Mergeable Documentation

Release 1.1

Justin Law, Shine Lee

May 12, 2023

Contents

Mergeable helps automate your team's GitHub workflow without a single line of code.

Usage

1. [Install](#) the Mergeable GitHub App.

2. Create your recipe(s) using *Configuration* or check out some ready to use examples at *Recipes*.

3. Commit and push the recipes to your repository at `.github/mergeable.yml`

4. (Optional) You can also create a default configuration for an organisation by creating a repo called *.github* and adding your file there. See *Organisation-wide defaults* for details.

Note: You can also deploy to your own server. See deploying

Configuration

Mergeable is **highly** configurable.

First, you'll need to start by creating a `.github/mergeable.yml` file in your repository.

Hint: Check out our *Recipes* page for examples and most commonly used settings

---

Next, we'll go into how the configuration is structured.

2.1 Basics

Mergeable configuration consists of an array of independent recipes where each recipe needs to have the following properties:

when: specify webhook event(s) in which to process the validation

name: a friendly name which appears on the PR for the associated check

filter: specify a series of optional filters to be checked and only runs validators if they are passing

validate: specify a series of validator to be checked

pass: specify a series of action to execute if the validation suite returned a *pass*

fail: specify a series of action to execute if the validation suite returned a *fail*

error: specify a series of action to execute if the validation suite returned a *error*

Each recipe appears as a separate check in the pull request.

Here is a full example of how a recipe looks -

```
version: 2
mergeable:
  - when: {{event}}, {{event}} # can be one or more
```

(continues on next page)

```
   name: check name A
   filter:
     # list of filters (optional). Specify one or more.
     - do: {{filter}}
       {{option}}: # name of an option supported by the validator.
         {{sub-option}}: {{value}} # an option will have one or more sub-options.
   validate:
     # list of validators. Specify one or more.
     - do: {{validator}}
       {{option}}: # name of an option supported by the validator.
         {{sub-option}}: {{value}} # an option will have one or more sub-options.
   pass: # list of actions to be executed if all validation passes. Specify one or␣
→more. Omit this tag if no actions are needed.
     - do: {{action}}
   fail: # list of actions to be executed when at least one validation fails.␣
→Specify one or more. Omit this tag if no actions are needed.
     - do: {{action}}
   error: # list of actions to be executed when at least one validator throws an␣
→error. Specify one or more. Omit this tag if no actions are needed.
     - do: {{action}}

 - when: {{event}}, {{event}} # example for second recipe
   name: check name B
   filter:
     # list of filters (optional). Specify one or more.
     - do: {{filter}}
       {{option}}: # name of an option supported by the validator.
         {{sub-option}}: {{value}} # an option will have one or more sub-options.
   validate:
     # list of validators. Specify one or more.
     - do: {{validator}}
       {{option}}: # name of an option supported by the validator.
         {{sub-option}}: {{value}} # an option will have one or more sub-options.
   pass: # list of actions to be executed if all validation passes. Specify one or␣
→more. Omit this tag if no actions are needed.
     - do: {{action}}
   fail: # list of actions to be executed when at least one validation fails.␣
→Specify one or more. Omit this tag if no actions are needed.
     - do: {{action}}
   error: # list of actions to be executed when at least one validator throws an␣
→error. Specify one or more. Omit this tag if no actions are needed.
     - do: {{action}}
```

Note: There are some default actions that'll be automatically applied based on the events specified

2.2 Filters

Filters are checks that mergeable will process in order to determine if validator will be executed.

Note: Each filter have certain events that it can support, so keep an eye out for them.

Hint: Don't see an filter that should be on here? Let us know by creating an issue on github

---

Filter List

2.2.1 Author

```
- do: author
  must_include:
      regex: 'user-1'
      message: 'Custom include message...'
  must_exclude:
      regex: 'user-2'
      message: 'Custom exclude message...'
  team: 'org/team-slug'  # verify that the author is in the team
  # all of the message sub-option is optional
```

you can use `and` and `or` options to create more complex filters

```
- do: author
  and:
    - must_exclude:
        regex: 'bot-user-1'
        message: 'Custom message...'
  or:
    - must_include:
        regex: 'user-1'
        message: 'Custom message...'
    - must_include:
        regex: 'user-2'
        message: 'Custom message...'
```

you can also nest `and` and `or` options

```
- do: author
  and:
    - or:
      - must_include:
          regex: 'user-1'
          message: 'Custom message...'
      - must_include:
          regex: 'user-2'
          message: 'Custom message...'
    - must_exclude:
        regex: 'bot-user-1'
        message: 'Custom message...'
```

