subscriptions

Url	Short Description	Details Page
/api/v1/subscriptions.get	Get all subscriptions.	Link
/api/v1/subscriptions.getOne	Get the subscription by room Id.	Link
/api/v1/subscriptions.read	Mark a room as read.	Link
/api/v1/subscriptions.unread	Mark messages as unread.	Link

Get One

Get the subscription by room id.

URL	Requires Auth	HTTP Method
/api/v1/subscriptions.getOne	yes	GET

Query Parameters

Argument	Example	Required	Description
roomId	ByehQjC44FwMeiLbX	Required	The channel's id

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/subscriptions.getOne?roomId=5of4weExaH7y
```

Example Result

```
1 {
2   "subscription": {
3     "_id": "Rd8qisB7G4tF4Fibu",
4     "open": true,
5     "alert": false,
6     "unread": 0,
```

```

7      "userMentions": 0,
8      "groupMentions": 0,
9      "ts": "2018-03-12T17:52:13.041Z",
10     "rid": "5of4weEXaH7yncxz9",
11     "name": "test",
12     "fname": "test",
13     "customFields": {},
14     "t": "p",
15     "u": {
16       "_id": "47cRd58HnWwpqxhaZ",
17       "username": "rocket.cat",
18       "name": null
19     },
20     "ls": "2018-03-12T17:52:13.041Z",
21     "_updatedAt": "2018-03-13T19:36:27.696Z",
22     "roles": [
23       "owner"
24     ],
25     "disableNotifications": false,
26     "emailNotifications": "nothing",
27     "audioNotificationValue": "beep",
28     "desktopNotificationDuration": 2,
29     "audioNotifications": "all",
30     "mobilePushNotifications": "mentions",
31     "f": false,
32     "meta": {
33       "revision": 0,
34       "created": 1521051029632,
35       "version": 0
36     }
37   },
38   "success": true
39 }
```

Change Log

Version	Description
0.63.0	Added

Get

Get all subscriptions. Please note that the **unread** counter value depends on your settings at the *Administration => General* section.

URL	Requires Auth	HTTP Method
/api/v1/subscriptions.get	yes	GET

Query Parameters

Argument	Example	Required	Description	Valid Format(s)
updatedSince	2017-11-25T15:08:17.248Z	Optional	Date and time from which to fetch changes	ISO 8601 datetime. Optional seconds, optional milliseconds, optional timezone, always with colon time separators

When you provide the `updatedSince` query parameter in the URL then the `update` and `remove` in the response will contain only those updated and removed since this date and time.

Example Call to get all of the rooms

```
1 curl -H "X-Auth-Token: ijFlJ1yfidXhwEYY284Anoq_iEs0eMMVCupzNhX22tB" \
2 -H "X-User-Id: hw5DThnhQmxDWnavu" \
3 -H "Content-type: application/json" \
4 http://localhost:3000/api/v1/subscriptions.get?updatedSince=2017-11-2
```

Example Result

```
1  {
2    "update": [
3      {
4        "t": "c",
5        "ts": "2017-11-25T15:08:17.249Z",
6        "name": "general",
7        "fname": null,
8        "rid": "GENERAL",
9        "u": {
10          "_id": "EoyAmF4mxx5HxJHJB",
11          "username": "rocket.cat",
12          "name": "Rocket Cat"
13        },
14        "open": true,
15        "alert": true,
16        "unread": 1,
17        "userMentions": 1,
18        "groupMentions": 0,
19        "_updatedAt": "2017-11-25T15:08:17.249Z",
20        "_id": "5ALsG3QhpJfdMpyc8"
21      }
22    ],
23    "remove": [],
24    "success": true
25 }
```

Change Log

Version	Description
0.60.0	Added

Read

Mark any room (channel, group and DMs) as read.

URL	Requires Auth	HTTP Method
/api/v1/subscriptions.read	yes	POST

Payload

Argument	Example	Required	Description	Valid Format(s)
rid roomId	L6su9mxwdDkvdSaT9hvzu8z6mHFigiXy6Y OR GENERAL	Required	A valid roomId for a channel, group or DM	Alphanumeric mixed-case string

Example Call

Using `rid` as argument.

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-Type: application/json" \
4   http://localhost:3000/api/v1/subscriptions.read \
5   -d '{"rid":"L6su9mxwdDkvdSaT9hvzu8z6mHFigiXy6Y"}'
```

Example Result

```
1  {
2      "success": true
3 }
```

Example Call

Using `roomId` as argument.

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -H "Content-Type: application/json" \
4     http://localhost:3000/api/v1/subscriptions.read \
5     -d '{"roomId":"L6su9mxwdDkvdSaT9hvzu8z6mHFigiXy6Y"}'
```

Example Result

```
1  {
2      "success": true
3 }
```

Change Log

Version	Description

3.11.0 `roomId` can be used as argument instead of `rid`.

0.61.0 Endpoint Added.

Unread

Mark messages as unread by roomId or from a message. NOTE: You can send `roomId` or the id of the message (`firstUnreadMessage._id`) you want from it to be unread.

URL	Requires Auth	HTTP Method
/api/v1/subscriptions.unread	yes	POST

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -H "Content-Type: application/json" \
4     http://localhost:3000/api/v1/subscriptions.unread \
5     -d '{"roomId": "L6su9mxwdDkvdSaT9hvzu8z6mHFigiXy6Y"}'
```

Example Result

```
1 {
2     "success": true
3 }
```

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
```

```
3      -H "Content-Type: application/json" \
4      http://localhost:3000/api/v1/subscriptions.unread \
5      -d '{"firstUnreadMessage": { "_id": "L6su9mxwdDkvdSaT9hvzu8z6mHFigiXy'}
```

Example Result

```
1  {
2    "success": true
3 }
```

Change Log

Version	Description
0.65.0	Endpoint Added.

Teams

Url	Short Description	Details Page
/api/v1/teams.addMembers	Adds users to the team.	Link
/api/v1/teams.addRooms	Adds rooms to the team.	Link
/api/v1/teams.autocomplete	Lists the teams whose names match a given pattern.	Link
/api/v1/teams.create	Creates a new team.	Link
/api/v1/teams.delete	Removes a team.	Link
/api/v1/teams.info	Gets a team's information.	Link
/api/v1/teams.leave	Leaves a team.	Link
/api/v1/teams.list	Lists the public and private teams the caller is part of.	Link
/api/v1/teams.listAll	Lists all of the teams and their information.	Link
/api/v1/teams.listRooms	Lists all rooms of the team.	Link
/api/v1/teams.listRoomsOfUser	Lists only the team's rooms the user has joined.	Link
/api/v1/teams.members	Retrieves all team members.	Link
/api/v1/teams.removeMember	Removes a member from a team.	Link
/api/v1/teams.removeRoom	Removes a room from a team.	Link
/api/v1/teams.update	Updates an existing team (name and type).	Link

/api/v1/teams.updateMember

Updates a team member's roles, limited to permissions.

[Link](#)

/api/v1/teams.updateRoom

Updates a room from a team, limited to permissions.

[Link](#)

Add Members

Adds user to the team. Requires `add-team-member` or `edit-team-member` permissions.

URL	Requires Auth	HTTP Method
<code>/api/v1/teams.addMembers</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>teamId</code> <code>teamName</code>	<code>ByehQjC44FwMeiLbX</code> <code>team1</code>	Required	The team id. The team name.
<code>members</code>	<code>[{ "userId": "8dugqGhuRvCBLdZft", "roles": ["member"] }]</code>	Required	The users (ids and roles) to add to the team.

Example Call

```
1 curl -H "X-Auth-Token: 8h2mKAwxB3AQrFSjLVKMooJyjdCFaA7W45sWlHP8Iz0" \
2 -H "X-User-Id: ew28FnZqipDpvKw3R" \
3 -H "Content-type: application/json" \
4 http://localhost:3000/api/v1/teams.addmembers \
5 -d '{ "teamId": "Isjsldk32sdkfA", "members": [{ userId: "aobEdbYhXfu" }]}'
```

Example Result

```
1  {
2    "success": true
3 }
```

Add Rooms

Adds rooms to the team. Requires `add-team-channel` permission.

URL	Requires Auth	HTTP Method
<code>/api/v1/teams.addRooms</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>teamId</code>	<code>ByehQjC44FwMeiLbX</code>		The team id. The team name.
<code>teamName</code>	<code>team1</code>	Required	
<code>rooms</code>	<code>["8dugqGhuRvCBLdZft"]</code>	Required	The ids of the rooms to be added to the team.

Example Call

```
1 curl -H 'X-Auth-Token: 8h2mKAwxB3AQrFSjLVKMooJyjdCFaA7W45sWlHP8IzO' \
2 -H 'X-User-Id: 8dugqGhuRvCBLdZft' \
3 -H 'Content-Type: application/json' \
4 http://localhost:3000/api/v1/teams.addRooms \
5 -d '{ "teamId": "607e0d9b49d493189836bfac", "rooms": ["8Z7eRsibvD5AAN
```

Example Result

```
1  {
2    "rooms": [
3      {
4        "_id": "8Z7eRsibvD5AANfmK",
5        "name": "channelToAddToTeam",
6        "fname": "channelToAddToTeam",
7        "t": "c",
8        "msgs": 0,
9        "usersCount": 1,
10       "u": {
11         "_id": "8dugqGhuRvCBLdZft",
12         "username": "some.username"
13       },
14       "customFields": {},
15       "description": "",
16       "broadcast": false,
17       "encrypted": false,
18       "ts": "2021-04-27T19:59:07.258Z",
19       "ro": false,
20       "_updatedAt": "2021-04-27T19:59:07.313Z",
21       "teamId": "607e0d9b49d493189836bfa"
22     }
23   ],
24   "success": true
25 }
```

Autocomplete

List the teams whose names match a given pattern.

URL	Requires Auth	HTTP Method
/api/v1/teams.autocomplete	yes	GET

Query Parameters

Argument	Example	Required	Description
name	team	Required	The pattern (search filter for team names).

Example Call

```
1 curl -H 'X-Auth-Token: _2u_4MzRroRcnqc59GYUY_Kwgr9HgtZ9HCKn-2aIvMJ' \
2   -H 'X-User-Id: FL2fZL4ERhwA3gWiS' \
3   -H "Content-type: application/json" \
4   'http://localhost:3000/api/v1/teams.autocomplete?name=team'
```

Example Result

```
1 {
2   "teams": [
3     {
4       "_id": "Dgh2xwJ3NFKWvKSqY",
5       "name": "team1",
6       "fname": "team1",
```

```
7      "t": "p",
8      "teamId": "607e0d9b49d493189836bfac"
9    }
10   ],
11   "success": true
12 }
```

Create

Creates a new team. Requires `create-team` permission.

URL	Requires Auth	HTTP Method
<code>/api/v1/teams.create</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>name</code>	<code>TeamName</code>	Required	The team name.
<code>type</code>	<code>0 or 1</code>	Required	Privacy of the team (0 - Public, 1 - Private).
<code>members</code>	<code>["8dugqGhuRvCBLdZft"]</code>	Optional	The user ids to add to the team when it is created.
<code>room</code>	<code>{ readOnly: true }</code>	Optional	Additional options used to create the main team channel.

Example Call

```
1 curl -H "X-Auth-Token: 8h2mKAwxB3AQrFSjLVKMooJyjdCFaA7W45sWlHP8IzO" \
2 -H "X-User-Id: ew28FnZqipDpvKw3R" \
3 -H "Content-type: application/json" \
4 http://localhost:3000/api/v1/teams.create \
5 -d '{ "name": "teamName", "type": 0 }'
```

Example Result

```
1  {
2    "team": {
3      "_id": "608876e583314b02c8e6d857",
4      "name": "teamName",
5      "type": 0,
6      "createdAt": "2021-04-27T20:41:09.738Z",
7      "createdBy": {
8        "_id": "FL2fZL4ERhwA3gWiS",
9        "username": "some.username"
10      },
11      "_updatedAt": "2021-04-27T20:41:09.738Z",
12      "roomId": "yv9KEJdTFjmerZprx"
13    },
14    "success": true
15  }
```

Note: `roomId` is the team's main room's id.

Delete

Removes a team. Requires `delete-team` permission.

URL	Requires Auth	HTTP Method
/api/v1/teams.delete	yes	POST

Payload

Argument	Example	Required	Description
teamId	ByehQjC44FwMeiLbX	Required	The team id. The team name.
teamName	team1		
roomsToRemove	["8dugqGhuRvCBLdZft"]	Optional	The rooms (ids) to delete along with team.

Example Call

```
1 curl -H "X-Auth-Token: 8h2mKAwxB3AQrFSjLVKMooJyjdCFaA7W45sWlHP8Iz0" \
2 -H "X-User-Id: ew28FnZqipDpvKw3R" \
3 -H "Content-type: application/json" \
4 http://localhost:3000/api/v1/teams.delete \
5 -d '{ "teamId": "Isjsldk32sdkfA" }'
```

Example Result

```
1  {
2    "success": true
3 }
```

Info

Gets a team's information. If the team is not public, the caller user must be a member of the team.

URL	Requires Auth	HTTP Method
/api/v1/teams.info	yes	GET

Query Parameters

Argument	Example	Required	Description
teamId	ByehQjC44FwMeiLbX		The team id.
teamName	team1	Required	The team name.

Example Call

```
1 curl -H "X-Auth-Token: 8h2mKAwxB3AQrFSjLVKMooJyjdCFaA7W45sWlHP8IzO" \
2   -H "X-User-Id: ew28FnZqipDpvKw3R" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/teams.info?teamName=Team1
```

Example Result

```
1 {
2   "teamInfo": {
3     "_id": "607e0d9b49d493189836bfac",
```

```
4      "name": "Team1",
5      "type": 1,
6      "createdAt": "2021-04-19T23:09:15.106Z",
7      "createdBy": {
8          "_id": "FL2fZL4ERhwA3gWiS",
9          "username": "some.username"
10     },
11     "_updatedAt": "2021-04-19T23:09:15.106Z",
12     "roomId": "Dgh2xwJ3NFKWvKSqY"
13   },
14   "success": true
15 }
```

Note: `roomId` is the team's main room's id.

Leave

Causes the caller to be removed from the team.

URL	Requires Auth	HTTP Method
/api/v1/teams.leave	yes	POST

Payload

Argument	Example	Required	Description
teamId	ByehQjC44FwMeiLbx	Required	The team id.
teamName	team1	Required	The team name.
rooms	["8dugqGhuRvCBLdZft"]	Optional	The team rooms (ids) to remove the caller user from.

Example Call

```
1 curl -H "X-Auth-Token: 8h2mKAwxB3AQrFSjLVKMooJyjdCFaA7W45sWlHP8Iz0" \
2 -H "X-User-Id: ew28FnZqipDpvKw3R" \
3 -H "Content-type: application/json" \
4 http://localhost:3000/api/v1/teams.leave \
5 -d '{ "teamName": "team1" }'
```

Example Result

```
1  {
2    "success": true
3 }
```

List

Lists the public and private teams the caller is part of. It supports the [Offset](#), [Count](#), and [Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/teams.list	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 8h2mKAwxB3AQRFSjLVKMooJyjdCFaA7W45sWlHP8Iz0" \
2   -H "X-User-Id: ew28FnZqipDpvKw3R" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/teams.list
```

Example Result

```
1 {
2   "teams": [
3     {
4       "_id": "607e0d9b49d493189836bfac",
5       "name": "team1",
6       "type": 1,
7       "createdAt": "2021-04-19T23:09:15.106Z",
8       "createdBy": {
9         "_id": "ew28FnZqipDpvKw3R",
10        "username": "some.username"
11      },
12      "_updatedAt": "2021-04-19T23:09:15.106Z",
13      "roomId": "Dgh2xwJ3NFKWvKSqY",
14      "rooms": 2,
15      "numberOfUsers": 2
16    }
```

```
17  ],
18  "total": 1,
19  "count": 1,
20  "offset": 0,
21  "success": true
22 }
```

Note: `roomId` is the team's main room's id.

List All

Lists all of the teams and their information. Requires `view-all-teams` permission. It supports the [Offset and Count Parameters](#).

URL	Requires Auth	HTTP Method
<code>/api/v1/teams.listAll</code>	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 8h2mKAwxB3AQrFSjLVKMooJyjdCFaA7W45sWlHP8IzO" \
2   -H "X-User-Id: ew28FnZqipDpvKw3R" \
3   -H "Content-type: application/json" \
4   http://localhost:3000/api/v1/teams.listAll
```

Example Result

```
1  {
2    "teams": [
3      {
4        "_id": "Isjsldk32sdkfA",
5        "name": "team1",
6        "type": 0,
7        "createdAt": "2016-05-30T13:42:25.304Z",
8        "createdBy": {
9          "_id": "aobEdbYhXfu5hkeqG",
10         "username": "some.username"
11       },
12       "_updatedAt": "2020-05-21T13:14:07.096Z",
13       "roomId": "Dgh2xwJ3NFKWvKSqY",
14       "rooms": 2,
15       "numberOfUsers": 2
16     },
17   ]
```

```
17      {
18          "_id": "asdf97xcv87",
19          "name": "team2",
20          "type": 1,
21          "createdAt": "2016-05-30T13:42:25.304Z"
22          "createdBy": {
23              "_id": "aobEdbYhXfu5hkeqG",
24              "username": "example"
25          },
26          "_updatedAt": "2020-05-21T13:14:07.096Z",
27          "roomId": "Fda3TghyU03WqsYEv",
28          "rooms": 5,
29          "numberOfUsers": 4
30      }
31 ],
32 "total": 2,
33 "count": 2,
34 "offset": 0,
35 "success": true
36 }
```

Note: `roomId` is the team's main room's id.

List Rooms

List all rooms of the team. It supports the [Offset and Count Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/teams.listRooms	yes	GET

Query Parameters

Argument	Example	Required	Description
teamId	ByehQjC44FwMeiLbX	Required	The team id.
teamName	team1	Required	The team name.
filter	Example Room Name	Optional	The room's name (search filter).
type	autoJoin	Optional	The room's type (search filter).

Example Call

```
1 curl -H 'X-Auth-Token: _2u_4MzRroRcnqc59GYUY_Kwgr9HgtZ9HCKn-2aIvMJ' \
2 -H 'X-User-Id: FL2fZL4ERhwA3gWiS' \
3 -H "Content-type: application/json" \
4 'http://localhost:3000/api/v1/teams.listRooms?teamName=team1&type=au'
```

Example Result

```
1  {
2    "rooms": [
3      {
4        "_id": "8Z7eRsibvD5AANfmK",
5        "name": "channelAutoJoin",
6        "fname": "channelAutoJoin",
7        "t": "c",
8        "msgs": 7,
9        "usersCount": 1,
10       "u": {
11         "_id": "FL2fZL4ERhwA3gWiS",
12         "username": "some.username"
13       },
14       "customFields": {},
15       "description": "",
16       "broadcast": false,
17       "encrypted": false,
18       "ts": "2021-04-27T19:59:07.258Z",
19       "ro": false,
20       "_updatedAt": "2021-04-28T20:57:18.362Z",
21       "teamId": "607e0d9b49d493189836bfa",
22       "teamDefault": true
23     }
24   ],
25   "total": 1,
26   "count": 1,
27   "offset": 0,
28   "success": true
29 }
```

List Rooms Of User

Lists only the team's rooms the user has joined. Requires [view-all-team-channels](#) permission. It supports the [Offset and Count Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/teams.listRoomsOfUser	yes	GET

Query Parameters

Argument	Example	Required	Description
teamId	ByehQjC44FwMeiLbX	Required	The team id.
teamName	team1	Required	The team name.
userId	LrTDRE27fWPEvaAwC	Required	The user id (the endpoint returns the team channels this user has joined).

Example Call

```
1 curl -H 'X-Auth-Token: _2u_4MzRroRcnqc59GYUY_Kwgr9HgtZ9HCKn-2aIvMJ' \
2 -H 'X-User-Id: FL2fZL4ERhwA3gWiS' \
3 -H "Content-type: application/json" \
4 'http://localhost:3000/api/v1/teams.listRoomsOfUser?teamName=team1&u'
```

Example Result

```
1  {
2    "rooms": [
3      {
4        "_id": "8Z7eRsibvD5AANfmK",
5        "name": "someChannel",
6        "fname": "someChannel",
7        "t": "c",
8        "msgs": 7,
9        "usersCount": 1,
10       "u": {
11         "_id": "FL2fZL4ERhwA3gWiS",
12         "username": "some.username"
13       },
14       "customFields": {},
15       "description": "",
16       "broadcast": false,
17       "encrypted": false,
18       "ts": "2021-04-27T19:59:07.258Z",
19       "ro": false,
20       "_updatedAt": "2021-04-28T20:57:18.362Z",
21       "teamId": "607e0d9b49d493189836bfac",
22       "teamDefault": true,
23       "isLastOwner": true
24     }
25   ],
26   "total": 1,
27   "count": 1,
28   "offset": 0,
29   "success": true
30 }
```

Members

Lists the public and private teams the caller is part of. It supports the [Offset](#), [Count](#), and [Sort Query Parameters](#).

URL	Requires Auth	HTTP Method
/api/v1/teams.members	yes	GET

Query Parameters

Argument	Example	Required	Description
teamId	ByehQjC44FwMeiLbX	Required	The team id. The team name.
teamName	team1	Required	The team name.
name	Example Name	Optional	The user's name (search filter).
username	some.username	Optional	The username (search filter).
status	["online", "away", "busy"]	Optional	The user's status (search filter).

Example Call

```
1 curl -H 'X-Auth-Token: _2u_4MzRroRcnqc59GYUY_Kwgr9HgtZ9HCKn-2aIvMJ' \
2 -H 'X-User-Id: FL2fZL4ERhwA3gWiS' \
3 -H "Content-type: application/json" \
4 'http://localhost:3000/api/v1/teams.members?teamName=team1&status[]='
```

Example Result

