

Getting Involved Guide for Developers

Community 1.5.1 release

Getting Involved Guide is intended to assist programmers looking to explore Jumbune code and contribute in Jumbune. They may also report bugs, join mailing lists, and/or provide feedback

Table of Contents

Introduction	3
Getting source code	3
Build the project	4
Become a committer	4
Report or Fix Bugs on Issue Tracker	5
Connect to social Jumbune	5

Introduction

As a developer you can improve Jumbune, fix bugs, or add features you are looking for. All these contributions help keep a project active and strengthen the community. The Jumbune team will therefore welcome and encourage participation, and attempt to make it as easy as possible for people to get involved.

You may contribute to Jumbune in following ways:

1. Join mailing lists
2. Report bugs / fix bugs
3. Become a contributor/committer
4. Provide feedback and improvement suggestions
5. Promote Jumbune

You can access Jumbune community on GitHub: <https://github.com/impetus-opensource/jumbune>

Getting source code

Configure your GitHub user id and public key before proceeding with checkout. For more help on configuring ssh-keys please visit <https://help.github.com/articles/generating-ssh-keys>

To get the source code, perform the following steps:

1. **Fork the Jumbune repository:**

To fork the project, simply click on the “Fork” button in GitHub repository.

2. **Clone your fork:**

```
git clone https://github.com/impetus-opensource/jumbune.git
```

3. **Configure remotes:**

```
mkdir jumbune
```

```
cd jumbune
```

```
git remote add upstream https://github.com/impetus-opensource/jumbune.git
```

```
git fetch upstream
```

Build the project

Jumbune is built using maven. Once you have maven configured, you can start building Jumbune.

“**package**” goal will compile the code and package it into a target directory. This is useful for running test cases. Tests results are available at the location “\target\surefire-reports”.

Below is the command for your reference:

```
mvn clean package
```

Install command will compile and package, it will also put the package in your local repository. Other projects can refer to it and grab it from your local repository. This can be done by using the command:

```
mvn clean install
```

For building YARN version of Jumbune use the following command

```
mvn clean install -P yarn
```

Become a committer

Coding contributions are welcome. Refer to general coding [guidelines](#) to get acquainted with the process.

To commit code in Jumbune, perform the following steps:

1. **Create a private branch**

```
git checkout -b branch-name
```

Once you complete enhancements on a feature or bug fix, you may push commits and initiate a pull request.

2. **Commit the changes into private branch:**

```
git commit -m 'User code changes comment here'
```

```
git push origin HEAD:branch-name
```

3. **Pull request:**

Select “pull request”, switch to your branch and compare your changes with the master. Submit pull request once you complete self-review.

Our developers will then merge your code or will guide you through improving the patch. The patch will be of course committed under your name and mail, so that you receive appropriate credits.

Report or Fix Bugs on Issue Tracker

Jumbune has an open JIRA, you can access Jumbune project: <http://jumbune.org/jira/JUM>. If you come across a potential issue while using Jumbune, have a suggestion for enhancement of existing functionality, or want to propose a new feature, report to Jira. Or you can pick an issue, create a new issue or feature request, and submit patch for issue or new feature.

Connect to social Jumbune

Twitter

- Follow us on Twitter: **@jumbune**
- Hashtag Jumbune with: **#jumbune**

LinkedIn

- Join Jumbune's LinkedIn group: <http://linkd.in/1mUmcYm>

Forums

- Users: To join send an email to, users-subscribe@collaborate.jumbune.org
- Dev: To join send an email to, dev-subscribe@collaborate.jumbune.org
- Issues: To join send an email to, issues-subscribe@collaborate.jumbune.org