Supported Events:

```
'pull_request.*', 'pull_request_review.*'
```

2.2.2 Repository

```
- do: repository
  visibility: 'public' # Can be public or private
  name:
    must_include:
      regex: 'my-repo-name'
    must_exclude:
      regex: 'other-repo-name'
  topics:
    must_include:
      regex: 'my-topic'
      message: 'Custom message...'
    must_exclude:
      regex: 'other-topic'
      message: 'Custom message...'
      # all of the message sub-option is optional
```

you can use `and` and `or` options to create more complex filters

```
- do: repository
  topics:
    and:
      - must_include:
          regex: 'topic-1'
          message: 'Custom message...'
      - must_include:
          regex: 'topic-2'
          message: 'Custom message...'
    or:
      - must_include:
          regex: 'topic-3'
          message: 'Custom message...'
      - must_include:
          regex: 'topic-4'
          message: 'Custom message...'
```

you can also nest `and` and `or` options

```
- do: repository
  topics:
    and:
      - or:
          - must_include:
              regex: 'topic-1'
              message: 'Custom message...'
          - must_include:
              regex: 'topic-2'
              message: 'Custom message...'
      - must_include:
          regex: 'topic-3'
          message: 'Custom message...'
```

Supported Events:

```
'pull_request.*', 'pull_request_review.*'
```

2.2.3 Payload

Check against any available fields within the payload, each event can have different field, please refer to **'github API documentation<https://docs.github.com/en/developers/webhooks-and-events/webhook-events-and-payloads>'_** for available fields.

An example to check if a pull_request_review event has *state* of *changes_requested*

```
- do: payload
  review:
    state:
      must_include:
        regex: 'changes_requested'
```

To check if a *pull_request* event is not a *draft*

```
- do: payload
  pull_request:
    draft:
      boolean:
        match: false
```

An example to check if a pull_request event has a *label* named *foo*

```
- do: payload
  pull_request:
    labels:
      must_include:
        regex: 'foo'
        key: 'name'
```

Each field must be checked using one of the following options

```
boolean:
  match: true/false
must_include:
  regex: 'This text must be included'
  regex_flag: 'none' # Optional. Specify the flag for Regex. default is 'i', to
→disable default use 'none'
  key: 'name' # Optional. If checking an array of objects, this specifies the key to
→check.
must_exclude:
  regex: 'Text to exclude'
  regex_flag: 'none' # Optional. Specify the flag for Regex. default is 'i', to
→disable default use 'none'
  key: 'name' # Optional. If checking an array of objects, this specifies the key to
→check.
```

Supported Events:

```
'pull_request.*', 'pull_request_review.*', issues.*'
```

2.3 Validators

Validators are checks that mergeable will process in order to determine whether an action should be executed.

Note: Each validator have certain events that it can support, so keep an eye out for them.

Hint: Don't see an validator that should be on here? Let us know by creating an issue on github

Validator List

2.3.1 Age

```
- do: age // validate based on the age of PR
  created_at:
    days: 1
    message: 'PR needs to at least 1 day old in order to merge' # optional, custom
→message to display if the validation fails
  updated_at:
    days: 1
    message: 'PR needs to be update free for 1 day before merging' # optional, custom
→message to display if the validation fails
```

Supported Events:

```
'pull_request.*', 'pull_request_review.*',
```

2.3.2 Approvals

```
- do: approvals
  min:
    count: 2 # Number of minimum reviewers. In this case 2.
    message: 'Custom message...'
  required:
    reviewers: [ user1, user2 ] # list of github usernames required to review
    owners: true # Optional boolean. When true, the file .github/CODEOWNERS is read
→and owners made required reviewers
    assignees: true # Optional boolean. When true, PR assignees are made required
→reviewers.
    requested_reviewers: true # Optional boolean. When true, all the requested
→reviewer's approval is required
    message: 'Custom message...'
  block:
    changes_requested: true # If true, block all approvals when one of the reviewers
→gave 'changes_requested' review
    message: 'Custom message...'
  limit:
    teams: ['org/team_slug'] # when the option is present, only the approvals from
→the team members will count
    users: ['user1', 'user2'] # when the option is present, approvals from users in
→this list will count
    owners: true # Optional boolean. When true, the file .github/CODEOWNER is read
→and only owners approval will count
  exclude:
    users: ['bot1', 'bot2'] # when the option is present, approvals from users in
→this list will NOT count
```

Note: in `limit` options, if more than one sub option is present, the union of the results will be used.