```
1  {
2      "members": [
3          {
4              "user": {
5                  "_id": "FL2fZL4ERhwA3gWiS",
6                  "username": "some.username",
7                  "name": "Example Name",
8                  "status": "online"
9              },
10             "roles": [
11                 "owner"
12             ],
13             "createdBy": {
14                 "_id": "FL2fZL4ERhwA3gWiS",
15                 "username": "some.username"
16             },
17             "createdAt": "2021-04-19T23:09:15.109Z"
18         }
19     ],
20     "total": 3,
21     "count": 1,
22     "offset": 0,
23     "success": true
24 }
```

Remove Member

Removes a member from a team. Requires `edit-team-member` permission.

URL	Requires Auth	HTTP Method
/api/v1/teams.removeMember	yes	POST

Payload

Argument	Example	Required	Description
teamId	ByehQjC44FwMeiLbX	Required	The team id.
teamName	team1	Required	The team name.
userId	ew28FnZqipDpvKw3R	Required	The user id of who to remove from the team.
rooms	["8dugqGhuRvCBLdZft"]	Optional	The rooms (ids) from which the user should be removed.

Example Call

```
1 curl -H 'X-Auth-Token: _2u_4MzRroRcnqc59GYUY_Kwgr9HgtZ9HCKn-2aIvMJ' \
2 -H 'X-User-Id: FL2fZL4ERhwA3gWiS' \
3 -H "Content-type: application/json" \
4 http://localhost:3000/api/v1/teams.removeMember \
5 -d '{ "teamName": "team1", "userId": "kRwRRzfNybEDfQeij" }'
```

Example Result

```
1  {
2    "success": true
3 }
```

Remove Room

Removes a room from a team. Requires `remove-team-channel` permission.

URL	Requires Auth	HTTP Method
<code>/api/v1/teams.removeRoom</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>teamId</code>	<code>ByehQjC44FwMeiLbX</code>	Required	The team id. The team name.
<code>teamName</code>	<code>team1</code>		
<code>roomId</code>	<code>8Z7eRsibvD5AANfmK</code>	Required	The room (id) to remove from the team.

Example Call

```
1 curl -H 'X-Auth-Token: _2u_4MzRroRcnqc59GYUY_Kwgr9HgtZ9HCKn-2aIvMJ' \
2 -H 'X-User-Id: FL2fZL4ERhwA3gWiS' \
3 -H "Content-type: application/json" \
4 http://localhost:3000/api/v1/teams.removeRoom \
5 -d '{ "teamName": "team1", "roomId": "8Z7eRsibvD5AANfmK" }'
```

Example Result

```
1  {
2    "room": {
3      "_id": "8Z7eRsibvD5AANfMk",
4      "name": "roomRemovedFromTheTeam",
5      "fname": "roomRemovedFromTheTeam",
6      "t": "c",
7      "msgs": 7,
8      "usersCount": 1,
9      "u": {
10        "_id": "FL2fZL4ERhwA3gWiS",
11        "username": "some.username"
12      },
13      "customFields": {},
14      "description": "",
15      "broadcast": false,
16      "encrypted": false,
17      "ts": "2021-04-27T19:59:07.258Z",
18      "ro": false,
19      "_updatedAt": "2021-04-28T20:57:18.362Z"
20    },
21    "success": true
22  }
```

Update

Updates an existing team (name and type). Requires `edit-team` permission.

URL	Requires Auth	HTTP Method
/api/v1/teams.update	yes	POST

Payload

Argument	Example	Required	Description
teamId	ByehQjC44FwMeiLbX	Required	The team id.
data	{ "name": "newTeamName", "type": 1 }	Required	The new team name and type (0 - public, 1 - private).

Example Call

```
1 curl -H 'X-Auth-Token: _2u_4MzRroRcnqc59GYUY_Kwgr9HgtZ9HCKn-2aIvMJ' \  
2   -H 'X-User-Id: FL2fZL4ERhwA3gWiS' \  
3   -H 'Content-Type: application/json' \  
4   http://localhost:3000/api/v1/teams.update \  
5   -d '{ "teamId": "ByehQjC44FwMeiLbX", "data": { "name": "newTeamName",
```

Example Result

```
1 {
```

```
2      "success": true  
3  }
```

Update Member

Updates a team member's roles. Requires `edit-team-member` permission.

URL	Requires Auth	HTTP Method
/api/v1/teams.updateMember	yes	POST

Payload

Argument	Example	Required	Description
teamId teamName	ByehQjC44FwMeiLbX team1	Required	The team id. The team name.
member	{ "userId": "8dugqGhuRvCBLdZft", "roles": ["owner"] }	Required	The user (id and new roles) to update in the team.

Example Call

```
1 curl -H 'X-Auth-Token: _2u_4MzRroRcnqc59GYUY_Kwgr9HgtZ9HCKn-2aIvMJ' \
2 -H 'X-User-Id: FL2fZL4ERhwA3gWiS' \
3 -H "Content-type: application/json" \
4 'http://localhost:3000/api/v1/teams.updateMember' \
5 -d '{ "teamName": "team1", "member": { "userId": "LrTDRE27fWPPEvaAwC" }}
```

Example Result

```
1  {
2    "success": true
3 }
```

Update Room

Updates a room from a team. Requires `edit-team-channel` permission.

URL	Requires Auth	HTTP Method
<code>/api/v1/teams.updateRoom</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>roomId</code>	<code>Dgh2xwJ3NFKWvKSqY</code>	Required	The room id.
<code>isDefault</code>	<code>true</code>	Required	Sets the room as Auto-join (<code>true</code>) or not (<code>false</code>).

Example Call

```
1 curl -H 'X-Auth-Token: _2u_4MzRroRcnqc59GYUY_Kwgr9HgtZ9HCKn-2aIvMJ' \
2 -H 'X-User-Id: FL2fZL4ERhwA3gWiS' \
3 -H "Content-type: application/json" \
4 http://localhost:3000/api/v1/teams.updateRoom \
5 -d '{ "roomId": "8Z7eRsibvD5AANfmK", "isDefault": true }'
```

Example Result

```
1 {
```

```
2   "room": {
3     "_id": "8Z7eRsibvD5AANfK",
4     "name": "channelToUpdate",
5     "fname": "channelToUpdate",
6     "t": "c",
7     "msgs": 7,
8     "usersCount": 1,
9     "u": {
10       "_id": "FL2fZL4ERhwA3gWiS",
11       "username": "some.username"
12     },
13     "customFields": {},
14     "description": "",
15     "broadcast": false,
16     "encrypted": false,
17     "ts": "2021-04-27T19:59:07.258Z",
18     "ro": false,
19     "_updatedAt": "2021-04-28T20:57:18.362Z",
20     "teamId": "607e0d9b49d493189836bfac",
21     "teamDefault": true
22   },
23   "success": true
24 }
```

Users

Url	Short Description	Details Page
/api/v1/users.presence	Gets all connected users presence.	Link
/api/v1/users.create	Create a new user.	Link
/api/v1/users.createToken	Create a user authentication token.	Link
/api/v1/users.deactivateIdle	Deactivate idle users.	Link
/api/v1/users.delete	Deletes an existing user.	Link
/api/v1/users.deleteOwnAccount	Deletes your own user.	Link
/api/v1/users.forgotPassword	Send email to reset your password.	Link
/api/v1/users.generatePersonalAccessToken	Generate Personal Access Token.	Link
/api/v1/users.getAvatar	Gets the URL for a user's avatar.	Link
/api/v1/users.getPersonalAccessTokens	Gets the user's personal access tokens.	Link
/api/v1/users.getPreferences	Gets the user's preferences.	Link
/api/v1/users.getPresence	Gets the online presence of a user.	Link
/api/v1/users.getStatus	Gets the user's status.	Link

/api/v1/users.getUsernameSuggestion	Gets a suggestion a new username to user.	Link
/api/v1/users.info	Gets a user's information, limited to the caller's permissions.	Link
/api/v1/users.list	All of the users and their information, limited to permissions.	Link
/api/v1/users.regeneratePersonalAccessToken	Regenerate a user personal access token.	Link
/api/v1/users.register	Register a new user.	Link
/api/v1/users.removeOtherTokens	Remove all other user tokens	Link
/api/v1/users.removePersonalAccessToken	Remove a personal access token.	Link
/api/v1/users.requestDataDownload	Request users download data.	Link
/api/v1/users.resetAvatar	Reset a user's avatar	Link
/api/v1/users.setAvatar	Set a user's avatar	Link
/api/v1/users.setPreferences	Set a user's preferences	Link
/api/v1/users.setStatus	Set the user's status	Link
/api/v1/users.setActiveStatus	Set a user's active status.	Link
/api/v1/users.update	Update an existing user.	Link
/api/v1/users.updateOwnBasicInfo	Update basic information of own user.	Link

Create Token

IMPORTANT To be able to use this endpoint you must set the ENV VAR (CREATE_TOKENS_FOR_USERS=true).

Create a user authentication token. This is the same type of session token a user would get via login and will expire the same way. Requires `user-generate-access-token` permission.

Example if you use snaps:

```
1 echo "CREATE_TOKENS_FOR_USERS=true" > /var/snap/rocketchat-server/comm  
2 sudo systemctl restart snap.rocketchat-server.rocketchat-server.servic
```

Create a user authentication token. Requires `user-generate-access-token` permission.

URL	Requires Auth	HTTP Method
/api/v1/users.createToken	yes	POST

Payload

Argument	Example	Required	Description
<code>userId</code> or <code>username</code>	BsNr28znDkG8aeo7W	Required	The id or username of the user.

Example Call - Via userId

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/users.createToken \
5   -d '{ "userId": "BsNr28znDkG8aeo7W" }'
```

Example Result

```
1 {
2   "data": {
3     "userId": "BsNr28znDkG8aeo7W",
4     "authToken": "2jdk99wuSjXP0201XlAks9sjDjAhSJmskAKW301mSuj9Sk",
5   },
6   "success": true
7 }
```

Example Call - Via username

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/users.createToken \
5   -d '{ "username": "test" }'
```

Example Result

```
1 {
2   "data": {
3     "userId": "BsNr28znDkG8aeo7W",
4     "authToken": "2jdk99wuSjXP0201XlAks9sjDjAhSJmskAKW301mSuj9Sk",
```

```
5  },
6  "success": true
7 }
```

Change Log

Version	Description
2.1.0	Added ENV VAR to be able to use this endpoint (process.env.CREATE_TOKENS_FOR_USERS).
0.56.0	Added

Create

Create a new user. Requires `create-user` permission.

URL	Requires Auth	HTTP Method
<code>/api/v1/users.create</code>	yes	POST

Note

- To save `customFields` you must first define the `customFields` in admin panel (Accounts -> Registration -> Custom fields).

Payload

Argument	Example	Required	Description
<code>email</code>	<code>example@example.com</code>	Required	The email address for the user.
<code>name</code>	<code>Example User</code>	Required	The display name of the user.
<code>password</code>	<code>pass@w0rd</code>	Required	The password for the user.
<code>username</code>	<code>example</code>	Required	The username for the user.
<code>active</code>	<code>false</code>	Optional Default: <code>true</code>	Whether the user is active, which determines if they can login or not.

<code>roles</code>	<code>['bot']</code>	Optional Default: <code>['user']</code>	The roles the user has assigned to them on creation.
<code>joinDefaultChannels</code>	<code>false</code>	Optional Default: <code>true</code>	Whether the user should join the default channels when created.
<code>requirePasswordChange</code>	<code>true</code>	Optional Default: <code>false</code>	Should the user be required to change their password when they login?
<code>sendWelcomeEmail</code>	<code>true</code>	Optional Default: <code>false</code>	Should the user get a welcome email?
<code>verified</code>	<code>true</code>	Optional Default: <code>false</code>	Should the user's email address be verified when created?
<code>customFields</code>	<code>{ twitter: '@example' }</code>	Optional Default: <code>undefined</code>	Any custom fields the user should have on their account.

Example Call

```

1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/users.create \
5   -d '{"name": "name", "email": "email@user.tld", "password": "anypassy"

```

Example Result

```
1  {
2      "user": {
3          "_id": "BsNr28znDkG8aeo7W",
4          "createdAt": "2016-09-13T14:57:56.037Z",
5          "services": {
6              "password": {
7                  "bcrypt": "$2a$10$5I5nUzqNEs8jKhi7BFS55uFYRf5TE4ErSUH8HymMNAbp"
8              }
9          },
10         "username": "uniqueusername",
11         "emails": [
12             {
13                 "address": "email@user.tld",
14                 "verified": false
15             }
16         ],
17         "type": "user",
18         "status": "offline",
19         "active": true,
20         "roles": [
21             "user"
22         ],
23         "_updatedAt": "2016-09-13T14:57:56.175Z",
24         "name": "name",
25         "settings": {}
26     },
27     "success": true
28 }
```

Note

- The `customFields` will not be returned if it does not exist on the server.

Change Log

Version	Description

0.48.0 `role` property is now `roles` which is an array of strings for the roles to create the user with.

0.45.0 Users created via this now join the default channels.

0.40.0 Added

Delete Own Account

Deletes your own user. Requires `Allow Users to Delete Own Account` enabled.
Accessible from Administration -> Accounts.

URL	Requires Auth	HTTP Method
<code>/api/v1/users.deleteOwnAccount</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>password</code>	<code>BsNr28znDkG8aeo7W</code>	Required	The password of user (encrypted in SHA256).
<code>confirmRelinquish</code>	<code>true</code>	Optional Default: <code>false</code>	Deletes own account even if user is the last owner of a room.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/users.deleteOwnAccount \
5   -d '{ "password": "BsNr28znDkG8aeo7W" }'
```

Example Result

```
1  {
2    "success": true
3 }
```

Change Log

Version	Description
3.7.0	Added <code>confirmRelinquish</code> to the payload.
0.67.0	Added

Delete

Deactivate Idle users. Requires `edit-other-user-active-status` permission.

URL	Requires Auth	HTTP Method
<code>/api/v1/users.deactivateIdle</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>daysIdle</code>	2	Required	The days idle.
<code>role</code>	admin	Optional, default <code>user</code>	User role.

Example Call - Via userId

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/users.deactivateIdle \
5   -d '{ "daysIdle": 2, "role": "admin" }'
```

Example Result

```
1 {
2   "count": 1,
```

```
3      "success": true  
4  }
```

Change Log

Version	Description
3.1.0	Added

Delete

Deletes an existing user. Requires `delete-user` permission.

URL	Requires Auth	HTTP Method
<code>/api/v1/users.delete</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>userId</code> or <code>username</code>	<code>BsNr28znDkG8aeo7W</code>	Required	The id or username of the user.
<code>confirmRelinquish</code>	<code>true</code>	Optional Default: <code>false</code>	Deletes user even if it is the last owner of a room.

Example Call - Via userId

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/users.delete \
5   -d '{ "userId": "BsNr28znDkG8aeo7W" }'
```

Example Result

```
1  {
2    "success": true
3 }
```

Example Call - Via username

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/users.delete \
5   -d '{ "username": "test" }'
```

Example Result

```
1  {
2    "success": true
3 }
```

Change Log

Version	Description
3.7.0	Added <code>confirmRelinquish</code> to the payload.
0.35.0	Added

Forgot Password

Send email to reset your password.

URL	Requires Auth	HTTP Method
/api/v1/users.forgotPassword	no	POST

Note

- Please ensure that you have completed the configuration of the Email; otherwise, your users will not receive the mail normally. Accessible from Administration -> Email.

Payload

Argument	Example	Required	Description
email	email@rocket.cat	Required	The email to send password reset link

Example Call - As JSON

```
1 curl -H "Content-type:application/json" \
2       http://localhost:3000/api/v1/users.forgotPassword \
3       -d '{ "email": "email@rocket.cat" }'
```

Result

```
1  {
2    "status": "success"
3 }
```

Change Log

Version	Description
0.64.0	Added

Generate Personal Access Token

Generate Personal Access Token. Requires `create-personal-access-tokens` permission.

URL	Requires Auth	HTTP Method
<code>/api/v1/users.generatePersonalAccessToken</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>tokenName</code>	<code>mypersonaltoken</code>	Required	The name of the token.
<code>bypassTwoFactor</code>	<code>true</code> or <code>false</code>	Optional, default: <code>false</code>	If 2FA requirement should be ignored when using this token.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2 -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3 -H "Content-type:application/json" \
4 http://localhost:3000/api/v1/users.generatePersonalAccessToken \
5 -d '{ "tokenName": "mypersonaltoken", "bypassTwoFactor": true }'
```

Example Result

```
1  {
2    "token": "2jdk99wuSjXP0201XlAks9sjDjAhSJmskAKW301mSuj9Sk"
3    "success": true
4 }
```

Change Log

Version	Description
3.1.0	Added <code>bypassTwoFactor</code> param
0.69.0	Added

Get Avatar

Gets the URL for a user's avatar.

URL	Requires Auth	HTTP Method
/api/v1/users.getAvatar	no	GET

Payload

Argument	Example	Required	Description
userId or username	BsNr28znDkG8aeo7W	Required	The id or username of the user.

Example Call - Via userId

```
curl http://localhost:3000/api/v1/users.getAvatar?userId=BsNr28znDkG8aeo7W
```

Example Result

```
http://localhost:3000/avatar/bobsmith
```

Example Call - Via username

```
curl http://localhost:3000/api/v1/users.getAvatar?username=bobsmith
```

Example Result

```
http://localhost:3000/avatar/bobsmith
```

N.B. the response is a 307 temporary redirect

Change Log

Version	Description
0.50.0	Added

Get Personal Access Tokens

Gets the user's personal access tokens. Requires [create-personal-access-tokens](#) permission.

URL	Requires Auth	HTTP Method
/api/v1/users.getPersonalAccessTokens	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -H "Content-type:application/json" \
4     http://localhost:3000/api/v1/users.getPersonalAccessTokens
```

Example Result

```
1  {
2    "tokens": [
3      {
4        "name": "myToken",
5        "createdAt": "2018-08-01T17:17:48.068Z",
6        "lastTokenPart": "R8Agh3"
7      }
8    ],
9    "success": true
10 }
```

Change Log

Version	Description
0.69.0	Added

Get Preferences

Gets all preferences of the user.

URL	Requires Auth	HTTP Method
/api/v1/users.getPreferences	yes	GET

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/users.getPreferences
```

Example Result

```
1 {
2   "preferences": {
3     "newRoomNotification": "door",
4     "newMessageNotification": "chime",
5     "muteFocusedConversations": true,
6     "useEmojis": true,
7     "convertAsciiEmoji": true,
8     "saveMobileBandwidth": true,
9     "collapseMediaByDefault": false,
10    "autoImageLoad": true,
11    "emailNotificationMode": "all",
12    "roomsListExhibitionMode": "category",
13    "unreadAlert": true,
14    "notificationsSoundVolume": 100,
15    "desktopNotifications": "default",
16    "mobileNotifications": "default",
17    "enableAutoAway": true,
```

```
18     "highlights": [],
19     "desktopNotificationDuration": 0,
20     "viewMode": 0,
21     "hideUsernames": false,
22     "hideRoles": false,
23     "hideAvatars": false,
24     "hideFlexTab": false,
25     "sendOnEnter": "normal",
26     "roomCounterSidebar": false
27   },
28   "success": true
29 }
```

Get Presence

Gets a user's presence if the query string `userId` or `username` is provided, otherwise it gets the callee's.

URL	Requires Auth	HTTP Method
/api/v1/users.getPresence	yes	GET

Query Parameters

Argument	Example	Required	Description
<code>userId</code> or <code>username</code>	BsNr28znDkG8aeo7W	Optional	The id or username of the user. If not provided, the auth user is updated.

Other Users Example Call - Via userId

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/users.getPresence?userId=BsNr28znDkG8aeo
```

Example Result

```
1 {
2   "presence": "offline",
3   "success": true
4 }
```

Other Users Example Call - Via username

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/users.getPresence?username=test
```

Example Result

```
1 {
2   "presence": "offline",
3   "success": true
4 }
```

Self Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/users.getPresence
```

Example Result

```
1 {
```

```
2   "presence": "offline",
3   "connectionStatus": "offline",
4   "lastLogin": "2016-12-08T18:26:03.612Z",
5   "success": true
6 }
```

Change Log

Version	Description
0.49.0	Updated to support <code>userId</code> or <code>username</code>
0.48.0	Renamed to <code>users.getPresence</code>
0.35.0	Added as <code>user.getPresence</code>

Get Status

Gets a user's Status if the query string `userId` or `username` is provided, otherwise it gets the callee's.

URL	Requires Auth	HTTP Method
<code>/api/v1/users.getStatus</code>	yes	GET

Query Parameters

Argument	Example	Required	Description
<code>userId</code> or <code>username</code>	<code>BsNr28znDkG8aeo7W</code>	Optional	The id or username of the user. If not provided, the auth user is used.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/users.getStatus?userId=BsNr28znDkG8aeo7W
```

Example Result

```
1 {
2   "message": "My status update",
3   "connectionStatus": "online",
4   "status": "away",
```

```
5      "success": true
6  }
```

Self Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/users.getStatus
```

Example Result

```
1  {
2    "message": "Latest status",
3    "connectionStatus": "online",
4    "status": "online",
5    "success": true
6  }
```

Change Log

Version	Description
1.2.0	Added

Get Username Suggestion

Gets a suggestion a new username to user.

URL	Requires Auth	HTTP Method
/api/v1/users.getUsernameSuggestion	yes	GET

Other Users Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/users.getUsernameSuggestion
```

Example Result

```
1 {
2   "result": "rocket.cat",
3   "success": true
4 }
```

Change Log

Version	Description
0.65.0	Added

Info

Retrieves information about a user, the result is only limited to what the callee has access to view. It supports **Fields Query Parameter** with the `userRooms` field, that returns the rooms that the user is part of. The `rooms` field returns the `unread` property and this field is variable, based on the setting `Unread_Count` (Admin Panel => General => Unread_Count), this setting provides the ability to choose between options to count the unread messages.

URL	Requires Auth	HTTP Method
/api/v1/users.info	yes	GET

Query Parameters

Argument	Example	Required	Description
userId or username	BsNr28znDkG8aeo7W	Required	The id or username of the user.

