

# Zulip Bot Prize

Zulip will be giving \$100 for the best bot built on top of Zulip during the hackathon.

## Important

Follow the [submission guidelines](#) mentioned in the doc properly. You would not be eligible for the prize if the submission is not made according to the guidelines. The following category of students are also **not eligible** for this prize

- Did Google Summer of Code with Zulip in the past.
- Did Zulip Summer of Code or related programs funded by Zulip in the past.
- Was a Google Code-In winner or finalist with Zulip in the past.

## What is Zulip?

[Zulip](#) is the world's most productive chat application. The threaded conversation model of Zulip gives a topic to each and every conversation. This means you can have multiple conversations on the same channel. Since each conversation has a topic it is very easy to catch up even if you were away from Zulip for a long time.

Zulip is fully open source and has more than 400 contributors from all over the world making it the most active open source group chat application. Zulip also participates in [Google Summer of Code](#) and [Outreachy](#).

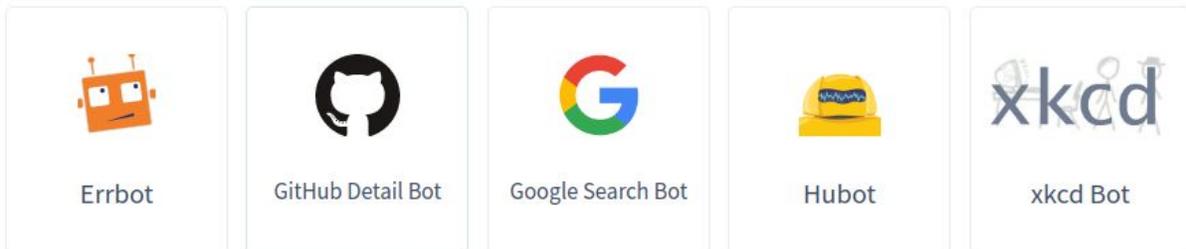
## What type of bots can I build?

You can build any bot you want. If you need some ideas here are some

- Checkin bot that will ping teams every couple of hours regarding their progress and gives a dashboard for organisers to view the responses.
- <https://twitter.com/hashtag/requestforbot>

We have received a lot of submissions in the past for bots that get the cricket score, weather, IMDB score etc (general theme being bot that makes request to an API service and returning back the result). Think out of the box if you want to win the prize :)

[Here](#) are some bots we already have



## How do I get start building a bot?

You can find all the resources for building a bot at <https://zulipchat.com/api>.

The easiest way to build a bot is using the REST API. [Zulip's REST API](#) is easy to work with directly, but there are API libraries available for various popular languages. The Python library is the most advanced (and has tools for easily writing interactive bots that react to messages), so we recommend it if you're trying to decide. You can install it by running

```
pip install zulip
```

Here are some examples of python zulip API library

<https://github.com/zulip/python-zulip-api/tree/master/zulip/zulip/examples>

Our recommended way to write a bot is using the [interactive bot framework](#). You can check the [source code](#) of some bots to have a better understanding.

You can obtain authentication credentials or `.zuliprc` for api/bot/integration by visiting <https://organization-name.zulipchat.com/#settings/your-bots>.

Replace organisation-name with the subdomain of the organisation. Ask the hackathon organisers for the subdomain if you are not sure.

## How do I run some bots to see how they work before starting to build my own bot?

This is pretty easy. You can run one of [these](#) bots to see how the bots work in Zulip. Check [this](#) guide to see how to run these bots.

## Can I build an [integration](#)?

Yes you can :)

## Can I build a bot just using the Zulip [REST API](#)?

Yes you can. In short you can use anything offered by Zulip to build the bot. The submission would be considered as long as your bot works.

## Where should I ask for help?

You can ask for help in <https://chat.zulip.org/#narrow/stream/integrations>

## Submission guidelines

- Should be a **working bot** built for Zulip
- Should be **open source** and must have clear instructions in the repository(GitHub, GitLab etc) on **how to run the bot**. If the bot is live and can be tested by us mention the relevant links to that as well in the README file.
- Create a **short video demo** of the bot and upload in Youtube or any video sharing site. Make sure that the person watching the video is fully able to understand what the bot does and what its functionalities are.
- Submit the link to source code and video of the bot in <https://chat.zulip.org/#narrow/stream/integrations> with topic **hackathon name - bot name**. (hackathon name and bot name should be replaced with the name of the hackathon and bot). Anyone in the team can make this submission.

## What is the judging criteria?

Ease of use, code cleanliness, functionality and idea are the main four factors. Each factor has equal importance. The judging would be done by the Zulip team.

## How would the money be transferred?

We would be following up with the hackathon organisers to collect the email of all the participants as well as payment details(UPI) in case your team win the prize :)

## Resources

- Repo of Zulip server - <https://github.com/zulip/zulip>
- API - <https://zulipchat.com/api>
- Zulip Community Server - [chat.zulip.org](https://chat.zulip.org)

- Zulip API Python Library - <https://github.com/zulip/python-zulip-api>