Note: If you receive an error for '**Resource not accessible by integration**' for Owners file, it means you haven't given mergeable read file permission

---

Note: `owners` file now support teams as well, make sure to use *@organization/team-slug* format.

Supported Events:

```
'pull_request.*', 'pull_request_review.*'
```

2.3.3 Assignee

```
- do: assignee
  max:
    count: 2 # There should not be more than 2 assignees
    message: 'test string' # this is optional
  min:
    count: 2 # min number of assignees
    message: 'test string' # this is optional
```

Supported Events:

```
'pull_request.*', 'pull_request_review.*', 'issues.*'
```

2.3.4 Author

```
- do: author
  must_include:
      regex: 'user-1'
      message: 'Custom include message...'
  must_exclude:
      regex: 'user-2'
      message: 'Custom exclude message...'
  team: 'org/team-slug'  # verify that the author is in the team
  # all of the message sub-option is optional
```

you can use `and` and `or` options to create more complex filters

```
- do: author
  and:
    - must_exclude:
        regex: 'bot-user-1'
        message: 'Custom message...'
  or:
    - must_include:
        regex: 'user-1'
        message: 'Custom message...'
```

(continues on next page)

```
    - must_include:
        regex: 'user-2'
        message: 'Custom message...'
```

you can also nest `and` and `or` options

```
- do: author
  and:
    - or:
      - must_include:
          regex: 'user-1'
          message: 'Custom message...'
      - must_include:
          regex: 'user-2'
          message: 'Custom message...'
    - must_exclude:
        regex: 'bot-user-1'
        message: 'Custom message...'
```

Supported Events:

```
'pull_request.*', 'pull_request_review.*'
```

2.3.5 BaseRef

```
- do: baseRef
  must_include:
    regex: 'master|feature-branch1'
    regex_flag: 'none' # Optional. Specify the flag for Regex. default is 'i', to
→disable default use 'none'
    message: 'Custom message...'
  must_exclude:
    regex: 'feature-branch2'
    regex_flag: 'none' # Optional. Specify the flag for Regex. default is 'i', to
→disable default use 'none'
    message: 'Custom message...'
  mediaType:  # Optional. Required by status.* events to enable the groot preview on
→some Github Enterprise servers
    previews: 'array'
```

Simple example:

```
- do: baseRef
  must_exclude:
    regex: 'master'
    message: 'Merging into repo:master is forbidden'
```

Example with groot preview enabled (for status.* events on some older Github Enterprise servers)

```
- do: baseRef
  must_include:
    regex: 'master|main'
    message: 'Auto-merging is only enabled for default branch'
  mediaType:
```

```
    previews:
      - groot
```

Supported Events:

```
'pull_request.*', 'pull_request_review.*', 'check_suite.*', status.*
```

2.3.6 Change set

```yaml
- do: changeset # validate against the files in the PR
  no_empty:
    enabled: false # Cannot be empty when true.
    message: 'Custom message...'
  must_include:
    regex: 'yarn.lock'
    message: 'Custom message...'
  must_exclude:
    regex: 'package.json'
    message: 'Custom message...'
  begins_with:
    match: 'A String' # or array of strings
    message: 'Some message...'
  ends_with:
    match: 'A String' # or array of strings
    message: 'Come message...'
  min:
      count: 2 # min number of files in a PR
      message: 'Custom message...'
  max:
    count: 2 # max number of files in a PR
    message: 'Custom message...'
  files: # status of files to be included in changeset. If no 'files' option is␣
→provided, all files are included.
    added: true # default: false. If true, added files are included.
    modified: false # default: false. If true, modified files are included.
    removed: true # default: false. If true, deleted files are included.
  # note that setting file status sub-options (added, modified, removed) to false is␣
→optional.
  # all of the message sub-option is optional
```