Other Users Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \  
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \  
3   http://localhost:3000/api/v1/users.info?userId=BsNr28znDkG8aeo7W
```

Example Result Regular User Callee

```
1  {
2    "user": {
3      "_id": "nSYqWzZ4GsKTX4dyK",
4      "type": "user",
5      "status": "offline",
6      "active": true,
7      "name": "Example User",
8      "utcOffset": 0,
9      "username": "example",
10     "avatarETag": "thwftThP3Band942S",
11   },
12   "success": true
13 }
```

Example Result Admin Callee that requests user's rooms

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/users.info?userId=BsNr28znDkG8aeo7W&field=rooms
```

```
1  {
2    "user": {
3      "_id": "nSYqWzZ4GsKTX4dyK",
4      "createdAt": "2016-12-07T15:47:46.861Z",
5      "services": {
6        "password": {
7          "bcrypt": ...
8        },
9        "email": {
10          "verificationTokens": [
11            {
12              "token": "...",
13              "address": "example@example.com",
14              "when": "2016-12-07T15:47:46.930Z"
15            }
16          ]
17        },
18        "resume": {
19          "loginTokens": [
```

```
20          {
21              "when": "2016-12-07T15:47:47.334Z",
22              "hashedToken": "..."
23          }
24      ]
25  }
26 },
27 "emails": [
28     {
29         "address": "example@example.com",
30         "verified": true
31     }
32 ],
33 "type": "user",
34 "status": "offline",
35 "active": true,
36 "roles": [
37     "user"
38 ],
39 "name": "Example User",
40 "lastLogin": "2016-12-08T00:22:15.167Z",
41 "statusConnection": "offline",
42 "utcOffset": 0,
43 "username": "example",
44 "avatarETag": "thwftThP3Band942S",
45 "rooms": [
46     {
47         "_id": "PoffwDEebAwcmS5A8",
48         "rid": "GENERAL",
49         "name": "general",
50         "t": "c",
51         "unread": 1
52     },
53     {
54         "_id": "hr93s8GcJedZbkeah",
55         "rid": "oaqXoFLru3YLuau2J",
56         "name": "test",
57         "t": "c",
58         "unread": 1,
59         "roles": [
60             "owner"
61         ]
62     }
63 ],
64 },
65 "success": true
66 }
```

Change Log

Version	Description
3.4.0	Added <code>unread</code> property inside <code>rooms</code> object
0.70.0	Added <code>rooms</code> property to response if the user request it and has the <code>view-other-user-channels</code> permission
0.49.0	Updated to support <code>userId</code> or <code>username</code>
0.48.0	Renamed to <code>users.info</code>
0.35.0	Added as <code>user.info</code>

List

Gets all of the users in the system and their information, the result is only limited to what the callee has access to view. It supports the [Offset, Count, and Sort Query Parameters](#) along with [Query and Fields Query Parameter](#).

URL	Requires Auth	HTTP Method
/api/v1/users.list	yes	GET

Query Parameters

Argument	Example	Required	Description
fields	{ name: 1, emails: 0 }	Optional	Field include hash (value of 1 to include, 0 to exclude).
query	{ active: true, type: { \$in: ['user', 'bot'] } }	Optional	Query filter hash.

Other Users Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/users.list
```

Example Result Regular User Callee

```
1  {
2    "users": [
3      {
4        "_id": "nSYqWzZ4GsKTX4dyK",
5        "type": "user",
6        "status": "offline",
7        "active": true,
8        "name": "Example User",
9        "utcOffset": 0,
10       "username": "example"
11     },
12     {
13       ...
14     }
15   ],
16   "count": 10,
17   "offset": 0,
18   "total": 10,
19   "success": true
20 }
```

Example Result Admin Callee

```
1  {
2    "users": [
3      {
4        "_id": "nSYqWzZ4GsKTX4dyK",
5        "createdAt": "2016-12-07T15:47:46.861Z",
6        "services": {
7          "password": {
8            "bcrypt": ...
9          },
10         "email": {
11           "verificationTokens": [
12             {
13               "token": "...",
14               "address": "example@example.com",
15               "when": "2016-12-07T15:47:46.930Z"
16             }
17           ]
18         },
19         "resume": {
20           "loginTokens": [
```

```
21          {
22              "when": "2016-12-07T15:47:47.334Z",
23              "hashedToken": "..."
24          }
25      ]
26  }
27 },
28 "emails": [
29 {
30     "address": "example@example.com",
31     "verified": true
32 }
33 ],
34 "type": "user",
35 "status": "offline",
36 "active": true,
37 "roles": [
38     "user"
39 ],
40 "name": "Example User",
41 "lastLogin": "2016-12-08T00:22:15.167Z",
42 "statusConnection": "offline",
43 "utcOffset": 0,
44 "username": "example"
45 },
46 {
47     ...
48 }
49 ],
50 "count": 3,
51 "offset": 2,
52 "total": 10,
53 "success": true
54 }
```

Change Log

Version	Description
0.49.0	Count and offset query parameters supported.
0.35.0	Added

Regenerate Personal Access Token

Regenerate a user personal access token. Requires `create-personal-access-tokens` permission.

URL	Requires Auth	HTTP Method
<code>/api/v1/users.regeneratePersonalAccessToken</code>	yes	POST

Payload

Argument	Example	Required	Description
<code>tokenName</code>	<code>mypersonaltoken</code>	Required	The name of the token.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/users.regeneratePersonalAccessToken \
5   -d '{ "tokenName": "mypersonaltoken" }'
```

Example Result

```
1 {
2   "token": "sSfSoNW8Pb7S8BHEz3uJXYRM3__ML9GxPY1PxGzqAcn",
3   "success": true
```

```
4    }
```

Change Log

Version	Description
0.69.0	Added

Register

URL	Requires Auth	HTTP Method
/api/v1/users.register	no	POST

Payload

Argument	Example	Required	Description
username	rogersmith	Required	The username for the user.
email	roger@example.com	Required	The email for the user.
pass	passw0rd	Required	The password for the user.
name	Roger Smith	Required	The name of the user. Can be "" if <code>Require Name For Signup</code> is disabled in <code>Accounts > Registration</code>
secretURL	Jjwjg6gouWLXhMGKW	Optional	String appended to secret registration URL (if using).

Example Call

```
1 curl -H "Content-type:application/json" \
2   http://localhost:3000/api/v1/users.register \
3   -d '{ "username": "rogersmith", "email": "roger@example.com", "pass": "
```

Example Result

```
1  {
2    "user": {
3      "_id": "nSYqWzZ4GsKTX4dyK",
4      "type": "user",
5      "status": "offline",
6      "active": true,
7      "name": "Example User",
8      "utcOffset": 0,
9      "username": "example"
10     },
11    "success": true
12 }
```

Change Log

Version	Description
0.50.0	Added

Register

URL	Requires Auth	HTTP Method
/api/v1/users.removeOtherTokens	yes	POST

Example Call

```
1 curl -H "Content-type:application/json" \
2       http://localhost:3000/api/v1/users.removeOtherTokens \
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
3.1.0	Added

Remove Personal Access Token

Remove a personal access token. Requires `create-personal-access-tokens` permission.

URL	Requires Auth	HTTP Method
/api/v1/users.removePersonalAccessToken	yes	POST

Payload

Argument	Example	Required	Description
tokenName	mypersonaltoken	Required	The name of the token.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/users.removePersonalAccessToken \
5   -d '{ "tokenName": "mypersonaltoken" }'
```

Example Result

```
1 {
2   "success": true
3 }
```

Change Log

Version	Description
0.69.0	Added

Request Data Download

Request the user's data for download.

URL	Requires Auth	HTTP Method
/api/v1/users.requestDataDownload	yes	GET

Query Parameters

Argument	Example	Required	Description
fullExport	true or false	Optional, default false	If needs a full export.

Other Users Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/users.requestDataDownload
```

Example Result

```
1 {
2   "requested": false,
3   "exportOperation": {
4     "_id": "uDsuaw7vGd9B7j8qD",
5     "createdAt": "2019-06-07T23:02:13.359Z",
6     "userId": "hjwGZafNqExtFNmN7",
7     "roomList": [
```

```
8          {
9              "roomId": "GENERAL",
10             "roomName": "general",
11             "userId": null,
12             "exportedCount": 8,
13             "status": "completed",
14             "targetFile": "general.html",
15             "type": "c"
16         }
17     ],
18     "status": "uploading",
19     "exportPath": "/tmp/userData/hjwGZafNqExtFNmN7/partial",
20     "assetsPath": "/tmp/userData/hjwGZafNqExtFNmN7/partial/assets",
21     "fileList": [],
22     "generatedFile": "/tmp/zipFiles/hjwGZafNqExtFNmN7.zip",
23     "fullExport": false,
24     "_updatedAt": "2019-06-07T23:15:00.326Z"
25 },
26 "success": true
27 }
```

Change Log

Version	Description
1.2.0	Added as <code>users.requestDataDownload</code>

Reset Avatar

URL	Requires Auth	HTTP Method
/api/v1/users.resetAvatar	yes	POST

Payload

Argument	Example	Required	Description
userId or username	BsNr28znDkG8aeo7W	Required	The id or username of the user.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \  
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \  
3   -H "Content-type:application/json" \  
4   http://localhost:3000/api/v1/users.resetAvatar \  
5   -d '{ "userId": "BsNr28znDkG8aeo7W" }'
```

Example Result

```
1 {  
2   "success": true  
3 }
```

Change Log

Version	Description
0.55.0	Added

Set Active Status

URL	Requires Auth	HTTP Method
/api/v1/users.setActiveStatus	yes	POST

Payload

Argument	Example	Required	Description
activeStatus	true	Required	The value of the active status.
userId	5HmCfpob7jp2uibTC	Required	The user's id to be changed.
confirmRelinquish	true	Optional Default: false	Allows user to be deactivated even if it is the last owner of a room.

Note:

If `activeStatus=false` & `confirmRelinquish=true` and the user is the last remaining owner of a room, the oldest member of that room will be chosen as the new owner.

Example call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/users.setActiveStatus \
5   -d '{"activeStatus": false, "userId": "5HmCfpob7jp2uibTC"}'
```

Example Result

```
1  {
2      "user": {
3          "_id": "jJNyu4BQFqdgEcqnR",
4          "active": false
5      },
6      "success": true
7 }
```

Change Log

Version	Description
3.7.0	Added <code>confirmRelinquish</code> to the payload.
0.75.0	Added.

Set Avatar

URL	Requires Auth	HTTP Method
/api/v1/users.setAvatar	yes	POST

Uploading a File

Arguments

Argument	Example	Required	Description
image	Any image file	Required	The image file to use as the new avatar, as form data.
userId or username	BsNr28znDkG8aeo7W	Optional	The id or username of the user. If not provided, the auth user is updated.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -F "image=@my-own-avatar.png" \
4     http://localhost:3000/api/v1/users.setAvatar
```

Example Result

```
1 {
2     "success": true
3 }
```

Providing a URL

Arguments

Argument	Example	Required	Description
avatarUrl	http://domain.tld/to/my/own/avatar.jpg	Required	URL of the new avatar for the user.
userId or username	BsNr28znDkG8aeo7W	Optional	The id or username of the user. If not provided, the auth user is updated.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \  
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \  
3   -H "Content-type:application/json" \  
4   -d '{"avatarUrl": "http://domain.tld/to/my/own/avatar.jpg"}' \  
5   http://localhost:3000/api/v1/users.setAvatar
```

Example Result

```
1 {  
2   "success": true  
3 }
```

Change Log

Version	Description
0.56.0	Add support for <code>username</code> argument.
0.48.0	Set other users avatars if the callee has permission.
0.46.0	Added

Set Preferences

URL	Requires Auth	HTTP Method
/api/v1/users.setPreferences	yes	POST

Payload

Argument	Example	Required	Description
userId	BsNr28znDkG8aeo7W	Required	The user ID of the user whose preferences are being set.
data.newRoomNotification	test	Required	New room notification setting.
data.newMessageNotification	test	Required	New message notification setting.
data.useEmojis	true	Required	Use emojis setting.
data.convertAsciiEmoji	true	Required	Convert Ascii emoji setting.
data.saveMobileBandwidth	true	Required	Save mobile bandwidth setting.
data.collapseMediaByDefault	true	Required	Collapse media by default setting.
data.autoImageLoad	true	Required	Auto image load setting.
data.emailNotificationMode	test	Required	Email notification mode setting.
data.roomsListExhibitionMode	test	Required	Rooms list exhibition mode setting.
data.unreadAlert	true	Required	Unread alert setting.
data.notificationsSoundVolume	100	Required	Notifications sound volume setting.
data.desktopNotifications	test	Required	Desktop notifications setting.
data.mobileNotifications	test	Required	Mobile notifications setting.

<code>data.enableAutoAway</code>	<code>true</code>	Required	Ena
<code>data.highlights</code>	<code>[]</code>	Required	Hig
<code>data.desktopNotificationDuration</code>	<code>100</code>	Required	Dur
<code>data.desktopNotificationRequireInteraction</code>	<code>false</code>	Required	Det on it (r vers setti de).
<code>data.viewMode</code>	<code>0</code>	Required	Vie
<code>data.hideUsernames</code>	<code>false</code>	Required	Hid
<code>data.hideRoles</code>	<code>false</code>	Required	Hid
<code>data.hideAvatars</code>	<code>false</code>	Required	Hid
<code>data.sendOnEnter</code>	<code>test</code>	Required	Ser
<code>data.roomCounterSidebar</code>	<code>true</code>	Required	Dis sid
<code>data.language</code>	<code>pt-BR</code>	Required	Lan
<code>data.sidebarShowFavorites</code>	<code>true</code>	Optional	Show
<code>data.sidebarShowUnread</code>	<code>true</code>	Optional	Show
<code>data.sidebarSortby</code>	<code>test</code>	Optional	Show
<code>data.sidebarViewMode</code>	<code>test</code>	Optional	Show
<code>data.sidebarHideAvatar</code>	<code>true</code>	Optional	Show
<code>data.groupByType</code>	<code>true</code>	Optional	Grp

data.muteFocusedConversations	true	Optional	Mu

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     -H "Content-type:application/json" \
4     http://localhost:3000/api/v1/users.setPreferences \
5     -d '{ "userId": "BsNr28znDkG8aeo7W", "data": { } }'
```

Example Result

```
1 {
2     "success": true,
3     "user": {
4         "_id": "rocketchat.internal.admin.test",
5         "settings": {
6             "preferences": {
7                 "newRoomNotification": "door",
8                 "newMessageNotification": "chime",
9                 "muteFocusedConversations": true,
10                "useEmojis": true,
11                "convertAsciiEmoji": true,
12                "saveMobileBandwidth": true,
13                "collapseMediaByDefault": false,
14                "autoImageLoad": true,
15                "emailNotificationMode": "all",
16                "roomsListExhibitionMode": "category",
17                "unreadAlert": true,
18                "notificationsSoundVolume": 100,
19                "desktopNotifications": "default",
20                "mobileNotifications": "default",
21                "enableAutoAway": true,
22                "highlights": [],
23                "desktopNotificationDuration": 0,
24                "desktopNotificationRequireInteraction": false,
25                "viewMode": 0,
```

```
26             "hideUsernames": false,
27             "hideRoles": false,
28             "hideAvatars": false,
29             "hideFlexTab": false,
30             "sendOnEnter": "normal",
31             "roomCounterSidebar": false
32         }
33     }
34 }
35 }
```

Change Log

Version	Description
2.3.0	Added <code>desktopNotificationRequireInteraction</code> property.

Set Status

Sets a user Status when the status message and state is given.

URL	Requires Auth	HTTP Method
/api/v1/users.setStatus	yes	POST

Arguments

Argument	Example	Required	Description
message	My status update.	Required	The user's status message.
status	online	Optional	The user's status like online , away , busy , offline .

Example Call

```
1 curl -H "X-Auth-Token: 40tB-Cn5YQJ74QMLQXi4Zf4E_-e0P5CrklU2pW0tV9M" \
2   -H "X-User-Id: uunbZHiuEnib8Pawj" \
3   -d '{"message":"My status update", "status": "online"}' \
4   http://localhost:3000/api/v1/users.setStatus
```

Example Result

```
1 {
```

```
2      "success": true
3  }
```

Change Log

Version	Description
1.2.0	Added

Update own basic information

URL	Requires Auth	HTTP Method
/api/v1/users.updateOwnBasicInfo	yes	POST

Note

- To update the password or email for the user you must send `currentPassword` property encrypted in SHA256 together in payload.
- To save customFields you must first define the customFields in admin panel (Accounts -> Registration -> Custom fields)

Payload

Argument	Example
<code>data.email</code>	example@example.com
<code>data.name</code>	Example User
<code>data.username</code>	example

data.currentPassword

5994471abb01112afcc18159f6cc74b4f511b99806da59b3caf5a9c

data.newPassword

passw0rd

customFields

{ twitter: '@example' }

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/users.updateOwnBasicInfo \
5   -d '{"data": {"email": "rocket.cat@rocket.chat", "newPassword": "passw0rd", "currentPassword": "5994471abb01112afcc18159f6cc74b4f511b99806da59b3caf5a9c", "username": "rocket.cat", "name": "Example user"}' \
6   -H "Content-Type: application/json"
```

Example Result

```
1  {
2      "user": {
3          "_id": "47cRd58HnWwpqxhaZ",
4          "createdAt": "2018-03-09T13:17:11.951Z",
5          "services": {
6              "password": {
7                  "bcrypt": "$2a$10$zb/dp.AHZAhwPzgm0HcQA.5e0fu6pAYU5gchg307
8              },
9              "email": {
10                  "verificationTokens": [
11                      {
12                          "token": "rzDjTJBoN8qdc5LCVM3xqaRHEx0lR9mxkFutB6fv
13                          "address": "rocket.cat@rocket.chat",
14                          "when": "2018-03-09T13:17:11.968Z"
15                      },
16                      {
17                          "token": "bFdoXbtvVl4NYXNMQn-sb6t9TwuBYkkntakvkNq
18                          "address": "rocket.cat@rocket.chat",
19                          "when": "2018-03-09T13:18:33.414Z"
20                      },
21                      {
22                          "token": "reYkTD_0fKHYgU1p60N-UyJjvXGODl6EoZ-GLQlg
23                          "address": "rocket.cat@rocket.chat",
24                          "when": "2018-03-09T14:41:21.633Z"
25                      }
26                  ]
27              },
28              "resume": {
29                  "loginTokens": [
30                      {
31                          "when": "2018-03-09T13:17:12.110Z",
32                          "hashedToken": "T6sIP43ctqMedzVsxiF/dln27ua7l56/Ry
33                      },
34                      {
35                          "hashedToken": "0z7NG4SF000ZvR71nNqAhK0yv/90eGJ74W
36                      }
37                  ]
38              }
39          },
40          "emails": [
41              {
42                  "address": "rocket.cat@rocket.chat",
43                  "verified": false
44              }
45          ],
46          "type": "user",
47          "status": "online",
48          "active": true,
49          "name": "Example User",

```

```
50      "_updatedAt": "2018-03-09T20:21:08.992Z",
51      "roles": [
52          "admin",
53          "user",
54          "bot"
55      ],
56      "lastLogin": "2018-03-09T20:15:20.019Z",
57      "statusConnection": "online",
58      "utcOffset": -3,
59      "username": "rocket.cat",
60      "settings": {
61          "profile": {}
62      },
63      "customFields": {
64          "twitter": "userstwitter"
65      }
66  },
67  "success": true
68 }
```

Change Log

Version	Description
0.62.2	Added as <code>user.updateOwnBasicInfo</code>

Update

URL	Requires Auth	HTTP Method
/api/v1/users.update	yes	POST

Note

- To save `customFields` you must first define the `customFields` in admin panel (Accounts -> Registration -> Custom fields)
- To update the password for the user, "edit-other-user-password" permission must be set accordingly using an administrator account (Administration > Permissions).

Payload

Argument	Example	Required	Description
<code>userId</code>	BsNr28znDkG8aeo7W	Required	The id of the user to update.
<code>data.email</code>	example@example.com	Optional	The email address for the user.
<code>data.name</code>	Example User	Optional	The display name of the user.
<code>data.password</code>	pass@w0rd	Optional	The password for the user.
<code>data.username</code>	example	Optional	The username for the user.

data.active	false	Optional Default: true	Whether the user is active, which determines if they can login or not.
data.roles	['bot']	Optional Default: ['user']	The roles the user has assigned to them.
data.joinDefaultChannels	false	Optional Default: true	Whether the user should join the default channels.
data.requirePasswordChange	true	Optional Default: false	Should the user be required to change their password when they login?
data.sendWelcomeEmail	true	Optional Default: false	Should the user get a welcome email?
data.verified	true	Optional Default: false	Should the user's email address be verified?

`data.customFields`

```
{ twitter:  
  '@example' }
```

Optional
Default:
`undefined`

Any custom
fields the user
should have
on their
account.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \  
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \  
3   -H "Content-type:application/json" \  
4   http://localhost:3000/api/v1/users.update \  
5   -d '{"userId": "BsNr28znDkG8aeo7W", "data": { "name": "new name", "em
```

Example Result

```
1 {  
2   "user":{  
3     "_id": "BsNr28znDkG8aeo7W",  
4     "createdAt": "2016-09-13T14:57:56.037Z",  
5     "services": {  
6       "password": {  
7         "bcrypt": "$2a$10$5I5nUzqNEs8jKhi7BFS55uFYRf5TE4ErSUH8HymMNAbp"  
8       }  
9     },  
10    "username": "uniqueusername",  
11    "emails": [  
12      {  
13        "address": "newemail@user.tld",  
14        "verified": false  
15      }  
16    ],  
17    "type": "user",  
18    "status": "offline",  
19    "active": true,  
20  }  
21}
```

```
20     "roles": [
21         "user"
22     ],
23     "_updatedAt": "2016-09-13T14:57:56.175Z",
24     "name": "new name",
25     "customFields": {
26         "twitter": "userstwitter"
27     }
28 },
29     "success": true
30 }
```

Change Log

Version	Description
0.48.0	Renamed to <code>users.update</code>
0.35.0	Added as <code>user.update</code>

Users Presence

Gets all connected users presence.

URL	Requires Auth	HTTP Method
/api/v1/users.presence	yes	GET

Query Parameters

Argument	Example	Required	Description	Valid Format(s)
from	2019-05-22T12:11:45.392Z	Optional	The last date you got a status change	ISO 8601 datetime. Timezone, milliseconds and seconds are optional

All users presence example call

If you don't pass `from` parameter, you'll get a list of all users presence and the result will have a `full` field with value `true` indicating that.

