Improvement Ideas for TogetherJava

written by Leslie

Contents

1   Preamble

The following document is just a collection of ideas I've come up with or members of the community suggested at some point in time. They should not be taken as a task to fullfill. It is rather more my opinions on those things and open for discussion.

2   Discord Server

2.1   Tech channel

Split *#lobby* with *#tech_lobby*. So we can enforce or motivate more actual programming related talks. I kinda believe some people would be more interested in talking about Programming rather than just chit-chat. So create *#tech_lobby* and tell people to move when they start getting chatty instead of talking about programming.
Basically have 2 lobbies then apply some more moderator action in them.

2.2   Language channels

Some people requested a *#german* channel. Maybe this could be applied to other languages aswell when the need comes up. We could potentially make this channel hidden from the public to not bloat the chanellist. Then add a **German** role and make this channel only accessible for members with that role. Management of this role could either be performed by mods similar to the **Professional** role or even let people with the German role assign that role to others that want it, since it has no security risk or whatever.

2.3   Spam Czar

Community request

There were some occurances in the early morning hours that a troll came online when no mod was active. We are trying our best but at some times we just simply cannot react. We are also just human beings. It was requested that some members recieve a role **Spamczar** that is allowed to use the *¿spamlord* command.

2.4   Move Bots in Memberlist

The bots should be fairly on top. So people can see what bots we support currently. At least those on top should be our self provided bots. Very important here is if we are going to implement *Modmail Bot* members should see that we have such a bot that might resolve concerns.

2.5   Changes about Modertation mentality

Community request

There was a request for adding more mods. Also moderation should be totally hidden meaning, any discussion about why someone was banned or shall we ban him? etc only in modchat. The modteam in itself can be different but the outer shell shall stay consistent and seem as one entity. Rules should be reworked, some shall be better defined some rules should be based more on situational judgement f.e. politics.

2.6 General FAQ Channel

Community request

Something like a readonly channel that answers a lot of common asked questions. To add some new question people or maybe just top helpers can propose one. Since TopHelpers know the asking community the best usually.

3 Infrastructure

3.1 Github

3.1.1 TogetherJava Organisation

DJSakamura currently holds the TogetherJava Github organisation. We should ask him to somehow give it to us and make each mod owner whatever of that group. Just a role that is the equivalent of moderator or admin there. And make any future projects available there.

3.1.2 Bot situation

By my current knowledge we have **InoriChan**, **TogetherJavaBot** and **JShellBot** all distributed on 3 totally different Github Accounts. In my opinion we should move them all to the mentioned *organisation*. This makes it easier for interested members to actually make Pullrequests to the bots.

3.1.3 Issuesystem

We should actually enforce an issue system on the projects that we have on Github at the moment or in the future. First of all, we can perform bugtracking. Since f.e. InoriChan was thought of a funbot for newer members. Seeing which features or bugs are currently open might motivate them to say "Yes I'll try to fix this or implement that feature". Also if a bug was found we can just say "Please open a ticket" instead of trying to remember it and fix it when that one contributor got time. Making everything easier.

3.1.4 Force Pullrequests

By forcing Pullrequests we can ensure that a) the code stays at a certain quality since it is enforced to be revieweed and b) by commenting on the PRs newer committers also learn dos and don'ts of writing good code.

3.1.5 Force Correct Branching

Enforce a policy for **develop** and **master** branches. Pull Requests are all merged into develop. And after some reviewing and testing they will recieve a release and will be merged into master.

3.1.6 More contributors

Currently, the only ones able to review and approve Pullrequests are **Doppey** and **I Al Istannen**. This is in my opinion not enough to actually make Github work for this Server. We should at least give all moderators a Contributor role on all Together Java projects.
In my opinion make even trusted members a contributor if they are willing to do it. Trusted meaning, they know how to code. Also give them a *Trusted Committer* role or similar in Discord.

3.2 Server

3.2.1 Aquiring a V-Server

In my opinion it might even make sense to rent a V-Server for our Discord Guild. There were multiple occasions where InoriChan was offline and noone could take a look at why or get it online again. For just 5 bucks in total we could rent a V-Server that could host all our bots. Istannen and me can confirm Hetzner would be a suitable hoster. We can split up the 5 bucks between the mods. I would be willing to spend money on it.
Give if at all only Moderators sudo access on this machine. I'd suggest at least Istannen and me since we both got quite some good experience with Linux.

3.2.2 CI/CD

If we agree on the concerns stated in *Aquiring a V-Server* we should invest time in setting up a CI/CD Service like Jenkins. This would make the update process a lot easier. We could combine the gains of *Force correct branching* and *Force Correct Branching*.
A jenkins pipeline could eventually consist of a Github hook that triggers everytime a new release of the existing projects got pushed. This pipeline should also autmatically deploy the newest release as a bot f.e.

4 Projects

4.1 Existing Bots

4.1.1 Seperation of Duties

I personally don't feel comfortable having a bot that is pure for fun and a bot for important stuff like moderating in one bot. In my opinion we should move all moderating commands from InoriChan over to TogetherJavaBot since moderator commands might require a little more quality control than funcommands.

4.1.2 Modmail Bot

Create a bot that is soley there to have an anonymous way of sending messages to the moderators. Moderators can respond trough this bot without even knowing who is sitting on the other end.

4.1.3 Refactor InoriChan

We should start just opening Issues for small refactoring tasks on InoriChan. This also might motivate members to actually do something on our repos. On top we get some help with refactoring the Codebase and make it easier to extend it.

4.2 Moderation Tools

4.2.1 Make A Real Moderation Backend

The discord API interface is just an interface to communicate to Discoed. A good practice and project might be to make a full REST API Backend for the moderation tools which is also connected to the Bots. Include a database for stuffs like warnings, reports for stuff they said. Definetly crosslink some concerns like The context of where a warn was spoken out or reports and connect them in a full threatreport.

This Backend would improve the changabillity of software which is the whole purpose of having software. Maybe even make a webfrontend for it. So that it's not just soley fitted for our Discordserver.

4.3   Web

4.3.1   Website/Blog For The Server

It might be a good idea to have some smaller website for this Discordserver. We could combine it with a small blog where people can suggest ideas in our server and then get permission to write a blog article about some interesting topic. This could also be combined with *Spring Webapp*.

4.3.2   Spring Webapp

Community request

It was requested to create a webapp for this server in Spring. The intention was so that every skill level has a good time working on it. Spring is a good idea for experienced devs. It might also have some domains for rather inexperienced devs with a little help.

4.4   General

4.4.1   Promoting FOSS

Community request

Somehow we should promote contributing to Opern Source more. I think this is a great idea. The first step would be to start promoting our server own projects more. Further promoting FOSS on GitHub should also be important since it motivates people to actually get into bigger codebases and learn more.

4.4.2   Track Projects

Community request

If projects come together make a list of the projects offered by the community and keep track of them. So it is easy to find something cool or conrtibutors