you can use `and` and `or` options to create more complex validations

```yaml
- do: changeset # validate against the files in the PR
  and:
    - must_include:
        regex: 'doc/.*'
        message: 'Custom message...'
    - must_include:
        regex: 'changelog.md'
        message: 'Custom message...'
  or:
    - must_include:
        regex: 'package-lock.json'
        message: 'Custom message...'
```

```
    - must_include:
        regex: 'yarn.lock'
        message: 'Custom message...'
```

you can also nest `and` and `or` options

```
- do: changeset # validate against the files in the PR
  and:
    - or:
      - must_include:
          regex: 'package-lock.json'
          message: 'Custom message...'
      - must_include:
          regex: 'package.json'
          message: 'Custom message...'
    - must_include:
        regex: 'yarn.lock'
        message: 'Custom message...'
```

Supported Events:

```
'pull_request.*', 'pull_request_review.*'
```

2.3.7 Commit

```
- do: commit
  message:
    regex: '^(feat|docs|chore|fix|refactor|test|style|perf)(\(\w+\))?:.+$'
    message: 'Custom message' # Semantic release conventions must be followed
    skip_merge: true # Optional, Default is true. Will skip commit with message that␣
→includes 'Merge'
    oldest_only: false # Optional, Default is false. Only check the regex against the␣
→oldest commit
    newest_only: false # Optional, Default is false. Only check the regex against the␣
→newest commit
    single_commit_only: false # Optional, Default is false. only process this␣
→validator if there is one commit
    message_type: '' # Optional, only check regex against the field specified.␣
→Default is '', which processes the 'message' field. Can also be set to 'author_email
→' or 'committer_email'
  jira:
    regex: '[A-Z][A-Z0-9]+-\d+'
    regex_flag: none
    message: 'The Jira ticket does not exist'
```

Supported Events:

```
'pull_request.*', 'pull_request_review.*'
```

2.3.8 Contents

```
- do: contents
  files: # determine which files contents to validate
    pr_diff: true # If true, validator will grab all the added and modified files in
↪the head of the PR
    ignore: ['.github/mergeable.yml'] # Optional, default ['.github/mergeable.yml'],
↪pattern of files to ignore
  must_include:
    regex: 'yarn.lock'
    message: 'Custom message...'
  must_exclude:
    regex: 'package.json'
    message: 'Custom message...'
  begins_with:
    match: 'A String' # or array of strings
    message: 'Some message...'
  ends_with:
    match: 'A String' # or array of strings
    message: 'Come message...'
```

Supported Events:

```
'pull_request.*', 'pull_request_review.*'
```

2.3.9 Dependent

Validates that the files specified are all part of a pull request (added or modified).

```
- do: dependent
  files: ['package.json', 'yarn.lock'] # list of files that are dependent on one
↪another and must all be part of the changes in a PR.
  message: 'Custom message...' # this is optional, a default message is used when not
↪specified.
```

Alternatively, to validate dependent files only when a specific file is part of the pull request, use the changed option:

```
- do: dependent
    changed:
      file: package.json
      files: ['package-lock.json', 'yarn.lock']
    message: 'Custom message...' # this is optional, a default message is used when
↪not specified.
```

The above will validate that both the files package-lock.json and yarn.lock is part of the modified or added files if and only if package.json is part of the PR.

Supported Events:

```
'pull_request.*', 'pull_request_review.*'
```

2.3.10 Description

```
- do: description
  no_empty:
    enabled: false # Cannot be empty when true.
```

(continues on next page)

```
    message: 'Custom message...' # this is optional, a default message is used when␣
↪not specified.
  must_include:
    regex: '### Goals|### Changes'
    regex_flag: 'none' # Optional. Specify the flag for Regex. default is 'i', to␣
↪disable default use 'none'
    message: >
     Please describe the goals (why) and changes (what) of the PR.
    # message is is optional, a default message is used when not specified.
  must_exclude:
    regex: 'DO NOT MERGE'
    regex_flag: 'none' # Optional. Specify the flag for Regex. default is 'i', to␣
↪disable default use 'none'
    message: 'Custom message...' # optional
  begins_with:
    match: '### Goals' # or array of strings
    message: 'Some message...' #optional
  ends_with:
    match: 'Any last sentence' # array of strings
    message: 'Come message...' # optional
  jira:
    regex: '[A-Z][A-Z0-9]+-\d+'
    regex_flag: none
    message: 'The Jira ticket does not exist'
```