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   http://localhost:3000/api/v1/users.presence
```

Example Result

```
1  {
2      "users":[
3          {
4              "_id":"rocket.cat",
5              "name":"Rocket.Cat",
6              "username":"rocket.cat",
7              "status":"online",
8              "utcOffset":0,
9              "avatarETag": "5BB9B5ny5DkKdrwkq"
10         },
11         {
12             "_id":"rocketchat.internal.admin.test",
13             "name":"RocketChat Internal Admin Test",
14             "username":"rocketchat.internal.admin.test",
15             "status":"online",
16             "utcOffset":-2,
17             "avatarETag": "iEbEm4bTT327NJjXt"
18         }
19     ],
20     "full": true,
21     "success":true
22 }
```

Users presence changed after period example call

If you pass `from` parameter, you'll get only a list users presence which have changed after the time of `from` parameter. The `full` result field will then be `false`, indicating it's a partial result.

If the value of `from` parameter is older than 10 minutes, the server will reply with a **full** result set.

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCufD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/users.presence?from=2019-05-22T12:11:45.
```

Example Result

```
1  {
2      "users":[
3          {
4              "_id":"rocket.cat",
5              "name":"Rocket.Cat",
6              "username":"rocket.cat",
7              "status":"online",
8              "utcOffset":0,
9              "avatarETag": "5BB9B5ny5DkKdrwkq"
10         },
11         {
12             "_id":"rocketchat.internal.admin.test",
13             "name":"RocketChat Internal Admin Test",
14             "username":"rocketchat.internal.admin.test",
15             "status":"online",
16             "utcOffset":-2,
17             "avatarETag": "iEbEm4bTT327NJjXt"
18         }
19     ],
20     "full": false,
21     "success":true
22 }
```

Change Log

Version	Description
1.1.0	Added

Video Conference

Url	Short Description	Details Page
/api/v1/video-conference/jitsi.update-timeout	Updates the timeout of Jitsi video conference in a channel.	Link

Update Jitsi Timeout

URL	Requires Auth	HTTP Method
/api/v1/video-conference/jitsi.update-timeout	yes	POST

Payload

Argument	Example	Required	Description
roomId	ehyoM5JSQi94RwqPE7	Required	The room Id of the room to update the jitsi timeout.

Example Call

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2   -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3   -H "Content-type:application/json" \
4   http://localhost:3000/api/v1/video-conference/jitsi.update-timeout \
5   -d '{ "roomId": "hyoM5JSQi94RwqPE7" }'
```

Example Result

```
1 {
2   "jitsiTimeout": "2019-01-16T13:17:11.951Z",
3   "success": true
4 }
```

Change Log

Version	Description
0.74.0	Added

Webdav

Url	Short Description	Details Page
/api/v1/webdav.getMyAccounts	Retrieves the user's webdav accounts.	Link

Get My Accounts

Retrieves the user's webdav accounts.

URL	Requires Auth	HTTP Method
/api/v1/webdav.getMyAccounts	yes	GET

Example call to get all the roles

```
1 curl -H "X-Auth-Token: 9HqLlyZ0ugoStsXCUfD_0YdwnNnunAJF8V47U3QHXSq" \
2     -H "X-User-Id: aobEdbYhXfu5hkeqG" \
3     http://localhost:3000/api/v1/webdav.getMyAccounts
```

Example Result

```
1 {
2   "accounts": [
3     {
4       "_id": "P3Gru7ocFCd4vpKEs",
5       "server_url": "http://localhost:8080/remote.php/webdav/",
6       "username": "admin",
7       "name": "Webdav account"
8     }
9   ],
10  "success": true
11 }
```

Change Log

Version	Description
2.4.0	Added

Access Tokens

With the addition of the ability to generate personal access tokens, you can use the REST API without having to sign in.

The administrator must grant to the user role the necessary permission

`create-personal-access-tokens` to be able to generate personal access tokens.

After that you can generate personal access tokens in your profile page:

`Profile -> My Account -> Security -> Personal Access Tokens`

The tokens that will be generated are irrecoverable, once generated, you must save it in a safe place, if the token is lost or forgotten, you can regenerate or delete the token.

With the token in hand you can add in the header `x-Auth-Token` along with your user id `X-User-Id` of the request that is made to the REST API.

Two Factor Authentication

Visit the [Two Factor Authentication](#) guide for general information.

Errors

When a call that requires two factor is made it will return an `errorType` `totp-require`. The `details` object will list the method that has been required (email on this example) so it's possible to inform the user to check his email for the code.

- **method:** The method selected by the server. Useful to inform the user where to look for the code.
- **codeGenerated:** Email only. Used to inform if the code was generated or if there are tokens available already.
- **codeCount:** (optional) Email only. The number of available codes already sent via email.
- **codeExpires:** (optional) Email only. A list of expiration dates of the tokens.
- **availableMethods:** The list of available methods for Two Factor. When calling an api it's possible to define the method to use.

```
1  {
2      "success": false,
3      "error": "TOTP Required [totp-required]",
4      "errorType": "totp-required",
5      "details": {
6          "method": "email",
7          "codeGenerated": false,
8          "codeCount": 1,
9          "codeExpires": [
10              "2020-01-02T13:06:42.408Z"
11          ],
12          "availableMethods": [
13              "email"
14          ]
15      }
16  }
```

Calling an endpoint with Two Factor

After receive the error it's necessary to pass the informed code to the API. For that we need to call the same endpoint passing the some new new headers:

Request (new headers)

- **x-2fa-code**: (string) The code informed by the user;
- **x-2fa-method**: (string) The desired method to check the Two Factor, usually the same from the error;

Result

- If the two factor was accepted the **result** and the **error** will came from the original endpoint;
- If the two factor was not accepted the **error totp-invalid** will be returned;

```
1 // Error example
2 {
3     "success": false,
4     "error": "TOTP Invalid [totp-invalid]",
5     "errorType": "totp-invalid",
6     "details": {
7         "method": "email"
8     }
9 }
```

Requesting a new email code

If the user didn't receive the code it's possible to request the server to send a new code via email by calling the endpoint `users.2fa.sendEmailCode` via POST passing the user's email or username as body. It's required to pass the email or username because this endpoint can be called when the user is not logged in.

Request

- **sendEmailCode:** (string) The user's username or email

Result

- If success: **array of emails** to where the code was sent;
- If error: **error-parameter-required** if the parameter `emailOrUsername` was not provided;
- If error: **error-invalid-user** if the user was not found with the provided `emailOrUsername` ;

Example

```
await APIClient.v1.post('users.2fa.sendEmailCode', undefined, {emailOrUserna
```

Handling password fallback

If an api request returns TOTP Required with a method *password*, then the API user's password is required to authenticate the request:

```
1 // Error example
2 {
3     "success":false,
4     "error":"TOTP Required [totp-required]",
5     "errorType":"totp-required",
6     "details":{
7         "method":"password",
8         "codeGenerated":false,
9         "availableMethods":[]
10    }
11 }
```

Request (new headers)

The request must be resubmitted to the same end-point with the two additional headers

- **X-2fa-code:** (string) The API user's password sha256 hashed;
- **X-2fa-method:** 'password';

```
1 curl -H "X-Auth-Token: $YOUR_AUTH_TOKEN" \
2     -H "X-User-Id: $YOUR_USER_ID" \
3     -H "Content-type: application/json" \
4     -H "X-2fa-code: $SHA_256_HASH_OF_API_USER_PASSWORD" \
5     -H "X-2fa-method: password" \
6     http://localhost:3000/api/v1/users.update \
7     -d '{"userId": "SOME_USER_ID", "data": { "requirePasswordChange": false }}
```

Enabling the Two Factor via Email

It's possible to enable the email check by calling the endpoint `users.2fa.enable-email` via POST. Note that the two factor via email will only work if the user has at least one verified email.

Result

- If success: `{success: true}` is returned;
- If error: `not-authorized` if the user is not logged in;

Example

```
await APIClient.v1.post('users.2fa.enable-email');
```

Disabling the Two Factor via Email

It's possible to disabled the email check by calling the endpoint `users.2fa.disable-email` via POST. Note this endpoint requires the two factor to be executed.

Result

- If success: `{success: true}` is returned;
- If error: A two factor verification error is returned;

Example

```
await APIClient.v1.post('users.2fa.disable-email');
```

Query and Fields

Starting in version `0.49`, some of the endpoints support the query parameters `query` and `fields`. These two fields are special in that they accept a json object. If an invalid json object is passed, the request will fail.

Query Parameter Information

- `query` - json object which accepts **MongoDB** query operators
 - `fields` - json object with properties that have either `1` or `0` to include them or exclude them
-

Query Example

To query the users which has a name that has `g` in it somewhere.

```
https://localhost:3000/api/v1/users.list?query={ "name": { "$regex": "g" } }
```

Note

The allowed structure is **EJSON**, which is similar to JSON, except in the `Date` and `binary` fields. For query objects that use `Date` fields, you should use the structure as the example below:

```
query={"_updatedAt": {"$gt": { "$date": 1542814057 } }} or  
query={"_updatedAt": {"$gt": {"$date": "2018-11-21T15:27:28.202Z" }}}
```

Fields Example

To only return the usernames for users, you would do something like this:

```
http://localhost:3000/api/v1/users.list?fields={"username": 1}
```

Top tip: Providing a `fields` value of `{"_id": false, "value": false}` will return all other fields.

Pagination

Starting in version `0.49`, some of the endpoints support the query parameters `offset`, `count` and `sort`. However, the defaults are different per server due to configuration settings.

Query Parameter Information

Parameter	Example	Description	Format
<code>offset</code>	<code>50</code>	Number of items to "skip" in the query, i.e. requests return <code>count</code> items, skipping the first <code>offset</code> items	Positive integer
<code>count</code>	<code>50</code>	How many items to return.	Integer
<code>sort</code>	<code>{"value": -1, "_id": 1}</code>	List of fields to order by, and in which direction	JSON object, with properties listed in desired order, with values of <code>1</code> for ascending, or <code>-1</code> for descending.

Settings

These are the settings for the REST API which relate to the offset and count. If you need to change them, go to `Administration -> General -> REST API`.

- `API_Default_Count` (*Default Count*) - Defines the default `count` for when the parameter is not provided
- `API_Upper_Count_Limit` (*Max Record Amount*) - Defines ceiling for the `count` and this value will be set if the `count` is higher

- `API_Allow_Infinite_Count` (*Allow Getting Everything*) - Determines whether `0` can be passed into the `count` to allow getting all of the records.
-

Example

To limit the results to 5 items and to get the third "*page*", you would provide `count=5` and `offset=10`.

```
http://localhost:3000/api/v1/channels.list?count=5&offset=10
```

To sort by name descending, and status ascending, you would provide

```
sort={"name": -1, "status": 1}
```

```
http://localhost:3000/api/v1/users.list?sort={"name": -1, "status": 1}
```

Rate Limiter

i Note: Some endpoints (like `/api/v1/users.updateOwnBasicInfo`) might trigger the rate limiter and cannot currently be customized/disabled in [Administration Panel](#)

The rate limiter is set by default on all api endpoints, with an amount set in the [Administration Panel => Rate Limiter => API Rate Limiter](#) for time interval (in milliseconds). To disable the rate limiter you can add the `api-bypass-rate-limit` permission for your user group role in the [Administration -> Permission](#).

To disable programmatically or change the rate limiter, such as the number of calls and the time interval, simply provide for the API function `.addRoute`, within the existing options object that already defines `authRequired: true/false`, a property `rateLimiterOptions`, providing `false` if you want disable the RateLimiter for this endpoint, or by providing a valid configuration object with the desired configuration, in the following format

```
{numRequestsAllowed: 10, intervalTimeInMS: 60000} .
```

The rate limiter is set by default in the development environment and can be disabled in the admin panel in the [Administration Panel => Rate Limiter => API Rate Limiter](#).

For requests that are made to an api endpoint with a rate limiter, three extra properties will be sent by the request headers.

1. `x-ratelimit-limit` : The number of calls allowed for a certain amount of time.
2. `x-ratelimit-remaining` : The number of requests remaining in the current rate limit window.
3. `x-ratelimit-reset` : The time at which the current rate limit window resets in **UTC epoch milliseconds**.

Livechat Widget API

Usage

Livechat Widget API code must be inserted after the Livechat Widget installation script and wrapped as a callback of `RocketChat();` function.

You can call multiple Livechat Widget APIs on the same page.

Methods

Set custom field

To set a custom field for a visitor, you can use the following code:

```
1 RocketChat(function() {
2     this.setCustomField('fieldName1', 'Any value you want to store');
3     this.setCustomField('fieldName2', 'A value set just once', false); //
4});
```

Set theme options

To change the online color of the Livechat widget, use the following code:

```
1 RocketChat(function() {
2     this.setTheme({
3         color: '#04436A', // widget title background color
4         fontColor: '#FFFFFF', // widget title font color
5         iconColor: '#1d74f5', // widget icon color
6         title: "Welcome to Rocket.Chat", // default widget title when the
7         offlineTitle: "Service is offline", // default widget title when t
8     });
9});
```

Assign chats to a specific department

To automatically assign a Livechat widget to a specific department (for example, to use a unique Livechat widget on more than one website), use the following code:

```
1 RocketChat(function() {  
2     this.setDepartment('FILL HERE DEPARTMENT NAME - case sensitive');  
3 });
```

Set visitor token

To set an external token for a visitor, you can use the following code:

```
1 RocketChat(function() {  
2     this.setGuestToken('FHwaLnp8fzjMupSAj');  
3 });
```

Set name field

To set the visitor name field, you can use the following code:

```
1 RocketChat(function() {  
2     this.setGuestName('visitor name');  
3 });
```

Set email field

To set the visitor email field, you can use the following code:

```
1 RocketChat(function() {  
2     this.setGuestEmail('sample@rocket.chat');  
3 });
```

Register visitor

To register the visitor without using the registration form, you can use the following code:

```
1 RocketChat(function() {  
2     this.registerGuest({  
3         token: 'FHwaLnp8fzjMupSAj', // The token field is not required. If  
4         name: 'visitor Name',  
5         email: 'sample@rocket.chat',  
6         department: 'my_department', // The department field is not required.  
7         customFields: [ // The customFields field is not required. If it is  
8             {key: 'my_custom_field_a', value: 'my_custom_field_a_value'},  
9             {key: 'my_custom_field_b', value: 'my_custom_field_b_value'}  
10            ]  
11        });  
12    });
```

Set Language for Widget

There are number of language options, which you can choose from to set language for your widget. To check supported languages refer [here](#). To set language of widget use the following code.

```
1 RocketChat(function() {  
2     this.setLanguage('af');  
3 });
```

Set a default Agent before starting a new conversation

The widget allows setting a specific agent before the conversation starts, to do this follow these steps:

```
1 RocketChat(function() {  
2     this.setAgent({  
3         _id: 'h24yNtyoCmvp96wgt',  
4         username: 'rocket.chat',  
5     });
```

```
6  });
```

Initialize the widget by configuring all available properties in just one call

The widget allows configuring all the settings in just one method, the following properties are acceptable:

```
1  RocketChat(function() {
2      this.initialize({
3          theme: {
4              color: '#04436A',
5              fontColor: '#FFFFFF',
6              iconColor: '#1d74f5',
7              title: "Welcome to Rocket.Chat",
8              offlineTitle: "Service is offline",
9          },
10         department: 'sales',
11         guestToken: 'FHwaLnp8fzjMupSAj',
12         language: 'en',
13     });
14 });
```

Change widget visibility

You can either hide or show widget in your website. To hide widget use the following code:

```
1  RocketChat(function() {
2      this.hideWidget();
3  });
```

To show widget use the following code.

```
1  RocketChat(function() {
2      this.showWidget();
3  });
```

Change widget window state

You can either open or close then widget in your website. To open widget(default state) use the following code:

```
1 RocketChat(function() {  
2     this.maximizeWidget();  
3 });
```

To close the widget use the following code.

```
1 RocketChat(function() {  
2     this.minimizeWidget();  
3 });
```

Events

onChatMaximized

Fired when the chat widget is maximized.

```
1 RocketChat(function() {  
2     this.onChatMaximized(function() {  
3         // do whatever you want  
4         console.log('chat widget maximized');  
5     });  
6 });
```

onChatMinimized

Fired when the chat widget is minimized.

```
1 RocketChat(function() {
```

```
2     this.onChatMinimized(function() {
3         // do whatever you want
4         console.log('chat widget minimized');
5     });
6 });
```

onChatStarted

Fired when the chat is started (the first message was sent).

```
1 RocketChat(function() {
2     this.onChatStarted(function() {
3         // do whatever you want
4         console.log('chat started');
5     });
6 });
```

onChatEnded

Fired when the chat is ended either by the agent or the visitor.

```
1 RocketChat(function() {
2     this.onChatEnded(function() {
3         // do whatever you want
4         console.log('chat ended');
5     });
6 });
```

onPrechatFormSubmit

Fired when the pre-chat form is submitted.

```
1 RocketChat(function() {
2     this.onPrechatFormSubmit(function(data) {
3         // data is an object containing the following fields: name, email
4     });
5 });
```

```
5          // do whatever you want
6          console.log('pre-chat form submitted');
7      });
8  });
```

onOfflineFormSubmit

Fired when the offline form is submitted.

```
1 RocketChat(function() {
2     this.onOfflineFormSubmit(function(data) {
3         // data is an object containing the following fields: name, email
4
5         // do whatever you want
6         console.log('offline form submitted');
7     });
8 });
```

onWidgetHidden

Fired when widget is hidden.

```
1 RocketChat(function() {
2     this.onWidgetHidden(function(data) {
3         // do whatever you want
4         console.log('chat widget hidden');
5     });
6 });
```

onAssignAgent

Fired when an agent is assigned to the chat.

```
1 RocketChat(function() {
2     this.onAssignAgent(function(data) {
3         // data is an object containing the following fields: name, userna
```

```
4          // do whatever you want
5          console.log('Agent assigned');
6      });
7  });
8});
```

onWidgetShown

Fired when widget is shown.

```
1 RocketChat(function() {
2     this.onWidgetShown(function(data) {
3         // do whatever you want
4         console.log('chat widget shown');
5     });
6});
```

onAgentStatusChange

Fired when the status of the current agent changes.

```
1 RocketChat(function() {
2     this.onAgentStatusChange(function(data) {
3         // data is an object containing the following fields: name, userna
4
5         // do whatever you want
6         console.log('The status of the agent has changed');
7     });
8});
```

onServiceOffline

Fired when a visitor tries to start a new conversation and the Livechat service is offline.

```
1 RocketChat(function() {
2     this.onServiceOffline(function(data) {
```

```
3           // do whatever you want
4           console.log('The Livechat service is offline');
5       });
6   });
```

Change Log

Version	Description
3.1.0	Added <code>setAgent</code> and <code>initialize</code> methods. Also, improved the <code>setTheme</code> method adding more options to customize the widget
3.0.0	Added <code>onServiceOffline</code> callback
2.2.0	Added <code>maximizeWidget</code> and <code>minimizeWidget</code> methods.
1.3.0	Added <code>onAssignAgent</code> and <code>onAgentStatusChange</code> methods.
1.1.0	Added <code>showWidget</code> and <code>hideWidget</code> methods along with <code>onWidgetHidden</code> and <code>onWidgetShown</code> events
1.0.0	Added <code>setLanguage</code> method
0.66.0	Added <code>setGuestToken</code> , <code>setGuestName</code> , <code>setGuestEmail</code> and <code>registerGuest</code> methods.
0.53.0	Added callback events and the ability to pass a flag to <code>setCustomField</code> so the value passed does not get wrote if there is already an existing value.
0.36.0	Added <code>setTheme</code> method
0.26.0	Added <code>setCustomField</code> method

Apps Development

Rocket.Chat App

- (i) You can also create your own App and publish it on our official [Marketplace](#) to share with the world. Users can download your application from the marketplace and use it.

Below are a few guides which will help you get started with Rocket.Chat App CLI to create and publish your own applications.

Rocket.Chat provides an Apps Engine CLI which helps you easily create, deploy, and publish your own extensions for it. This guide will help you get started with the CLI to start creating your own apps.

Webinar on Getting Started with Rocket.Chat Apps Development

We had a webinar some time ago on how to get started with Rocket.Chat Apps development. It is a bit outdated, but can get you through the basics. And updated guide can be found at [Creating an App](#).

[You can watch it here.](#)

Rocket.Chat App Engine CLI

Our official Apps Engine [Command Line Interface \(CLI\)](#) is available on GitHub and published on [NPM](#). It provides an easy interface for developing extensions. Follow the quick steps below to create your first application.

Installation

Before installing the CLI make sure that you have [Node](#) already installed on your machine. To verify Node installation, use the following command in your terminal.

```
1 node -v
2 # v10.15.3
3 # It should return you a valid version.
```

Once you have installed Node, run the following command in your terminal to install the CLI globally.

```
npm install -g @rocket.chat/apps-cli
```

It will take a while to install the CLI depending upon your internet connection. After installation, run the following command to verify the installation.

