

Issue Triage

This document describes the process Meteor contributors use to organize issues. We use Github [issues](#) in this repository to track bugs, and [Discussions](#) to track feature requests. Our goal is to maintain a list of issues that are relevant and well-defined (and [labeled](#)) such that a contributor can immediately begin working on the code for a fix or feature request. Contributors who want to dive in and write code aren't likely to prioritize working on issues that are ambiguous and have low impact.

You can filter issues that are missing triage using this [filter](#): `is:issue is:open -label:confirmed -label:not-ready -label:in-discussion -label:needs-reproduction -label:ready -label:in-development -label:pending-tests -label:waiting-feedback`

We would love to have more contributors who are willing to help out with triaging issues. You can begin by helping issue requesters create good reproductions and by confirming those reproductions on your own machine. It won't be long before the core maintainers notice your work and ask whether you'd like to be promoted to an issue maintainer.

- [Issue lifecycle](#)
 - [Bugs](#)
 - [Help questions](#)
 - [Feature requests](#)
- [Classification](#)
 - [Severity](#)
 - [Impact](#)
- [Issues ready to claim](#)

Lifecycle

All issues follow the flow outlined below. Your job as an issue maintainer is to work with the requester and others within the community towards the goal of having an issue either become 'claimable' or closed. Read on for more details on the process.



Flowchart

The first step is in determining whether the issue is a bug, help question or feature request. Read on for more details.

Bugs

1. Duplicates should be closed and marked as such.
2. Add the `Type:Bug` label and `Project:*` labels that apply (a best guess on the `Project:` is fine; sometimes it's hard to tell exactly which project the issue falls under).
3. Bugs should have a high-quality reproduction as described [here](#). You may need to help the reporter reduce their bug to a minimal reproduction. Leave the issue open.
4. A reproduction should be confirmed by at least one person other than the original reporter. Run the reproduction and validate that the bug exists; then make a note of your findings on the issue. If a reproduction is supplied but doesn't work, make a comment describing what happened.
5. Finally, add a status label and also classification, read more about labels [here](#).

Help questions

Our [forums](#) or Community Slack are the place to ask for help on using the framework. Close issues that are help requests and politely refer the author to the above locations.

Feature requests

1. For reasons described [here](#), we would prefer features to be built as separate packages. If the feature can clearly be built as a package, explain this to the requester and close the issue.
 - *If the feature could be built as a package and serves a particular need, encourage the user to contribute it themselves.*
 - *If the underlying issue could be better solved by existing technology, encourage them to seek help in the [forums](#).*
2. If you haven't closed the issue, add `Project:*` labels that apply (a best guess on the `Project:` is fine, sometimes it's hard to tell exactly which project the issue falls under).
3. If it's not possible to build the feature as a package (as you identified in step 1), explore whether creating hooks in core would make it possible to do so. If it would, redefine the issue as a request to create those hooks.
4. Work with the requester and others in the community to build a clear specification for the feature and update the issue description accordingly.
5. Finally, add a status label and also classification, read more about labels [here](#).

Is it ready?

- If the desired fixed or solution is already clear add the label `ready`.
- If discussions are still necessary add the label `in-discussion` and start the discussion.

Issues ready to claim

This state indicates that bugs/feature requests have reached the level of quality required for a contributor to begin writing code against (you can easily filter for [Ready](#))

Although this should have already been done by this stage, ensure the issue is correctly labeled and the title/description have been updated to reflect an accurate summary of the issue.

Contributors should comment on and/or assign themselves an issue if they begin working on it so that others know work is in progress.