you can use `and` and `or` options to create more complex validations

```
- do: description
  and:
    - must_include:
        regex: '### Goals'
        message: 'Custom message...'
    - must_include:
        regex: '### Changes'
        message: 'Custom message...'
  or:
    - must_include:
        regex: '### Bug Description'
        message: 'Custom message...'
    - must_include:
        regex: '### Feature Description'
        message: 'Custom message...'
```

you can also nest `and` and `or` options

```
- do: description
  and:
    - or:
      - must_include:
          regex: '### Bug Description'
          message: 'Custom message...'
      - must_include:
          regex: '### Feature Description'
          message: 'Custom message...'
    - must_include:
        regex: '### Changes'
        message: 'Custom message...'
```

Supported Events:

```
'pull_request.*', 'pull_request_review.*', 'issues.*'
```

2.3.11 HeadRef

```
- do: headRef
  must_include:
     regex: 'feature-branch1'
     regex_flag: 'none' # Optional. Specify the flag for Regex. default is 'i', to
→disable default use 'none'
     message: 'Custom message...'
  must_exclude:
     regex: 'feature-branch2'
     regex_flag: 'none' # Optional. Specify the flag for Regex. default is 'i', to
→disable default use 'none'
     message: 'Custom message...'
  jira:
     regex: '[A-Z][A-Z0-9]+-\d+'
     regex_flag: none
     message: 'The Jira ticket does not exist'
```

Simple example:

```
- do: headRef
  must_include:
     regex: '^(feature|hotfix)\/.+$'
     message: |
        Your pull request doesn't adhere to the branch naming convention described <a
→href="some link">there</a>!k
```

Supported Events:

```
'pull_request.*', 'pull_request_review.*'
```

2.3.12 Label

```
- do: label
  no_empty:
     enabled: false # Cannot be empty when true.
     message: 'Custom message...'
  must_include:
     regex: 'type|chore|wont'
     regex_flag: 'none' # Optional. Specify the flag for Regex. default is 'i', to
→disable default use 'none'
     message: 'Custom message...'
  must_exclude:
     regex: 'DO NOT MERGE'
     regex_flag: 'none' # Optional. Specify the flag for Regex. default is 'i', to
→disable default use 'none'
     message: 'Custom message...'
  begins_with:
     match: 'A String' # or array of strings
     message: 'Some message...'
```

(continues on next page)

```
ends_with:
    match: 'A String' # or array of strings
    message: 'Come message...'
jira:
  regex: '[A-Z][A-Z0-9]+-\d+'
  regex_flag: none
  message: 'The Jira ticket does not exist'
  # all of the message sub-option is optional
```

```
- do: label
  and:
    - must_include:
        regex: 'big|medium|small'
        message: 'Custom message...'
    - must_include:
        regex: 'type|chore|wont'
        message: 'Custom message...'
  or:
    - must_include:
        regex: 'Ready to merge'
        message: 'Custom message...'
    - must_include:
        regex: 'DO NOT MERGE'
        message: 'Custom message...'
```

you can also nest `and` and `or` options

```
- do: label
  and:
    - or:
      - must_include:
          regex: 'feat|fix|chore'
          message: 'Custom message...'
      - must_include:
          regex: 'major|minor|patch'
          message: 'Custom message...'
    - must_include:
        regex: 'Ready to merge'
        message: 'Custom message...'
```

Supported Events:

```
'pull_request.*', 'pull_request_review.*', 'issues.*'
```

2.3.13 Milestone

```
- do: milestone
  no_empty:
    enabled: true # Cannot be empty when true.
    message: 'Custom message...'
  must_include:
    regex: 'type|chore|wont'
    regex_flag: 'none' # Optional. Specify the flag for Regex. default is 'i', to
→disable default use 'none'
```

```
    message: 'Custom message...'
  must_exclude:
    regex: 'DO NOT MERGE'
    regex_flag: 'none' # Optional. Specify the flag for Regex. default is 'i', to␣
→disable default use 'none'
    message: 'Custom message...'
  begins_with:
    match: 'A String' # array of strings
    message: 'Some message...'
  ends_with:
    match: 'A String' # array list of strings
    message: 'Come message...'
  jira:
    regex: '[A-Z][A-Z0-9]+-\d+'
    regex_flag: none
    message: 'The Jira ticket does not exist'
    # all of the message sub-option is optional
```

```
- do: milestone
  and:
    - must_include:
        regex: 'V1'
        message: 'Custom message...'
    - must_include:
        regex: 'October'
        message: 'Custom message...'
  or:
    - must_include:
        regex: 'V2'
        message: 'Custom message...'
    - must_include:
        regex: 'Non breaking Changes'
        message: 'Custom message...'
```