```
1 rc-apps -v
2 # @rocket.chat/apps-cli/1.4.0 darwin-x64 node-v10.15.3
```

Note that the response may vary depending upon your machine and environment, but it should look similar. Now, you are all set to create your first app.

Creating an App

The development tools provide a command to quickly scaffold a new Rocket.Chat App. Simply run `rc-apps create` and a small form will ask you to provide information about the app you are developing, like its name (in this example we will call it `liftoff`), app's description, and developer info. This information will be placed in a file called `app.json` in the project's root folder.

You can check the `app-schema.json` file for all the detailed information and fields allowed in the app description file, the basic structure is similar to this:

```
1  {
2      "id": "88ce358a-5277-4356-a601-f519574303a4",
3      "version": "0.0.1",
4      "requiredApiVersion": "^1.4.0",
5      "iconFile": "icon.png",
6      "author": {
7          "name": "rocket.chat",
8          "homepage": "rocket.chat",
9          "support": "rocket.chat"
10     },
11     "name": "liftoff",
12     "nameSlug": "liftoff",
13     "classFile": "LiftoffApp.ts",
14     "description": "Tells if it is time to lift off"
15 }
```

Start Developing

The basic creation of an App is based on extending the App class from the Rocket.Chat Apps definition library. Your class also has to implement the constructor and optionally the initialize function, for more details on those check the [App definition documentation](#).

In this example, we already have our main file called `LiftoffApp.ts` that was generated when we first created our project:

```
1 import {
2     IAppAccessors,
3     ILogger,
4 } from '@rocket.chat/apps-engine/definition/accessors';
5 import { App } from '@rocket.chat/apps-engine/definition/App';
6 import { IAppInfo } from '@rocket.chat/apps-engine/definition/metadata';
7
8 export class LiftoffApp extends App {
9     constructor(info: IAppInfo, logger: ILogger, accessors: IAppAccessors)
10        super(info, logger, accessors);
11    }
12 }
```

Now let's add some functionality to it

Adding a Slashcommand

A Slashcommand is a way to call the app installed in Rocket.Chat. Your app can have multiple slashcommands and subcommands. In our example, we will add the `liftoff` slashcommand and it will be called like this by the user inside the chat:

```
/liftoff
```

First, we create a new directory called `commands` in our project's root and create a file called `liftoff.ts`. You can copy the file's content:

```
1 import {ISlashCommand, SlashCommandContext} from '@rocket.chat/apps-engine'
2 import {IModify, IRead} from '@rocket.chat/apps-engine/definition/accessor'
3 import {App} from '@rocket.chat/apps-engine/definition/App';
4
5 export class LiftoffCommand implements ISlashCommand {
6     public command = 'liftoff';
7     public i18nDescription = 'Tells the user if it is time to liftoff';
8     public i18nParamsExample = '';
9     public providesPreview = false;
10
11    constructor(private readonly app: App) {}
12 }
```

```

13     public async executor(context: SlashCommandContext, read: IRead, modify: ISlashCommandModify) {
14         const message = 'Time to lift off!';
15
16         const messageStructure = await modify.getCreator().startMessage();
17         const sender = context.getSender(); // the user calling the slash command
18         const room = context.getRoom(); // the current room
19
20         messageStructure
21             .setSender(sender)
22             .setRoom(room)
23             .setText(message);
24
25         await modify.getCreator().finish(messageStructure);
26     }
27 }

```

You can learn more about organising complex slash commands in our [Sub-command pattern](#) recipe

Registering the slashcommand

After adding our slashcomamnd logic, we have to register the slashcommand in out app by extending its configuration:

```

1 import {
2     IAppAccessors,
3     ILogger,
4     IConfigurationExtend,
5 } from '@rocket.chat/apps-engine/definition/accessors';
6 import { App } from '@rocket.chat/apps-engine/definition/App';
7 import { IAppInfo } from '@rocket.chat/apps-engine/definition/metadata';
8
9 import { LiftoffCommand } from './commands/liftoff';
10
11 export class LiftoffApp extends App {
12     constructor(info: IAppInfo, logger: ILogger, accessors: IAppAccessors) {
13         super(info, logger, accessors);
14     }
15
16     protected async extendConfiguration(configuration: IConfigurationExtend) {
17         await configuration.slashCommands.provideSlashCommand(new LiftoffCommand());
18     }
19 }

```

We first had to import the `Liftoff` class and then register an instance of using the `provideSlashCommand` function. We pass the app's instance (`this`) so our slashcommand have access to all the functionalities of the app.

Updating the app

If you want to, for example, change the message sent to the room from `Time to lift off!` to `Lift off now!`, you have to simply save the modifications and run:

```
rc-apps deploy --url http://localhost:3000 --username <your_username> --pass
```

The app will be updated and by sending `/liftoff`, the message will reflect the change you have made in the app.

Testing the App

Now that you have your App ready, you can test it before submitting it.

To test your app, you need a Rocket.Chat server running locally on your machine and an admin user in it.

See [Installing Rocket.Chat for Developing](#) to run Rocket.Chat in develop mode. Enable Apps development mode by navigating to `Administration > General` then scroll down to Apps and click on the `True` radio button over the Enable development mode.

or run it in preview mode with docker using the command:

```
docker run -it --rm -p 3000:3000 -v `pwd`/rocketdb:/var/lib/mongodb rocketchat
```

Having the server running, simply run inside the app project's directory:

```
rc-apps deploy --url http://localhost:3000 --username user_username --password user_password
```

Where:

`http://localhost:3000` is your local server URL (if you are running in another port, change the `3000` to the appropriate port)

`user_username` is the username of your admin user.

`user_password` is the password of your admin user.

If you want to update the app deployed in your Rocket.Chat instance after making changes to it, you can run:

```
rc-apps deploy --url http://localhost:3000 --username user_username --password user_password
```

After version 1.9 of the App Engine CLI, the `--update` flag isn't strictly necessary for updating an existing App, you can just run the `deploy` command without it.

App Submission to the Marketplace

So, you just finished writing your first Rocket.Chat App, and you have successfully tested it locally which means you are ready to submit it to the Marketplace for others to use. You can now submit your App to the Marketplace.

- (i) When you submit an App, there is a three-step process that takes place. In the first step, you submit the App to the Marketplace (explained below). Secondly, you will receive an email that asks for you to approve or reject the App submission (more on this later as well). Finally, after you approve the entry, then our staff will review it and determine whether it is approved or not.

Packaging the app

Currently the Rocket.Chat servers and Marketplace allow submission of zip files, these files can be created by running `rc-apps package` which packages your app and creates the zip file under `dist` folder.

App Submission through CLI

To submit your App simply run:

```
rc-apps submit
```

The terminal window shows the following sequence of commands and output:

```
douglasgubert@Dougs-MacBook: ~/dev/rocket.chat/apps-tests/schedulertest
drwxr-xr-x 16 douglasgubert staff 512B Jul  7 11:02 room-create/
drwxr-xr-x 16 douglasgubert staff 512B Dec 14 21:25 schedulertest/
drwxr-xr-x 15 douglasgubert staff 480B Nov 26 17:58 settingupdatetest/
drwxr-xr-x 15 douglasgubert staff 480B Jul 20 19:04 slackcompatibleapp/
drwxr-xr-x 6 douglasgubert staff 192B May  8 2020 subscription-testing_0.0.12/
drwxr-xr-x 13 douglasgubert staff 480B Dec 10 17:25 testversionsync/
drwxr-xr-x 15 douglasgubert staff 416B Jun 22 20:23 three-sixty-dialog-integration_0.0.1/
drwxr-xr-x 15 douglasgubert staff 480B Aug 20 18:33 transfertest/
drwxr-xr-x 14 douglasgubert staff 448B Mar 29 2020 user-update-test/
~/dev/rocket.chat/apps-tests npm install -g @rocket.chat/apps-cli@latest ✓ at 13:41:29
npm WARN deprecated cryptiles@4.1.3: This module has moved and is now available at @hapi/cryptiles. Please update your dependencies as this version is no longer maintained as it may contain bugs and security issues.
npm WARN deprecated boom@7.3.0: This module has moved and is now available at @hapi/boom. Please update your dependencies as this version is no longer maintained as it may contain bugs and security issues.
npm WARN deprecated hoek@6.1.3: This module has moved and is now available at @hapi/hoek. Please update your dependencies as this version is no longer maintained as it may contain bugs and security issues.
'Users/douglasgubert/.nvm/versions/node/v12.19.0/bin/rc-apps' -> '/Users/douglasgubert/.nvm/versions/node/v12.19.0/lib/node_modules/@rocket.chat/apps-cli/bin/run'
+ @rocket.chat/apps-cli@1.8.0
added 81 packages from 14 contributors and updated 231 packages in 23.249s

New patch version of npm available! 6.14.8 → 6.14.10
Changelog: https://github.com/npm/cli/releases/tag/v6.14.10
Run npm install -g npm to update!
```

Below this, another section of the terminal shows the submission process for the app 'schedulertest':

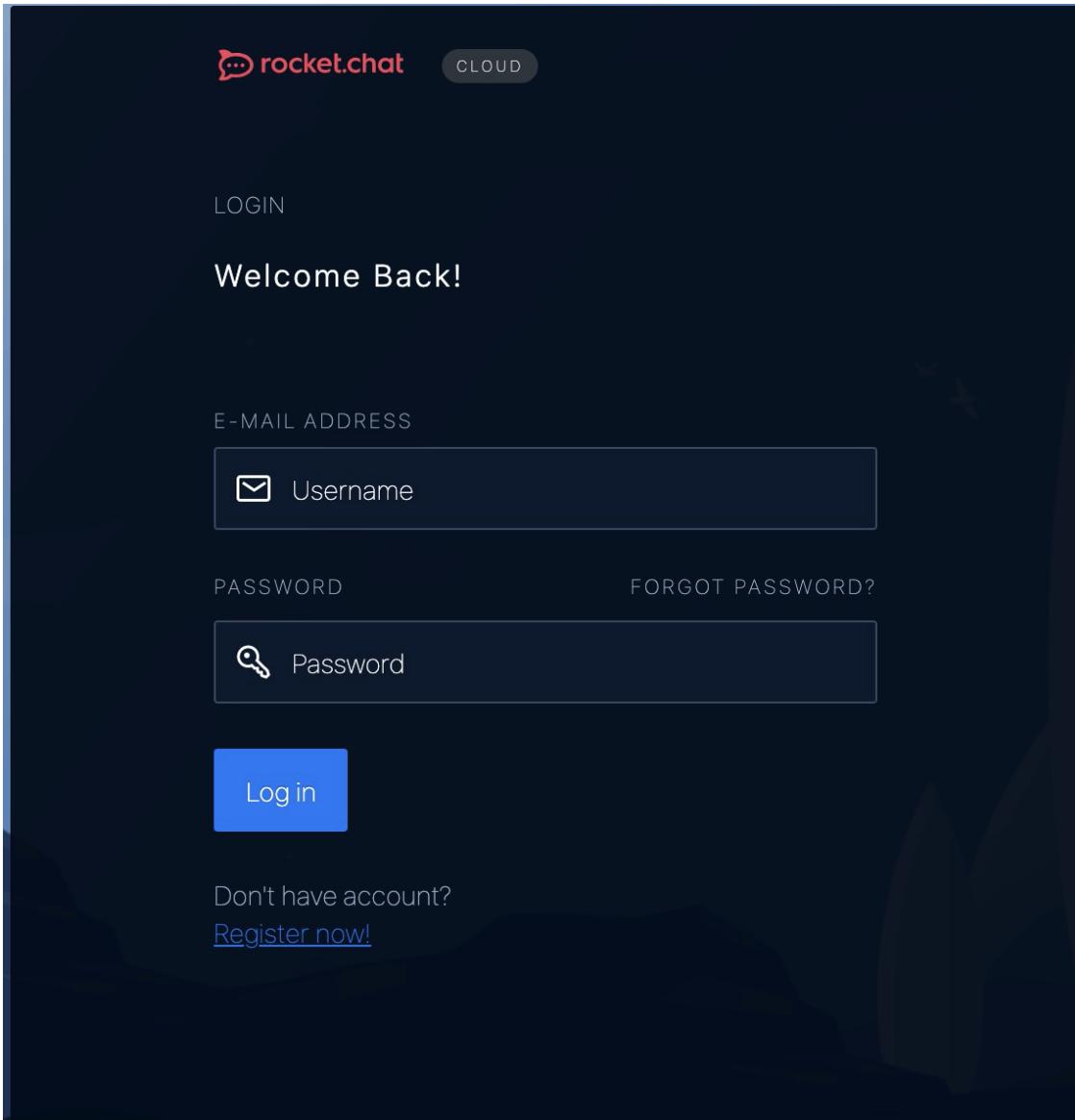
```
~/dev/rocket.chat/apps-tests schedulertest ✓ took 24s at 13:42:05
~/dev/rocket.chat/apps-tests/schedulertest rc-apps submit ✓ at 13:42:24
packaging your app... packaged!
fetching the available categories... fetched!
? Is this a new App? Yes
? Is this App free or will it require payment? Yes
? Please select the categories which apply to this App? Analytics
? Are you ready to submit? Yes
submitting your app... submitted!
~/d/r/apps-tests/schedulertest ✓ took 40s at 13:43:15
```

- If your app is free, it will go under review. If it's a paid app, you will be taken to Rocket.Chat Publish Portal.

Paid App Submission (through Publisher)

To submit an app through the publisher:

1. Go to <https://marketplace.rocket.chat/publisher/register> (provided in the CLI)
2. Log in using your account



1. Sign up your publisher account, as shown below:

Publisher Sign Up

Publisher Information

* Publisher Name

Please ensure this is your legal entity name (if you're an individual, please use your full name). It will be used for tax documents, reports, and will be shown publicly.

* City

Provide us the city where the legal entity name (and if you're an individual) is located.

* State

Provide us the state where the legal entity name (and if you're an individual) is located.

* Country

Your legal entity's country of establishment (or country of citizenship for individuals) must be provided.

* TOS Link

Please provide a link to your Terms of Service document. This link must be publicly accessible as users must be able to read it.

* Privacy Link

Please provide a link to your Privacy document. This link must be publicly accessible as users must be able to read it.

Your Personal Information

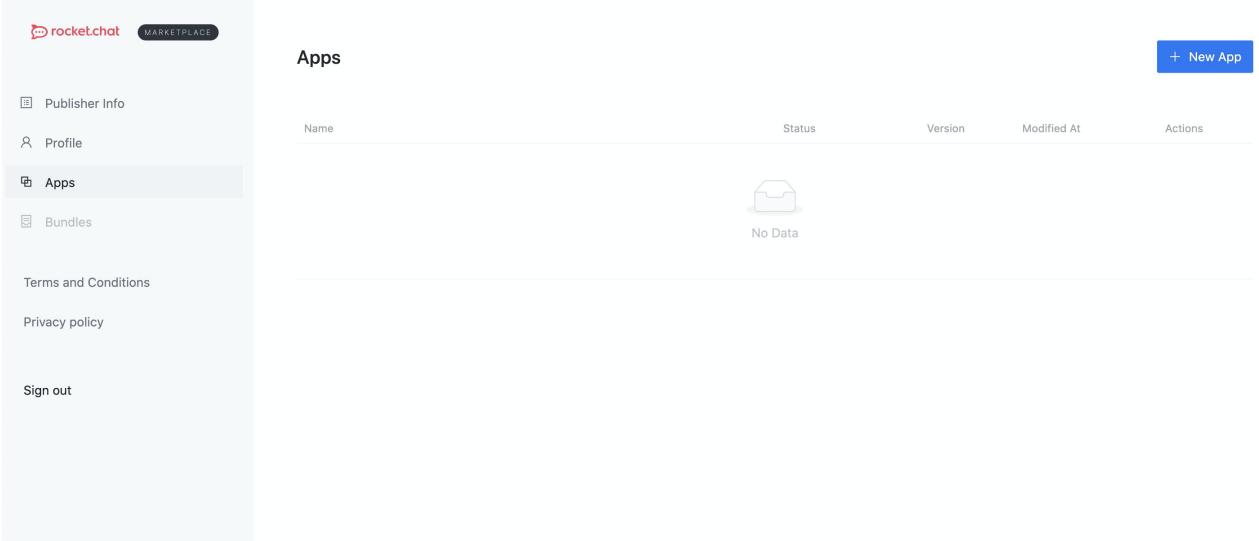
* Rocket.Chat Usern

Please provide us with your open.rocket.chat username. Why do we need this? In the event we need to contact you or we decide to do real time alerts.

I have read the [terms of service](#) and the [privacy agreement](#).

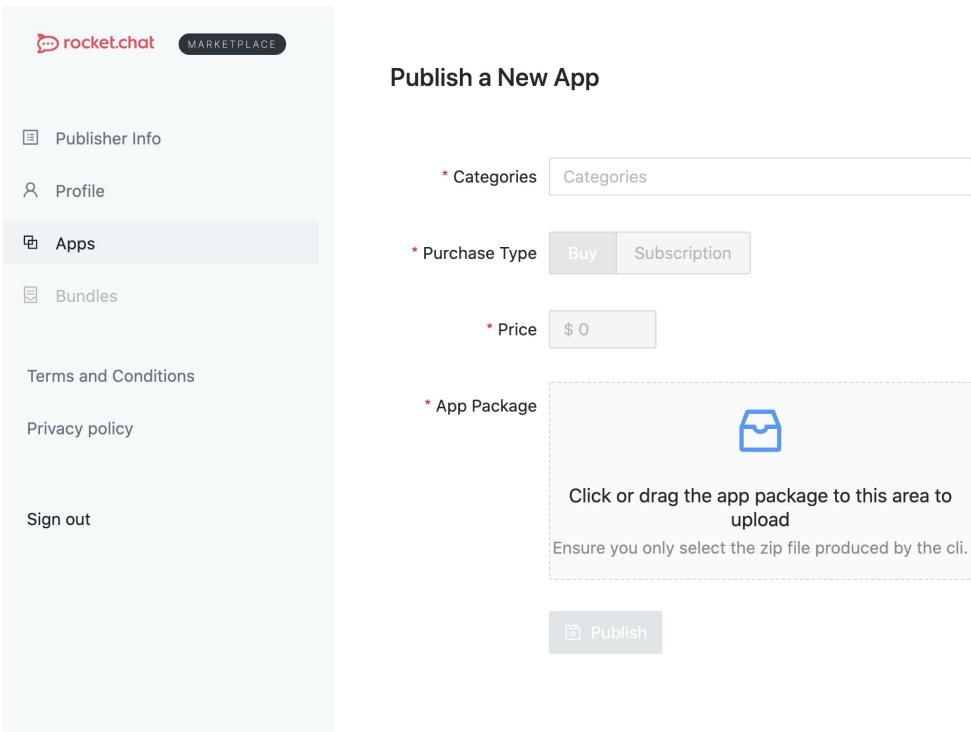
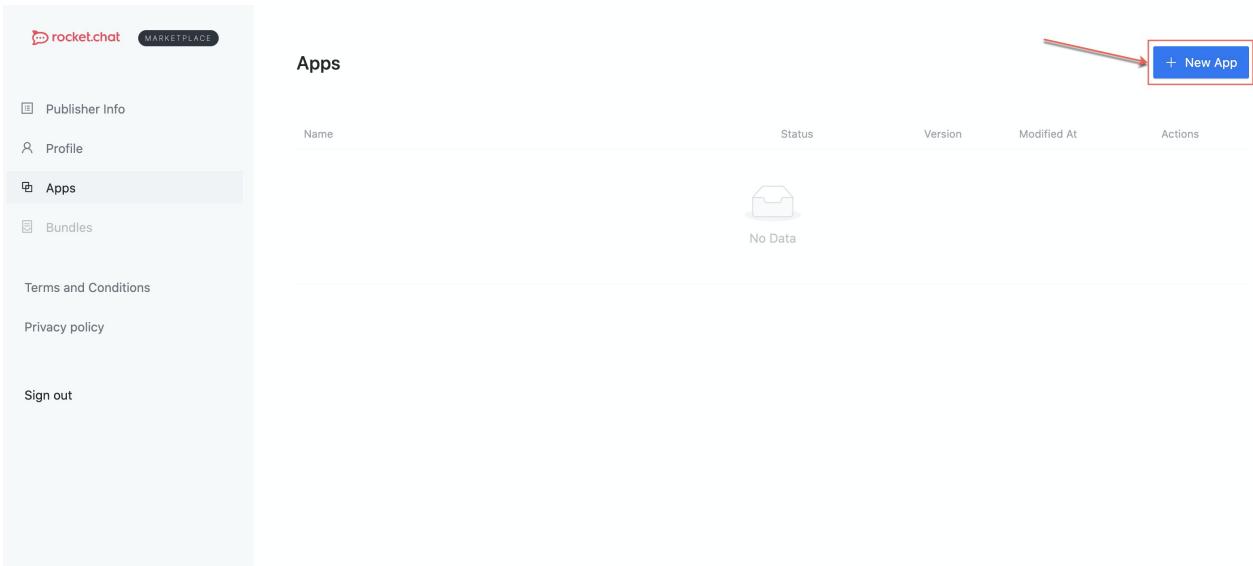
 **Sign Up**

Your publisher developer account is created, as shown below:



The screenshot shows the rocket.chat MARKETPLACE interface. On the left, there is a sidebar with links: Publisher Info, Profile, Apps (which is selected and highlighted in grey), Bundles, Terms and Conditions, Privacy policy, and Sign out. The main area is titled "Apps" and contains a table with columns: Name, Status, Version, Modified At, and Actions. There is one entry in the table: a folder icon under "Name", "No Data" under "Status", and a blue "+ New App" button under "Actions".

1. Click **New App** to add a new app:



To publish your app:

1. Choose the relevant category.
2. Select the purchase type. If your app is free, you can select the **Purchase Type: Buy** and enter the price \$0, as shown below:

Publish a New App

* Categories Communication × Developer Tools ×

* Purchase Type **Buy** Subscription

* Price \$ 0

* App Package



Click or drag the app package to this area to upload
Ensure you only select the zip file produced by the cli.

📎 poll_1.0.0.zip

Publish

If it is a fixed price app, select the **Purchase Type: Buy** and enter the price, as shown below:

Publish a New App

* Categories

* Purchase Type

* Price

* App Package 
Click or drag the app package to this area to upload
Ensure you only select the zip file produced by the cli.

 poll_1.0.0.zip

 Publish

It is a one-off payment and you are allowed to use it forever.



You are allowed to buy the app per workspace. If you have two workspaces and want to use the app on both, you need to buy the app separately.

If it is a subscription-based app, select the **Purchase Type: Subscription** and set up your plan **tier** wise. It allows you to pay a flat fee according to your unit. An example of this volume-based pricing is shown below:

The screenshot shows the Rocket.Chat Marketplace interface for publishing a new app. On the left, there's a sidebar with links like Publisher Info, Profile, Apps (which is selected), Bundles, Terms and Conditions, Privacy policy, and Sign out. The main area has tabs for Categories (Communication, Developer Tools), Purchase Type (Buy, Subscription), Pricing Plans (+ Add a plan), and App Package (with a file input field for 'poll_1.0.0.zip'). A large blue 'Publish' button is at the bottom. A modal window titled 'New Pricing Plan' is open on the right, showing fields for Strategy (Monthly, Yearly), Price (\$ 0.01), Trial Days (14), Is Per Seat (Yes checked), Tiers (Tier 1, Tier 2, Tier 3), and a 'Save' button.

1. Upload app package

2. Hit **Publish**.

After successful submission, you can see the status of your app as compiling, as shown below:

The screenshot shows the 'Poll' app details page. It includes sections for Publisher Info, Profile, Apps (selected), Bundles, Terms and Conditions, Privacy policy, and Sign out. The main content area shows the app icon, description ('A simple app to create polls on Rocket.Chat. Use the slash command: /poll [Question?]'), categories ('Communication, Developer Tools'), status ('Compiling'), purchase type ('Subscription'), pricing plans, and stats ('View Stats'). Below this is a 'Versions' section with a table showing one version (Name: Poll v1.0.0, Status: Compiling, Version: v1.0.0, Modified At: 14/01/2021). At the bottom are 'Jobs' and 'Actions' sections.

After we compile, you will be notified via email, as shown below:

Rocket.Chat App Marketplace Submission Status Update!



Rocket.Chat Marketplace <marketplace@rocket.chat>
to me ▾

4:37 AM (14 minutes ago)



Rocket.Chat App Marketplace

Hey there, we are writing to let you know that your Rocket.Chat Marketplace Submission for **Poll (v1.0.0)** has received a status update.

The new status is: **compiled**

Thanks,
Rocket.Chat Marketplace Team

Powered by [Rocket.Chat Rockets](#)

Then you get notified to review your app, as shown below:

Rocket.Chat App Marketplace Submission Received!



Rocket.Chat Marketplace <marketplace@rocket.chat>
to me ▾

4:36 AM (12 minutes ago)



Rocket.Chat App Marketplace

Hey there and thanks for submitting your Rocket.Chat App, **Poll v1.0.0!**

We have received your submission to the Marketplace. However, there is just one more step for you to do before the process is in our hands. We need you to approve or reject this submission.

Why do we require you to approve or reject it? We need your manual verification because we want to make sure that you are the one who submitted this submission.

[Approve](#) [Reject](#)

Please note, we will automatically reject the submission if we do not hear from you in seven (7) days.

Thanks,
Rocket.Chat Marketplace Team

You log in to your cloud account and approve it. After your approval, rocket.chat reviews it, and you will be notified via email that your app is now listed.

Fundamentals of Apps

→ **App Lifecycle**

/apps-development/fundamentals-of-apps/app-lifecycle

→ **Event Interfaces**

/apps-development/fundamentals-of-apps/event-interfaces

App Lifecycle

Introduction

Rocket.Chat Apps are fundamentally different from what you may be used to on other platforms, and that is because they actually run *inside a Rocket.Chat server*. This makes it possible for Rocket.Chat users to keep all their data inside their own structure, as the server does not need to send data to an externally hosted service that provides extra functionalities.

For an App to be added to a Rocket.Chat workspace, it needs to be installed in the server either by downloading the package via our [Marketplace](#) or by manually uploading the package to said workspace.

These are the steps the Apps-Engine takes to load a Rocket.Chat App:

- Unpackage the App (zip file) and compile the TypeScript code;
- Create a sandbox which will provide context for the App's code;
- Instantiate the App's main class;
- Initialize the App - `App.initialize`
- Enable the App - `App.onEnable`

Lifecycle Methods

The `App` class has some methods that can be extended in order to provide some control over various aspects of the App's lifecycle.

 Each one of the methods listed below is executed *for every instance of the App*. This means that they will be executed more than once if Rocket.Chat is running in HA mode (once per instance in the cluster)

initialize

