

Labels

Labels are used to organize our issues and PRs.

We should change the labels of issues and PRs when their status changes.

Status Labels

Labels to indicate the status of a specific issue or PR. These are the most important labels as they tell us in which stage a specific item is at the moment at a glance.

You can filter issues that are missing status labels 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`

Stage 1

- `confirmed` : We want to fix or implement it
- `not-ready` : Something is missing, we are not able to work on this issue yet
- `in-discussion` : We are still discussing how to solve or implement it
- `needs-reproduction` : We can't reproduce so it's blocked
- `invalid` : We don't need to analyze

Stage 2

- `ready` : We've decided how to solve or implement it
- `in-development` : We are already working on it

Stage 3

- `pending-tests` : Tests are not passing, stuck or we need new tests
- `waiting-feedback` : It's implemented but we need feedback that it is working as expected

Classification Labels

Assign a classification (via GH labels) that enables the community to determine how to prioritize which issues to work on. The classification is based on *Severity x Impact* .

Severity

[Severity:has-workaround, Severity:production, Severity:blocks-development]

- If there is a workaround, apply the `Severity:has-workaround` label.
- If the issue affects production apps, apply the `Severity:production` label.
- If the issue blocks development (e.g `meteor run` is broken), apply the `Severity:blocks-development` label.

Impact

[Impact:few, Impact:some, Impact:most]

This is a somewhat subjective label and is interpreted in conjunction with Github's upvotes. As a general guideline:

- `Impact:few` issues would go unnoticed by almost all users, apart from those using a very niche feature, or a feature in an unusual way.
- `Impact:some` issues would impact users using a feature that is commonly but not universally used.
- `Impact:most` issues would impact more or less every user of the framework.

Type

[Type:Bug, Type:Feature]

As a general guideline:

- `Type:Bug` a problem is happening because of an issue in Meteor code.
- `Type:Feature` a new behavior or functionality is desired.

Project Labels

They start with `Project:` and they are used to inform the parts of Meteor that are involved in this item.

Special Labels

- `good first issue` : Used to indicate items friendly to beginners in Meteor
- `hacktoberfest-accepted` : Used to indicate items accepted for [Hacktoberfest](#)