```
1 public async initialize(configurationExtend: IConfigurationExtend, environ  
2         //  
3 }
```

This method allows the App to control its internal initialization process, overriding the default one.

extendConfiguration

```
1 protected async extendConfiguration(configuration: IConfigurationExtend, e  
2         //  
3 }
```

This method is executed as part of the default initialization process of the App. It enables the App to provide robust functionalities such as API Endpoints or Slash Commands using the [configuration accessor](#).

onEnable

```
1 public async onEnable(environment: IEnvironmentRead, configurationModify:  
2         //  
3 }
```

This method is executed during the enabling process of the App. If it returns `false`, the Apps-Engine halts the enabling process and unloads the App's resources configured during initialization.

onDisable

```
1 public async onDisable(configurationModify: IConfigurationModify): Promise
2     //
3 }
```

This method is executed during the disabling process of the App. It doesn't allow the App to prevent the disabling from happening.

onSettingUpdated

```
1 public async onSettingUpdated(setting: ISetting, configurationModify: IConfigurationModify): Promise
2     //
3 }
```

This method is executed *after* a setting provided by the App is updated by an admin in the *App Administration Page*. See **Settings (coming soon)** for more information on how to manage App settings.

Event Interfaces

Concepts

Interfaces

Interfaces are a typescript feature that can enforce a class meets a particular contract. The Rocket.Chat Apps-Engine provides various kinds of event interfaces that you can implement one or more of them for your app.

Implementing an event interface if you want to listen to this event and handle its payload data. Typically, each interface contains one method that will be called before/after a Rocket.Chat event.

- (i) Some old event interface, for instance, the IPreMessageSentPrevent interface contains two methods: one is `excutePreMessageSentPrevent`, which is the real event handler; and the other one is `checkPreMessageSentPrevent`, which is used to signal to Apps-Engine whether the real event handler should actually be executed. We are going to deprecate `check*` methods and won't add `check*` methods to event interfaces anymore. Please avoid using them as much as possible.

Handlers

Handlers are essential "listeners" for different events, except there are various ways to handle an event. When something happens there are `pre` and `post` handlers. The set of `pre` handlers happens before the event is finalized. The set of `post` handlers happens after the event is finalized. With that said, the rule of thumb is that if you are going to modify, extend, or change the data backing the event then that should be done in the `pre` handlers. If you are simply wanting to listen for when something happens and not modify anything, then the `post` is the way to go.

The order in which they happen is:

- PreEventPrevent
- PreEventExtend
- PreEventModify
- PostEvent

Here is an explanation of what each of them means:

- **Prevent**: This is ran to determine whether the event should be prevented or not.
- **Extend**: This is ran to allow extending the data without being destructive of the data (adding an attachment to a message for example).
- **Modify**: This is ran and allows for destructive changes to the data (change any and everything).
- **PostEvent**: Is mostly for simple listening and no changes can be made to the data.

For simplicity consideration, we are gradually moving to the new pattern. There will be only two types of event interfaces added later:

- IPreEvent
- IPostEvent

IPreEvent means handlers which are called before an event happens and **IPostEvent** means handlers that are called after an event happens.

Examples

Let's try to implement some event interfaces to see how the event handlers work:

IPreMessageSentPrevent

Code

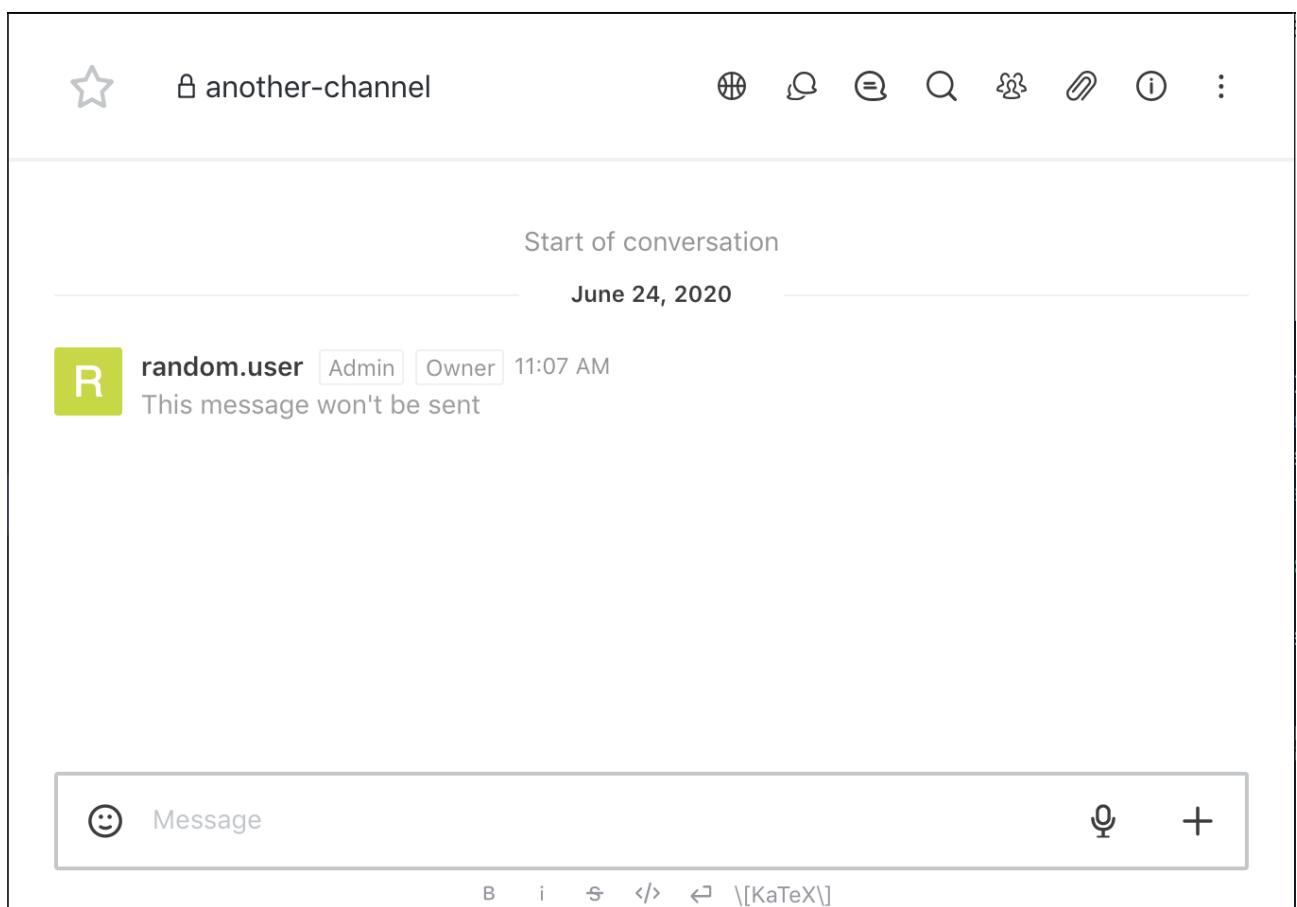
```

1 import { IAppAccessors, IHttp, ILogger, IPersistence, IRead } from '@rocket.chat/api';
2 import { App } from '@rocket.chat/apps-engine/definition/App';
3 import { IMessage, IPreMessageSentPrevent } from '@rocket.chat/apps-engine/definition/events';
4 import { IAppInfo } from '@rocket.chat/apps-engine/definition/metadata';

```

```
5
6  export class MessageEventsApp extends App implements IPreMessageSentPreven
7      constructor(info: IAppInfo, logger: ILogger, accessors: IAppAccessors)
8          super(info, logger, accessors);
9      }
10
11     public async executePreMessageSentPrevent(message: IMessage, read: IRe
12         return true;
13     }
14 }
```

Result



Message prevented by app

And you will see this in the server console

```
1 A Rocket.Chat App prevented the message sending. {
2     _id: 'sikKmLrZoQRaFzBwg',
3     rid: 'MCpMFsFqPfWhKGxXQ',
```

```
4     msg: "This message won't be sent",
5     ts: 2020-06-24T14:07:02.690Z,
6     u: {
7         _id: '2oKGD8QfRhyue62nX',
8         username: 'random.user',
9         name: undefined
10    }
11 }
```

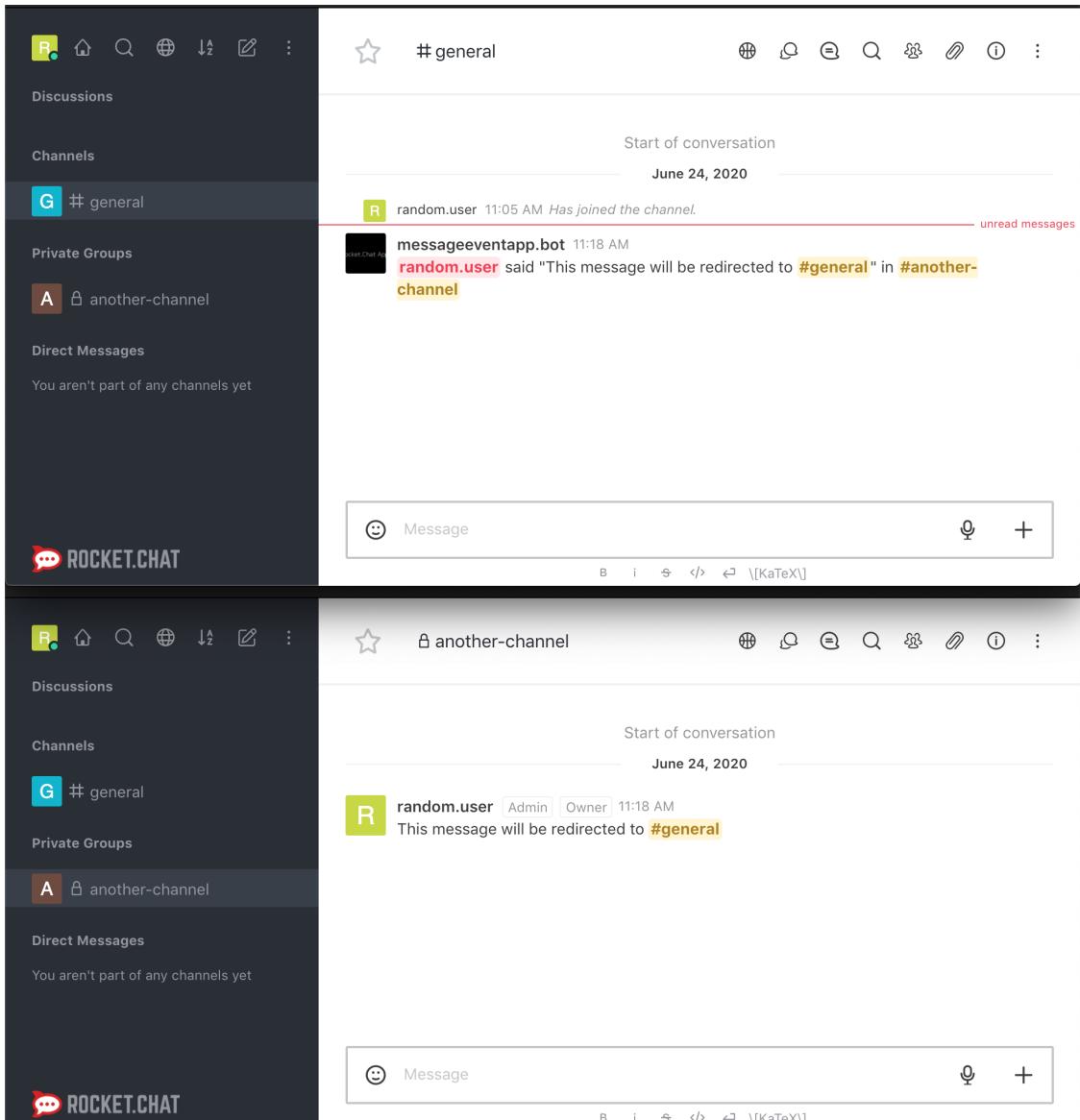
This simple app tries to prevent any message sending on this server. In the real world, you might (most likely) want to prevent messages according to the payload data provided instead of preventing all messages on the server.

IPostMessageSent

Code

```
1 import { IAppAccessors, IHttp, ILogger, IMessageBuilder, IModify, IPersist
2 import { App } from '@rocket.chat/apps-engine/definition/App';
3 import { IMessage, IPostMessageSent } from '@rocket.chat/apps-engine/definition/messages';
4 import { IAppInfo } from '@rocket.chat/apps-engine/definition/metadata';
5 import { IRoom } from '@rocket.chat/apps-engine/definition/rooms';
6
7 export class MessageEventsApp extends App implements IPostMessageSent {
8     constructor(info: IAppInfo, logger: ILogger, accessors: IAppAccessors) {
9         super(info, logger, accessors);
10    }
11
12    public async executePostMessageSent(message: IMessage, read: IRead, http: IHttp) {
13        if (message.room.slugifiedName === 'general') {
14            return;
15        }
16
17        const general = await read.getRoomReader().getByName('general');
18        const messageBuilder = modify.getCreator().startMessage({
19            text: `@${message.sender.username} said "${message.text}" in #${message.room.name}`
20        } as IMessage);
21
22        if (!general) {
23            return;
24        }
25        messageBuilder.setRoom(general);
26        await modify.getCreator().finish(messageBuilder);
27    }
}
```

Result



Message redirected to #general (top) from #another-channel (bottom)

This app is going to forward all messages sent to other channels, private groups or DMs to **#general**

References

Besides the two event interfaces we introduced above, you can check the full event interfaces list [here](#).

Scheduler API

How it works

The Scheduler API gives apps the possibility of creating tasks that will run in a defined schedule. It can me a one time event or a recurring task. It uses [agenda.js](#) as backend, so [schedule syntax](#) and internal processes are all according to its documentation.

Detailed documentation on the API can be found at the [Apps Engine's docs](#).

There's also an [example app](#) that can be used as guide.

Permissions

As per the [Permission System](#), the [Schedule API](#) needs the following permission in your app's manifest:

```
1  {
2    permissions: [
3      { "name": "scheduler" }
4    ]
5 }
```

How to use it

Registering job functions (processors)

To use the Scheduler API you'll need two things: the functions to be run as jobs (we call them `processors`) and the schedule in which they will be run on. This is configured in the app's `extendConfiguration` method. During the app's startup, the processors are registered to make them available for scheduling.

```
1 public async extendConfiguration(configuration: IConfigurationExtend) {  
2     configuration.scheduler.registerProcessors([  
3         {  
4             id: 'first',  
5             processor: async (jobData) => console.log(`[${ Date() }] this  
6         },  
7     ]);  
8 }
```

The `processor` is an `async` function and can receive arguments (`jobData`). The arguments are passed during scheduling.

Starting a job

To trigger the registered processor as a job, you must provide the `id` of the processor, the type of job it will be and optionally an object (`data`) that will be passed to the processor as argument (`jobData`). Available types are `RECURRING`, making the job run in a `interval`, and `ONETIME`, making the job run only once (`when`). Starting a job can be done when running a slashcommand, for example:

```
1 // slashcommand class  
2 public async executor(context: SlashCommandContext, read: IRead, modify: I  
3     // SCHEDULING A RECURRING TASK  
4     const task = {  
5         id: 'first',  
6         interval: '10 seconds',  
7         data: { test: true },  
8     };  
9     await modify.getScheduler().scheduleRecurring(task);  
10  
11    // SCHEDULING ONETIME TASK  
12    const task = {  
13        id: 'first',  
14        when: '8 seconds',  
15    };  
16    await modify.getScheduler().scheduleOnce(task);  
17 }
```

There's also the possibility of triggering a job as soon as it gets registered, without the need for manual or automated triggering.

When you register your processor in the `extendConfiguration` method, you can pass a prop called `startupSetting` in the processor's object:

```
1 import { StartupType } from '@rocket.chat/apps-engine/definition/scheduler
2 // ...
3 configuration.scheduler.registerProcessors([
4   {
5     id: 'first',
6     processor: async (jobData) => console.log(`[${ Date() }] this
7     startupSetting: {
8       type: StartupType.ONETIME,
9       when: '20 seconds',
10      data: { test: true },
11    }
12  },
13  {
14    id: 'second',
15    processor: async (jobData) => console.log(`[${ Date() }] this
16    startupSetting: {
17      type: StartupType.RECURRING,
18      interval: '20 seconds',
19    }
20  },
21]);
```

This will indicate that you want that particular processor to be scheduled as soon as it gets registered. You can define the "immediate scheduling" as a recurring job (`StartupType.RECURRING`) or a onetime job (`StartupType.ONETIME`). You can also pass data using the `data` object. It will work just like when you schedule a task using the `modify` accessor.

Cancelling a job

To stop a job, all you have to do is pass the id of the job you want to stop

```
1 const jobId = 'first';
```

```
2     await modify.getScheduler().cancelJob(jobId);
```

It will stop the running job (if any).

Cancelling all jobs from the app

To stop all the current running jobs from the app:

```
await modify.getScheduler().cancelAllJobs();
```

Recipes

→ **Making HTTP Requests**

/apps-development/recipes/making-http-
requests

→ **Storing User Input**

/apps-development/recipes/storing-user-
input

→ **Registering API endpoints**

/apps-development/recipes/registering-
api-endpoints

Sub-command pattern

In this recipe we will create a slashcommand that has two sub-commands. We will use the [App.RocketChat.Tester](#) as the base project for it. All the code for this recipe is in the branch [recipes/sub-command-pattern](#).

You can see the complete documentation on Slashcommands in Rocket.Chat Apps at [ISlashCommand](#)'s docs page.

Moving parts

You'll need to have handy:

- A Rocket.Chat server to deploy the app to
 - Our [Tester App](#) or a newly created app (your choice)
-

Create the slashcommand

Our "main command" (the slashcommand) will be called `phone`, and we will give it two sub commands, `text` and `call`, so when we use them, they will be as follows:

- `/phone text`
- `/phone call`

First, create a `slashcommands` directory at the root of the project and add a file named `PhoneCommand.ts`. Then add the following code:

```
1 import {
2     IHttp,
3     IModify,
4     IRead,
5 } from '@rocket.chat/apps-engine/definition/accessors';
```

```

6 import {
7     ISlashCommand,
8     SlashCommandContext,
9 } from '@rocket.chat/apps-engine/definition/slashcommands';
10
11 export class PhoneCommand implements ISlashCommand {
12     public command = 'phone'; // [1]
13     public i18nParamsExample = '';
14     public i18nDescription = '';
15     public providesPreview = false;
16
17     public async executor(context: SlashCommandContext, read: IRead, modify: IModify) {
18         const [subcommand] = context.getArguments(); // [2]
19
20         if (!subcommand) { // [3]
21             throw new Error('Error!');
22         }
23
24         switch (subcommand) { // [4]
25             case 'text': // [5]
26                 console.log('Texting!');
27                 break;
28
29             case 'call': // [6]
30                 console.log('Calling!');
31                 break;
32
33             default: // [7]
34                 throw new Error('Error!');
35         }
36     }
37 }

```

What this code tells us:

- [1] I want my command to be called `phone`.
- When it gets executed, [2] I want to get the argument the user passed after the command and to use it as the subcommand.
- [3] The argument is mandatory
- [4] Try to match the argument in the list of allowed subcommands
- [5] If `text` is match, log "Texting!"
- [6] If `call` is match, log "Calling!"
- [7] If anything else will throw an error

Learn more about Slashcommands in Rocket.Chat Apps at [ISlashCommand's docs page](#).

Register the slashcommand

After getting the slashcommand's code done, we have to register it at the app's main class, at the project's root.

```
1 import { IAppAccessors, IConfigurationExtend, ILogger } from '@rocket.chat/apps-engine/definition';
2 import { App } from '@rocket.chat/apps-engine/definition/App';
3 import { IAppInfo } from '@rocket.chat/apps-engine/definition/metadata';
4 import { PhoneCommand } from './slashcommands/PhoneCommand'; // [1]
5
6 export class RocketChatTester extends App {
7     constructor(info: IAppInfo, logger: ILogger, accessors: IAppAccessors) {
8         super(info, logger, accessors);
9     }
10
11     public async extendConfiguration(configuration: IConfigurationExtend) {
12         configuration.slashCommands.provideSlashCommand(new PhoneCommand());
13     }
14 }
```

Here we [1] import our new slashcommand class and then [2] register it in the app's configuration. Now it is available for us.

Deploy to the server

To deploy the app, run:

```
rc-apps deploy --url <server_url>><server_port> -u <user> -p <pwd>
```

Finally: calling the slashcommand

After deploying the app, you can type in any channel `/phone text` and the app will print "*Texting!*" to the console, or you can type `/phone call` and the app will print "*Calling!*" to the console.

If you don't see the result, enable info logs in your instance at *Administration > Logs > Log Level* by selecting "1 - Errors and Information".

Bonus: print the texts in the chat

Now, instead of console logging data to the instance's log, let's display the output in the chat.

In the `PhoneCommand.ts`, add the following private method:

```
1  private async sendMessage(context: SlashCommandContext, modify: IModify)
2    const messageStructure = modify.getCreator().startMessage();
3    const sender = context.getSender(); // [1]
4    const room = context.getRoom(); // [2]
5
6    messageStructure
7      .setSender(sender)
8      .setRoom(room)
9      .setText(message); // [3]
10
11    await modify.getCreator().finish(messageStructure); // [4]
12 }
```

This function [1] gets the user who called the command (in this case, yours), [2] selects the room in which the command was run, sets the received string as the message and then [4] sends the message in the room.

To know more about messaging, you can go to the [IMessageBuilder docs](#).

After that, change the `console.log`s from the switch block of the `executor` method and call the `sendMessage` method instead:

```
1     case 'text':
2         await this.sendMessage(context, modify, 'Texting!');
3         break;
4
5     case 'call':
6         await this.sendMessage(context, modify, 'Calling!');
7         break;
```

Important to note: `context` and `modify` are the arguments passed to the `executor` method and they will be passed over to the `sendMessage` method.

Now, instead of just printing text to the console, we print it in the channel we called the command from.

Save the file and redeploy the app by running:

```
rc-apps deploy --url <server_url><server_port> -u <user> -p <pwd> --update
```

Now, when you run the slashcommand `/phone text` or `/phone call`, you'll receive the response in the chat, as your user.

Extending Messages

In this recipe, we will create an app that is called through a slash command, sends a message and extends it in order to attach one image to it and add a custom field.

Attachments

Rocket.Chat supports numerous types of attachments (and applicable **customisations** to these attachments) which can be added to messages. For instance, you are able to attach images, documents, videos or audio files to messages, and creating your own **attachment class** is the first step in order to do so.

In this recipe, we will need to create an `ImageAttachment` class in the project's root, which can be implemented as follows:

```
1 import { IMessageAttachment } from '@rocket.chat/apps-engine/definition/me
2
3 export class ImageAttachment implements IMessageAttachment{
4     imageUrl?: string; // [1]
5
6     constructor(imgUrl: string){
7         this.imageUrl = imgUrl;
8     }
9
10 }
```

Here we [1] use a class' attribute with the **same identifier and type as in the `IMessageAttachment` interface**, which is essential to make your linked media visible to the user (since only the variables in your attachment class which are defined in the `IMessageAttachment` interface will be used to retrieve the attachment's media).

It is also possible to create your own classes for video or audio attachments, or even keep them all together in one single attachment class that may cover all of these possibilities. It's

all up to you! Just be sure to use the same attributes described in the `IMessageAttachment.d.ts` file.

- (i) The **audio** and **video** supported in Rocket.Chat message's attachments are the same as in HTML audio and video tags.

Custom Fields

Custom fields are sets of structures linked to messages where each field is arranged such as a dictionary entry. That is, each field must present a **key** and a corresponding **value**. These fields are stored in the server's database within its corresponding messages so that they can be retrieved later.

Create new auxiliary methods

It is helpful to define some auxiliary methods inside our slash command class so as to keep our code clear and concise.

Method: `sendMessage`

Firstly, we will need to modify the previously given `sendMessage` method so that the message's ID is returned after the message has been sent. Thus, the method must now return a `Promise<string>` instead of a `Promise<void>`.

The resulting `sendMessage` method is given below:

```
1 private async sendMessage(context: SlashCommandContext, modify: IModify, m
2     const messageStructure = modify.getCreator().startMessage();
3     const sender = context.getSender();
4     const room = context.getRoom();
```

```
5
6     messageStructure
7     .setSender(sender)
8     .setRoom(room)
9     .setText(message);
10
11    return (await modify.getCreator().finish(messageStructure)); // [1]
12 }
```

The only change that must be made (in comparison to the `sendMessage` described in previous pages) is [1] to return the result of the `finish` method, which is the ID of the message that has just been sent.

Method: `getMessageExtender`

After applying the changes in the `sendMessage` method and having the sent message's ID handy, we are now able to obtain the `messageExtender` object, which allows us to add custom fields and attachments.

The following asynchronous method can be used in order to obtain the `messageExtender` object:

```
1 private async getMessageExtender(context: SlashCommandContext, modify: IMo
2     const sender = context.getSender();
3     return modify.getExtender().extendMessage(messageId, sender); // [1]
4 }
```

Here we [1] use the message's ID returned by the `sendMessage` method in order to obtain the `messageExtender` object using the `modifyExtender` object.

Create the slash command

After developing the methods in the previous steps, we just now need to trigger these methods using a slash command, which can be done with the following code:

```

1 import { IHttp, IModify, IPersistence, IRead, IMessageExtender } from '@rocket.chat/api';
2 import { ISlashCommand, SlashCommandContext } from '@rocket.chat/apps-engine';
3 import { ImageAttachment } from '../ImageAttachment';
4
5 export class ExtendMessageCommand implements ISlashCommand{
6     public command = 'extend-message';
7     public i18nParamsExample = '';
8     public i18nDescription = '';
9     public providesPreview = false;
10
11    public async executor(context: SlashCommandContext, read: IRead, modify: IModify) {
12        const messageId = await this.sendMessage(context, modify, 'Sending');
13        const messageExtender = await this.getMessageExtender(context, modify);
14        const value = 1;
15        const img = new ImageAttachment('https://open.rocket.chat/images/logo.png');
16
17        messageExtender.addCustomField('key', value); // [4]
18        messageExtender.addAttachment(img); // [5]
19
20        await modify.getExtender().finish(messageExtender); // [6]
21    }
22
23    private async sendMessage(context: SlashCommandContext, modify: IModify) {
24        const messageStructure = modify.getCreator().startMessage();
25        const sender = context.getSender();
26        const room = context.getRoom();
27
28        messageStructure
29            .setSender(sender)
30            .setRoom(room)
31            .setText(message);
32
33        return (await modify.getCreator().finish(messageStructure));
34    }
35
36    private async getMessageExtender(context: SlashCommandContext, modify: IModify) {
37        const sender = context.getSender();
38        return modify.getExtender().extendMessage(messageId, sender);
39    }
40}

```

The main actions performed by the code above are:

- [1] Sends a message and stores the sent message's ID in the `messageId` variable;

- [2] Uses the sent message's ID. The `messageExtender` object is returned by the `getMessageExtender` method and stored in the `messageExtender` variable;
 - [3] Creates the attachment object as an instance of the `ImageAttachment` class. Here, the attachment is an image;
 - [4] Adds a custom field with the key `'key'` linked to the value `1` using the `addCustomField` method from the `messageExtender` object. Many more custom fields can be added to the same message just by calling the same method with distinct keys (and values of your choice);
 - [5] Adds the image attachment to the message using the `addAttachment` method from the `messageExtender` object. Many attachments can be added all at once using the `addAttachments` method;
 - [6] Finishes the `modifyExtender` object, which is crucial so as to apply the extensions made to the message and make them visible to the user.
-

Register the slash command

After having the slash command's code done, we have to register it in the app's main class, in the project's root.

```
1 import { App } from '@rocket.chat/apps-engine/definition/App';
2 import { IAppInfo } from '@rocket.chat/apps-engine/definition/metadata';
3 import { ExtendMessageCommand } from './slashcommands/ExtendSlashcommand';
4
5 export class RocketChatTester extends App {
6     constructor(info: IAppInfo, logger: ILogger, accessors: IAppAccessors)
7         super(info, logger, accessors);
8 }
9
10    public async extendConfiguration(configuration: IConfigurationExtend)
11        configuration.slashCommands.provideSlashCommand(new ExtendMessageC
12    }
13 }
```

Here we [1] import our new slash command class and then [2] register it in the app's configuration. Now it is available for us.

Deploy to the server

To deploy the app, run:

```
rc-apps deploy --url <server_url><server_port> -u <user> -p <pwd>
```

Finally: running the slash command

After performing the app's deployment, it is possible to run the slash command

```
/extend-message
```

in any channel.

Running the registered slash command will send a message in the current channel, which will be edited in order to add an image attachment and a custom field. The attached image can be seen in the edited message. In addition, the custom fields created can be found in any database client of your choice.

Making HTTP Requests

In this recipe we will create a slashcommand that runs a GET HTTP based on arguments you provide. We will use the [App.RocketChat.Tester](#) as the base project for it. All the code for this recipe is in the branch [recipes/http-requests](#).

You can see the complete documentation on HTTP requests in Rocket.Chat Apps at [IHttp's docs page](#).

Moving parts

You'll need to have handy:

- A Rocket.Chat server to deploy the app to
 - Our [Tester App](#) or a newly created app (your choice)
-

Create the slashcommand

Our command will be called `get`, so when we call it from the chat, we just have to type `/get <url>`.

First, let's create a `slashcommand` directory at the root of the project and add a file named `HTTPRequestCommand.ts`. Then add the following code:

HTTPRequestCommand.ts

```
1 import {
2     IHttp,
3     IModify,
4     IRead,
5 } from '@rocket.chat/apps-engine/definition/accessors';
6 import {
7     ISlashCommand,
```

```
8     SlashCommandContext,
9 } from '@rocket.chat/apps-engine/definition/slashcommands';
10
11 export class HTTPRequestCommand implements ISlashCommand {
12     public command = 'get'; // [1]
13     public i18nParamsExample = '';
14     public i18nDescription = '';
15     public providesPreview = false;
16
17     public async executor(context: SlashCommandContext, read: IRead, modify: IModify) {
18         const [url] = context.getArguments(); // [2]
19
20         if (!url) { // [3]
21             throw new Error('Error!');
22         }
23
24         await http.get(url); // [4]
25     }
26 }
```

What this code tells us:

- [1] The command will be called `get`.
- When it gets executed, [2] it gets the argument the user passed after the command and use it as the url
- [3] The argument is mandatory
- [4] Make the get request with the provided argument

Learn more about HTTP requests in Rocket.Chat Apps at [IHttp's docs page](#).

Optionally, you can store the GET request in a constant to `console.log` it when the command gets executed.

```
1 const response = await http.get(url);
2 console.log("result: " + response.data);
```

Register the slashcommand

After getting the slashcommand's code done, we have to register it at the app's main class, at the project's root.

```
1 import { IAppAccessors, IConfigurationExtend, ILogger } from '@rocket.chat/fuselage';
2 import { App } from '@rocket.chat/apps-engine/definition/App';
3 import { IAppInfo } from '@rocket.chat/apps-engine/definition/metadata';
4 import { HTTPRequestCommand } from './slashcommands/HTTPRequestCommand';
5
6 export class RocketChatTester extends App {
7     constructor(info: IAppInfo, logger: ILogger, accessors: IAppAccessors) {
8         super(info, logger, accessors);
9     }
10
11     public async extendConfiguration(configuration: IConfigurationExtend) {
12         configuration.slashCommands.provideSlashCommand(new HTTPRequestCom
13     }
14 }
```

Here we [1] import our new slashcommand class and then [2] register it in the app's configuration. Now it is available for us.

Deploy to the server

To deploy the app, run:

```
rc-apps deploy --url <server_url><server_port> -u <user> -p <pwd>
```

Finally: calling the slashcommand

After deploying the app, you can type in any channel `/get <some_url>` and the app will make a GET request to the provided url.

In this example we will use **JSONPlaceholder** to get some fake data to test our app:

- Enter `/get https://jsonplaceholder.typicode.com/todos/1` in a chat

- Your Rocket.Chat instance will print out to the console the following:

```
result: {"userId":1,"id":1,"title":"delectus aut autem","completed":false}
```

If you don't see the result, enable info logs in your instance at *Administration > Logs > Log Level* by selecting "1 - Errors and Information".

Bonus: print the request in the chat

Now, instead of console logging data to the instance's log, let's display the output in the chat.

In the `HTTPRequestCommand.ts`, add the following private method:

```
1  private async sendMessage(context: SlashCommandContext, modify: IModify) {
2      const messageStructure = modify.getCreator().startMessage();
3      const sender = context.getSender(); // [1]
4      const room = context.getRoom(); // [2]
5
6      messageStructure
7          .setSender(sender)
8          .setRoom(room)
9          .setText(message); // [3]
10
11     await modify.getCreator().finish(messageStructure); // [4]
12 }
```

This function **[1]** gets the user who called the command (in this case, yours), **[2]** selects the room in which the command was run, **[3]** sets the received string as the message and then **[4]** sends the message in the room.

To know more about messaging, you can go to the [IMessageBuilder docs](#).

After that, append this code at the end of the `executor` method:

```
1  const response = await http.get(url); // [1]
```

```
2     const message = JSON.stringify(response.data, null, 2); // [2]
3     await this.sendMessage(context, modify, message); // [3]
```

Instead of just making the request and not catching the response data, **[1]** we store the response in a constant, then **[2]** format its content as a string to **[3]** send it using our new `sendMessage` method.

Save the file and redeploy the app by running:

```
rc-apps deploy --url <server_url><server_port> -u <user> -p <pwd> --update
```

Now, when you run the slashcommand

```
/get https://jsonplaceholder.typicode.com/todos/1 , you'll receive the response in the chat, as your user, like the following:
```

```
1  {
2    "userId": 1,
3    "id": 1,
4    "title": "delectus aut autem",
5    "completed": false
6 }
```

Storing User Input

Introduction

The Rocket.Chat Apps-Engine provides `persistenceRead: IPersistenceRead` and `persistence: IPersistence` to help you read/write data from or to the RocketChat database.

We can get `persistenceRead: IPersistenceRead` through the following way:

```
1 // Get a persistence reader if you are using it in a method
2 // Here `this` means the main App class instance
3 const persistenceRead = this.getAccessors().reader.getPersistenceReader();
4
5 // Some methods provideds `read: IRead` parameter, so that you get a persi
6 // reader through this parameter too.
7 const persistenceRead = read.getPersistenceRead();
```

For `persistence: IPersistence`, you can only obtain it through parameter approach, which means you can not persist data within a method (typically is an event handler that you are going to implement) if the method doesn't have a `persistence: IPersistence` parameter.

```
1 function someMethod(context, read: IRead, persistence: IPersistence) {
2     console.log(persistence); // The only way to fetch a persistence write
3 }
```

Examples

Below is a complete example to show how we can manage persistence methods with a class. Imagine that you are going to persist some messages. You can create a class called

`MessagePersistence` or whatever name. Then you can add a series of static methods like below to `add/remove/query` data from the database.

```
1 import { IPersistence, IPersistenceRead } from '@rocket.chat/apps-engine/d
2 import { RocketChatAssociationModel, RocketChatAssociationRecord } from '@
3 import { IRoom } from '@rocket.chat/apps-engine/definition/rooms';
4
5 export class MessagePersistence {
6     // add a record
7     public static async persist(persis: IPersistence, room: IRoom, id: str
8         const associations: Array<RocketChatAssociationRecord> = [
9             new RocketChatAssociationRecord(RocketChatAssociationModel.MIS
10            new RocketChatAssociationRecord(RocketChatAssociationModel.ROO
11            new RocketChatAssociationRecord(RocketChatAssociationModel.MIS
12        ];
13
14        try {
15            await persis.updateByAssociations(associations, { id }, true);
16        } catch (err) {
17            console.warn(err);
18            return false;
19        }
20
21        return true;
22    }
23
24    // query all records within the "scope" - message
25    public static async findAll(persis: IPersistenceRead): Promise<Array<s
26        const associations: Array<RocketChatAssociationRecord> = [
27            new RocketChatAssociationRecord(RocketChatAssociationModel.MIS
28        ];
29
30        let result: Array<string> = [];
31        try {
32            const records: Array<{ id: string }> = (await persis.readByAss
33
34            if (records.length) {
35                result = records.map(({ id }) => id);
36            }
37        } catch (err) {
38            console.warn(err);
39        }
40
41        return result;
42    }
43
44    // query all records by room within the "scope" - message
45    public static async findByRoom(persis: IPersistenceRead, room: IRoom):
```

```

46     const associations: Array<RocketChatAssociationRecord> = [
47         new RocketChatAssociationRecord(RocketChatAssociationModel.MIS)
48         new RocketChatAssociationRecord(RocketChatAssociationModel.ROO)
49     ];
50
51     let result: Array<string> = [];
52     try {
53         const records: Array<{ id: string }> = (await persis.readByAss
54
55         if (records.length) {
56             result = records.map(({ id }) => id);
57         }
58     } catch (err) {
59         console.warn(err);
60     }
61
62     return result;
63 }
64
65 // query all records by room within the "scope" - message
66 public static async removeByRoom(persis: IPersistence, room: IRoom): P
67     const associations: Array<RocketChatAssociationRecord> = [
68         new RocketChatAssociationRecord(RocketChatAssociationModel.MIS)
69         new RocketChatAssociationRecord(RocketChatAssociationModel.ROO)
70     ];
71
72     try {
73         await persis.removeByAssociations(associations);
74     } catch (err) {
75         console.warn(err);
76         return false;
77     }
78
79     return true;
80 }
81
82 // remove all records by id within the "scope" - message
83 public static async removeById(persis: IPersistence, id: string): Prom
84     const associations: Array<RocketChatAssociationRecord> = [
85         new RocketChatAssociationRecord(RocketChatAssociationModel.MIS)
86         new RocketChatAssociationRecord(RocketChatAssociationModel.MIS)
87     ];
88
89     try {
90         await persis.removeByAssociations(associations);
91     } catch (err) {
92         console.warn(err);
93         return false;
94     }
95
96     return true;

```

```
97      }
98
99      // remove all records within the "scope" - message
100     public static async clear(persis): Promise<boolean> {
101         const associations: Array<RocketChatAssociationRecord> = [
102             new RocketChatAssociationRecord(RocketChatAssociationModel.MIS
103         ];
104
105         try {
106             await persis.removeByAssociations(associations);
107         } catch (err) {
108             console.warn(err);
109             return false;
110         }
111
112         return true;
113     }
114 }
```

Registering API endpoints

Introduction

An API endpoint is a location from which APIs can access the resources they need to carry out their function or can post data to it.

In this recipe, we will register a public API endpoint that receives data from external HTTP requests. We will try to forward the data received from the endpoint to the `#general` channel.

Register an API endpoint

First of all, let's create a new Rocket.Chat app from scratch using `rc-apps create`. (If `rc-apps` is an unrecognized command, please check out the [Getting Started](#) section to make initial preparations first).

In the main App class, we need to implement the `extendConfiguration` method, within which we use `configuration.api.provideApi` to register a new API endpoint

```
new Endpoint(this) .
```

```
1 // Main App Class
2 import { IAppAccessors, IConfigurationExtend, ILogger } from '@rocket.chat/fuselage';
3 import { ApiSecurity, ApiVisibility } from '@rocket.chat/apps-engine/definitions';
4 import { App } from '@rocket.chat/apps-engine/definition/App';
5 import { IAppInfo } from '@rocket.chat/apps-engine/definition/metadata';
6
7 import { Endpoint } from './endpoint';
8
9 export class RocketChatTester extends App {
10     constructor(info: IAppInfo, logger: ILogger, accessors: IAppAccessors) {
11         super(info, logger, accessors);
12     }
13
14     public async extendConfiguration(configuration: IConfigurationExtend) {
15         // Register API endpoints
16         configuration.api.provideApi({
```

```

17         visibility: ApiVisibility.PUBLIC,
18         security: ApiSecurity.UNSECURE,
19         endpoints: [new Endpoint(this)],
20     });
21 }
22 }
```

Notice that we imported an `Endpoint` class from another file. Now let's create a new file named `endpoint.ts` and implement the class `Endpoint` we used above.

```

1 // endpoint.ts
2 import { HttpStatusCode, IHttp, IModify, IPersistence, IRead } from '@rock
3 import { ApiEndpoint, IApiEndpointInfo, IApiRequest, IApiResponse } from '
4
5 export class Endpoint extends ApiEndpoint {
6     public path = 'api';
7
8     public async post(
9         request: IApiRequest, endpoint: IApiEndpointInfo, read: IRead, mod
10    ): Promise<IApiResponse> {
11         const body = Object.entries(request.content)
12             .map(([key, value]) => `${key}: ${value}`)
13             .join('\n');
14         const room = await read.getRoomReader().getByName('general');
15
16         if (!room) {
17             return {
18                 status: HttpStatusCode.NOT_FOUND,
19                 content: `Room "#general" could not be found`,
20             };
21         }
22
23         const messageBuilder = modify.getCreator().startMessage()
24             .setText(body)
25             .setRoom(room);
26         const messageId = await modify.getCreator().finish(messageBuilder)
27
28         return this.success(JSON.stringify({ messageId }));
29     }
30 }
```

In the file `endpoint.ts`, we created a class that extends the base class `ApiEndpoint`. We defined the path of the API endpoint by assigning the value `'api'` to the public property

```
path .
```

Then, let's implement the method `post`, which will be executed every time once the API endpoint receives an HTTP POST request from an external service. Following our original idea - forwarding any data received to the `#general` channel, we obtain the request content by `request.content` and create a message containing the request content, then send it to the channel `#general`.

Test the result

Using the CLI command `rc-apps deploy` to deploy the app to your Rocket.Chat server first. Navigate to *Rocket.Chat Admin -> Apps -> Apps Detailed Page* to check the full endpoint URL you registered for the App.

APIs

POST api

```
curl -X POST http://localhost:3000/api/apps/public/bc4dd4a1-bf9b-408e-83a4-aba7eba0bf02/api
```

Open the terminal and use curl post some data to the endpoint. The result should be like below:

```
1 > curl --data 'Jack=Hello :)&Lucy=Hi!' -X POST http://localhost:3000/api/a
2 {"messageId":"dREmKaR7qHyN98rtZ"}
```



appsrocketchattester.bot 11:49 PM

Jack: Hello 😊

Lucy: Hi!

Managing Internal State

It's a common need to manage internal state in your app. **Storing app's state in the memory inside of an app is not recommended at all** because a Rocket.Chat server can have several instances and each server has its own instance of an app. If the data, such as confirmation for context, stored in the memory between the different instances then the problem occurs. There're 2 approaches here suggested to be applied to solve the problem above.

Persistence APIs (Recommended)

There's **IPersistence** interface provided for creating/updating records in the persistence storage and **IPersistenceRead** interface provided for reading records. In the exmaple below, we'll teach you how to manage your app's internal state using persistence APIs.

Assuming that we're creating an app which records how many messages on the server sent, we can write a PostMessageSent event handler like bellow:

```
1  public async executePostMessageSent(message: IMessage, read: IRead, http: IHTTP) {
2      const association = new RocketChatAssociationRecord(RocketChatAssociations);
3      const persis = read.getPersistenceReader();
4      try {
5          let count = 0;
6          const record = await persis.readByAssociation(association);
7          if (record?.count) count = record.count;
8
9          await persistence.createWithAssociation({ count: count + 1 }, association);
10         console.log(`Message Sent Count: ${ count }`)
11     } catch (err) {
12         return this.getLogger().error(err);
13     }
14 }
15 }
```

Here, the internal state is "count" but not the count variable whose data stored in the memory. We use the temporary variable "count" for storing the number of messages sent retrieving from the persistence. Every time the handler executePostMessageSent called, we increase the count by 1 and then store it back to the persistence storage.

In this way, even in a cluster environment, your app inside each Rocket.Chat instance can share data from a single data source - Rocket.Chat persistence storage and maintain the data consistency.

[Check the full demo here for more details!](#)

Message customFields

Besides Persistence APIs, `customFields` is a public way of storing data related to a message. You can attach some custom "attributes" (called `customFields`) to a message by creating/modifying the message. You should only use the `customFields` if you're ok with potentially having them read and overwritten by someone else!

Slack Compatibility



This is still a draft and doesn't cover all subjects about Slack Compatibility yet

Would you like to have your app listed in [Rocket.Chat's Marketplace](#) but don't want to rewrite all the backend for your Slack listing?

Look no further!

This "compatibility layer" will help you make your Rocket.Chat App talk to your backend in no time :)

Initialize your Rocket.Chat App with bindings that make it compatible with your Slack App implementation.

Installation

First of all, you're gonna need to scaffold out a Rocket.Chat App directory, so make sure you follow our [Getting Started](#) guide (it's really quick).

We're gonna build upon the example in our Getting Started guide with the LiftOff app. After you've created the new app with our [CLI](#), you should have the following folder structure:

```
1  liftoff/
2  |  .vscode/
3  |  node_modules/
4  |  .editorconfig
5  |  .gitignore
6  |  LiftoffApp.ts
7  |  app.json
8  |  icon.png
9  |  package-lock.json
10 |  package.json
11 |  tsconfig.json
```

```
12 | tslint.json
```

Now, in your app's directory, you can install the Slack Compatibility Layer:

```
$ npm install RocketChat/slack-compatibility-for-apps
```

-  This will install the package directly from our [repository on GitHub](#). We will publish it to NPM when it gets a bit more feature complete and battle tested.

That's almost it! Rocket.Chat Apps cannot include npm packages yet, so this package will copy itself to a `vendor` folder in your app so that you can use it. Your folder structure should be like this now:

```
1 liftoff/
2   | .vscode/
3   | node_modules/
4   | vendor/
5   |   | slack-compatible-layer/
6   |   |   | src/
7   |   |   | vendor/
8   |   |   |   | SlackCompatibleApp.ts
9   |   |   | .editorconfig
10  |   |   | .gitignore
11  |   |   | LiftoffApp.ts
12  |   |   | app.json
13  |   |   | icon.png
14  |   |   | package-lock.json
15  |   |   | package.json
16  |   |   | tsconfig.json
17  |   |   | tslint.json
```

Now you just have to extend the `SlackCompatibleApp` class instead of the default `App` class from the Apps-Engine - this will make your Rocket.Chat App understand "Slack

language". The only thing left to do is config the main class of your app with the features you have:

```
1 import {
2     IAppAccessors,
3     ILogger,
4 } from '@rocket.chat/apps-engine/definition/accessors';
5 // import { App } from '@rocket.chat/apps-engine/definition/App';
6 import { IAppInfo } from '@rocket.chat/apps-engine/definition/metadata';
7
8 import { SlackCompatibleApp as App } from './vendor/slack-compatible-layer'
9
10 export class LiftOffApp extends App {
11     public config = {
12         interactiveEndpoint: 'https://my.example.app/interaction',
13         slashCommands: [
14             {
15                 command: 'liftoff',
16                 requestURL: 'https://my.example.app/liftoff.command',
17                 shortDescription: 'Tells the user if it is time to liftoff'
18             }
19         ],
20     }
21
22     constructor(info: IAppInfo, logger: ILogger, accessors: IAppAccessors)
23         super(info, logger, accessors);
24     }
25 }
```

That's it!

Permission System

How it works

Each feature available in the Apps Engine is mapped to a permission. Adding the permission in the app's manifest (`app.json` file) will unlock said feature to be used by the app. For example, if you add "send messages" as a permission in your app's manifest, your app will be able to send messages in the Rocket.Chat server. If your app tries to send a message without having the "send messages" permission listed in the manifest, the Apps Engine will block the feature and the app will not be able to perform the desired action.

When installing an app, being it by the Marketplace page or installing it manually via zip file, a window asks the user to review the permissions the app requires to work properly. The user installing the app can either accept and install the app or deny the permissions and not install it.

How to use it

In your app's manifest file (`app.json`), add the field `permissions`. It receives a list containing all the permissions you are asking for. Example:

```
1  {
2  ...
3  "permissions": [
4      {
5          "name": "user.read"
6      },
7      {
8          "name": "user.write"
9      },
10     {
11         "name": "upload.read"
12     },
13 ],
14 ...
15 }
```

Notice about rule enforcement

As of right now, the Apps Engine does not enforce the use of the permissions to make your apps work, meaning your current apps (including the ones published at the Marketplace) will still work as expected. All the apps that do not have the permissions listed in their manifest will still have access to all the features they need through the `default permissions`. They are applied to your app automatically when no `permissions` property is found in the app's manifest.

Until no sunsetting period for the "permissionless engine" is defined, setting permissions in your app's manifest will be optional.

Permissions

Access user information

Permission name: `user.read`

```
1  {
2    "permissions": [
3      {
4        "name": "user.read"
5      }
6    ]
7 }
```

Modify user information

Permission name: `user.write`

```
1  {
2    "permissions": [
3      {
4        "name": "user.write"
5      }
6    ]
7 }
```

```
5      }
6    ]
7  }
```

Access files uploaded to the server

Permission name: `upload.read`

```
1  {
2    "permissions": [
3      {
4        "name": "upload.read"
5      }
6    ]
7  }
```

Upload files to the server

Permission name: `upload.write`

```
1  {
2    "permissions": [
3      {
4        "name": "upload.write"
5      }
6    ]
7  }
```

Access settings in the server

Permission name: `server-setting.read`

```
1  {
2    "permissions": [
3      {
```

```
4      "name":"server-setting.read"
5    }
6  ]
7 }
```

Modify settings in the server

Permission name: `server-setting.write`

```
1  {
2    "permissions": [
3      {
4        "name":"server-setting.write"
5      }
6    ]
7 }
```

Access room information

Permission name: `room.read`

```
1  {
2    "permissions": [
3      {
4        "name":"room.read"
5      }
6    ]
7 }
```

Create and modify rooms

Permission name: `room.write`

```
1  {
2    "permissions": [
```

```
3      {
4          "name":"room.write"
5      }
6  ]
7 }
```

Access messages

Permission name: `message.read`

```
1  {
2      "permissions": [
3          {
4              "name":"message.read"
5          }
6      ]
7 }
```

Send and modify messages

Permission name: `message.write`

```
1  {
2      "permissions": [
3          {
4              "name":"message.write"
5          }
6      ]
7 }
```

Access Livechat status information

Permission name: `livechat-status.read`

```
1  {
```

```
2   "permissions": [
3     {
4       "name":"livechat-status.read"
5     }
6   ]
7 }
```

Modify Livechat custom field configuration

Permission name: `livechat-custom-fields.write`

```
1  {
2   "permissions": [
3     {
4       "name":"livechat-custom-fields.write"
5     }
6   ]
7 }
```

Access Livechat visitor information

Permission name: `livechat-visitor.read`

```
1  {
2   "permissions": [
3     {
4       "name":"livechat-visitor.read"
5     }
6   ]
7 }
```

Modify Livechat visitor information

Permission name: `livechat-visitor.write`

```
1  {
2    "permissions": [
3      {
4        "name":"livechat-visitor.write"
5      }
6    ]
7 }
```

Access Livechat message information

Permission name: `livechat-message.read`

```
1  {
2    "permissions": [
3      {
4        "name":"livechat-message.read"
5      }
6    ]
7 }
```

Modify Livechat message information

Permission name: `livechat-message.write`

```
1  {
2    "permissions": [
3      {
4        "name":"livechat-message.write"
5      }
6    ]
7 }
```

Access Livechat room information

Permission name: `livechat-room.read`

```
1  {
2    "permissions": [
3      {
4        "name":"livechat-room.read"
5      }
6    ]
7 }
```

Modify Livechat room information

Permission name: `livechat-room.write`

```
1  {
2    "permissions": [
3      {
4        "name":"livechat-room.write"
5      }
6    ]
7 }
```

Access Livechat department information

Permission name: `livechat-department.read`

```
1  {
2    "permissions": [
3      {
4        "name":"livechat-department.read"
5      }
6    ]
7 }
```

Modify Livechat department information

Permission name: `livechat-department.write`

```
1  {
2    "permissions": [
3      {
4        "name":"livechat-department.write"
5      }
6    ]
7 }
```

Register new slash commands

Permission name: `slashcommand`

```
1  {
2    "permissions": [
3      {
4        "name":"slashcommand"
5      }
6    ]
7 }
```

Register new HTTP endpoints

Permission name: `apis`

```
1  {
2    "permissions": [
3      {
4        "name":"apis"
5      }
6    ]
7 }
```

Access minimal information about the server environment

Permission name: `env.read`

```
1  {
2    "permissions": [
3      {
4        "name":"env.read"
5      }
6    ]
7 }
```

Access to the server network

Permission name: `networking`

```
1  {
2    "permissions": [
3      {
4        "name":"networking"
5      }
6    ]
7 }
```

Store internal data in the database

Permission name: `persistence`

```
1  {
2    "permissions": [
3      {
4        "name":"persistence"
5      }
6    ]
7 }
```

Register and maintain scheduled jobs

Permission name: `scheduler`

```
1  {
2    "permissions": [
3      {
4        "name":"scheduler"
5      }
6    ]
7 }
```

Interact with the UI (UIKit)

Permission name: `ui.interact`

```
1  {
2    "permissions": [
3      {
4        "name":"ui.interact"
5      }
6    ]
7 }
```

UIKit

→ **UIKit building blocks**

/apps-development/uikit/uikit-building-blocks

→ **Slack Compatibility Layer's unsupported properties**

/apps-development/uikit/slack-compatibility-layers-unsupported-properties

UIKit building blocks

Objects

Text object

Field	Type	Required?	Description
type	String	Yes	The available values are <code>plain_text</code> and <code>mrkdwn</code>
text	String	Yes	The actual text
emoji	Boolean	No	Works with the <code>plain_text</code> type

Example

```
1  {
2    type: 'plain_text',
3    text: 'lorem ipsum ;',
4    emoji: true,
5 }
```

Option object

Field	Type	Required?	Description
text	Object	Yes	The <code>text</code> object to be displayed at the menu. Can be either <code>plain_text</code> or <code>mrkdwn</code>
value	String	Yes	The actual value the option represents

Example

```
1  {
2    value: 'option_1',
3    text: {
4      type: 'plain_text',
5      text: 'lorem ipsum \u2019',
6      emoji: true,
7    }
8 }
```

Block elements

button

Field	Type	Required?	Description
type	String	Yes	The type of the block element, that in this case is 'button'
text	Object	Yes	The <code>text object</code> to be displayed in the button. Can be either <code>plain_text</code> or <code>mrkdwn</code>
value	String	No	A value sent along with the button information when an action is made upon the element
url	String	No	An URL *
style	String	No	The style of the button. Can be either <code>primary</code> or <code>danger</code>
actionId	String	Yes	An unique identifier for an action make upon the element

Example

```
1  {
2    type: 'button',
3    text: {
4      type: 'plain_text',
```

```
5           text: 'danger!',
6           emoji: true,
7     },
8   actionId: 'button_1_danger',
9   style: 'danger',
10 }
```

image

Field	Type	Required?	Description
type	String	Yes	The type of the block element, that in this case is 'image'
imageUrl	String	Yes	The URL of the image
altText	String	Yes	A text describing the image being displayed

Example

```
1  {
2   type: 'image',
3   imageUrl: 'https://picsum.photos/200/300',
4   altText: 'An image',
5 }
```

overflow menu

Field	Type	Required?	Description
type	String	Yes	The type of the block element, that in this case is 'overflow'
options	Array	Yes	An array with the possible options (the option object)
actionId	String	Yes	An unique identifier for an action make upon the element

Example

```

1  {
2    type: 'overflow',
3    actionId: 'overflow_1',
4    options: [
5      {
6        value: 'option_1',
7        text: {
8          type: 'plain_text',
9          text: 'lorem ipsum 🍃',
10         emoji: true,
11       }
12     },
13     {
14       value: 'option_2',
15       text: {
16         type: 'plain_text',
17         text: 'lorem ipsum 🍃',
18         emoji: true,
19       }
20     },
21   ],
22 }

```

plain text input

Field	Type	Required?	Description
type	String	Yes	The type of the block element, that in this case is 'plain_text_input'
actionId	String	Yes	An unique identifier for an action make upon the element
placeholder	Object	Yes	A placeholder text for the input (plain text object)
initialValue	String	No	The initial value of the field
multiline	Boolean	No	A flag that indicates whether the field should be a single line (default) or a bigger text area

Example

```

1  {
2    type: 'plain_text_input',
3    actionId: 'plain_text_input_1',
4    placeholder: {
5      type: 'plain_text',
6      text: 'Enter name',
7      emoji: true,
8    },
9    initialValue: 'John Doe',
10   multiline: false,
11 }

```

static select menu

Field	Type	Required?	Description
type	String	Yes	The type of the block element, that in this case is 'static_select'
actionId	String	Yes	An unique identifier for an action make upon the element
placeholder	Object	Yes	A placeholder text for the input (plain text object)
initialValue	String	No	The initial value selected (value field from the option object)
options	Array	Yes	An array with the possible options (the option object)

Example

```

1  {
2    type: 'static_select',
3    actionId: 'overflow_1',
4    initialValue: 'option_2',
5    options: [
6      {
7        value: 'option_1',
8        text: {
9          type: 'plain_text',

```

```

10           text: 'lorem ipsum 🍎',
11           emoji: true,
12       }
13     },
14   {
15     value: 'option_2',
16     text: {
17       type: 'plain_text',
18       text: 'lorem ipsum 🍎',
19       emoji: true,
20     }
21   },
22 ],
23 placeholder: {
24   type: 'plain_text',
25   text: 'Select an item',
26 },
27 }
```

multi static select menu

Field	Type	Required?	Description
type	String	Yes	The type of the block element, that in this case is 'multi_static_select'
actionId	String	Yes	An unique identifier for an action make upon the element
placeholder	Object	Yes	A placeholder text for the input (plain text object)
initialValue	Array of strings	No	The initial values selected (value field from the option object)
options	Array	Yes	An array with the possible options (the option object)

Example

```

1  {
2    type: 'static_select',
```

```

3   actionId: 'overflow_1',
4   initialValue: ['option_1' , 'option_2'],
5   options: [
6     {
7       value: 'option_1',
8       text: {
9         type: 'plain_text',
10        text: 'lorem ipsum \u2019',
11        emoji: true,
12      }
13    },
14    {
15      value: 'option_2',
16      text: {
17        type: 'plain_text',
18        text: 'lorem ipsum \u2019',
19        emoji: true,
20      }
21    },
22  ],
23  placeholder: {
24    type: 'plain_text',
25    text: 'Select an item',
26  },
27 }

```

Blocks

section

Field	Type	Required?	Description
blockId	String	No	An unique identifier for the block
type	String	Yes	The type of the block, that in this case is 'section'
text	Object	Yes	The <code>text</code> object to be displayed in the button. Can be either <code>plain_text</code> or <code>mrkdwn</code>
accessory	Object	No	One element that can be a <code>button element</code> , an <code>image element</code> or a <code>overflow menu</code>

Example

```
1  {
2    type: 'section',
3    blockId: 'section_1',
4    text: {
5      type: 'plain_text',
6      text: 'lorem ipsum Ⅰ',
7      emoji: true,
8    }
9    accessory: { /* one of the accessory elements */ } ,
10 }
```

divider

Field	Type	Required?	Description
blockId	String	No	An unique identifier for the block
type	String	Yes	The type of the block, that in this case is 'divider'

Example

```
1  {
2    type: 'divider',
3    blockId: 'divider_1',
4 }
```

image

Field	Type	Required?	Description
blockId	String	No	An unique identifier for the block
type	String	Yes	The type of the block, that in this case is 'image'

imageUrl	String	Yes	The URL of the image
altText	String	Yes	A text describing the image being displayed
title	Object	No	The <code>text object</code> to be displayed as the image's title. Can be either <code>plain_text</code> or <code>mrkdn</code>

Example

```

1  {
2    type: 'image',
3    blockId: 'image_1',
4    imageUrl: 'https://picsum.photos/200/300',
5    altText: 'An image',
6    title: {
7      type: 'plain_text',
8      text: 'lorem ipsum 🌎',
9      emoji: true,
10    }
11  }

```

actions

Field	Type	Required?	Description
blockId	String	No	An unique identifier for the block
type	String	Yes	The type of the block, that in this case is 'actions'
elements	Array	Yes	A list of interactive block elements

Example

```

1  {
2    type: 'actions',
3    blockId: 'actions_1',
4    elements: [ /* block elements */ ]
5  }

```

context

Field	Type	Required?	Description
blockId	String	No	An unique identifier for the block
type	String	Yes	The type of the block, that in this case is 'context'
elements	Array	Yes	A list of block elements. Allowed elements are <code>plain text object</code> and <code>image element</code>

Example

```
1  {
2    type: 'context',
3    blockId: 'context_1',
4    elements: [ /* block elements */ ]
5 }
```

input

Field	Type	Required?	Description
blockId	String	No	An unique identifier for the block
type	String	Yes	The type of the block, that in this case is 'input'
element	Array	Yes	The input element, that can be <code>plain text input</code> , <code>static select menu</code> and <code>multi static select menu</code>

Example

```
1  {
2    type: 'input',
```

```
3   blockId: 'input_1',
4   element: { /* input element */ }
5 }
```

Slack Compatibility Layer's unsupported properties

Text object - Markdown

- verbatim (string)
-

Option object

- url (string)
 - description (PlainTextElement)
-

Button element

- confirm (confirm)
-

From the "select" menus, only multistatic and static works

Multistatic select

- option_groups
 - max_selected_items
 - confirm
-

Static select

- option_groups
 - confirm
-

Overflow menu

- confirm
-

Plain text input

- min_length
 - max_length
-

Input block

- hind (added)
-

Section block

- fields

Fuselage

Contributing

Starting from the beginning

To start the development you need to clone the project.

```
$ git clone git@github.com:RocketChat/Rocket.Chat.Fuselage.git
```

Then, if you browse packages you will notice that there are several packages within the same repository.

```
1 cd Rocket.Chat.Fuselage/packages/
2 ls
3 # css-in-js
4 # emitter
5 # fuselage
6 # fuselage-emitter
7 # fuselage-hooks
8 # fuselage-polyfills
9 # fuselage-tokens
10 # fuselage-ui-kit
11 # icons
12 # memo
13 # mp3-encoder
14 # ui-kit
```

How to build your project

```
1 # at root folder
2 yarn
```

How to develop and test under Rocket.Chat the local code

It's pretty easy, inside Rocket.Chat project just run:

```
meteor npm i ${path_to_fuselage_folder}/packages/package
```

How not to fail your pull request

The fuselage package is covered by visual regression tests. So, if you are changing any component, if there is any desired visual change, don't forget to update the images. Docker is required for that step.

```
packages/fuselage
```

```
1 yarn update-storybook
```

If you are creating something new, be sure to create a story for the component, it will not fail the pull request, but it will probably not be approved or even considered ready for review.

My pull request was merged, what now?

As soon as your contribution gets accepted, approved, and merged it will start the build process, even if your **PR** is only merged on develop. After that, the package will be available at **npm** tagged as **@next**. With that, you can already work/open pull requests on Rocket.Chat. At the end of Rocket.Chat release, we will release a fuselage production version and update the fuselage packages on Rocket.Chat.