you can also nest `and` and `or` options

```
- do: milestone
  and:
    - or:
      - must_include:
          regex: 'V1'
          message: 'Custom message...'
      - must_include:
          regex: 'September'
          message: 'Custom message...'
    - must_include:
        regex: 'V2'
        message: 'Custom message...'
```

Note: When a closing keyword is used in the description of a pull request. The annotated issue will be validated against the conditions as well.

Supported Events:

```
'pull_request.*', 'pull_request_review.*', 'issues.*'
```

2.3.14 Project

```
- do: project
  must_include:
    regex: 'type|chore|wont'
    message: 'Custom message...'
```

Note: When a closing keyword is used in the description of a pull request. The annotated issue will be validated against the conditions as well.

Supported Events:

```
'pull_request.*', 'pull_request_review.*', 'issues.*'
```

2.3.15 Size

`size` validates that the size of changes in the pull request conform to a specified limit. We can pass in three options: `total`, `additions` or `deletions`. Each of this take in a count and message. Validates that the files specified are all part of a pull request (added or modified).

```
- do: size
  lines:
    total:
      count: 500
      message: Change is very large. Should be under 500 lines of additions and
↪deletions.
    additions:
      count: 250
      message: Change is very large. Should be under 250 lines of additions
    deletions:
      count: 500
      message: Change is very large. Should be under 250 lines of deletions.
    ignore_comments: false #if true, comments will not be counted toward the lines
↪count
```

`max` is an alias for total, so the below configuration is still valid.

Warning: currently `ignore_comments` feat only works with '.js' and '.py' extensions, if you have request for more file extensions please create a ticket here

---

```
- do: size
  lines:
    max:
      count: 500
      message: Change is very large. Should be under 500 lines of additions and
↪deletions.
```

It also supports an `ignore` or `match` setting to allow excluding or including certain files from the total size (e.g. for ignoring automatically generated files that increase the size a lot).

This option supports glob patterns, so you can provide either the path to a specific file or ignore whole patterns:

```
- do: size
  match: ['src']
  ignore: ['package-lock.json', 'src/tests/__snapshots__/**', 'docs/*.md']
  lines:
    total:
      count: 500
      message: Change is very large. Should be under 500 lines of additions and
↪deletions
```

In case both `match` and `ignore` are specified, the `match` filters are applied first.

Note that the glob functionality is powered by the minimatch library. Please see their documentation for details on how glob patterns are handled and possible discrepancies with glob handling in other tools.

The size validator currently excludes from the size count any files that were completely deleted in the PR.

Supported Events:

```
'pull_request.*', 'pull_request_review.*'
```

2.3.16 Stale

```
- do: stale
  days: 20 # number of days ago.
  type: pull_request, issues # what items to search for.
  ignore_drafts: true # if set to true, the stale check will ignore draft items
  ignore_milestones: true # if set to true, the stale check will ignore items that
↪have an associated milestone
  ignore_projects: true # if set to true, the stale check will ignore items that have
↪an associated project
  label: # optional property to filter the items that are actioned upon
    match: ['label1_to_match', 'label2_to_match'] # only items with matching labels
↪will be actioned upon and marked as stale
    ignore: ['label1_to_ignore', 'label2_to_ignore'] # items with these labels will
↪be ignored and not marked as stale
  time_constraint: # Optional, run the validator only if it in within the time
↪constraint
    time_zone: 'America/Los_Angeles' # Optional, UTC time by default, for valid
↪timezones see `here <https://momentjs.com/timezone/>`_
    hours_between: ['9', '17'] # Optional, 24 hours by default, run only if [0] >=
↪Hour Now <= [1]
    days_of_week: ['Mon', 'Tue', 'Wed', 'Thu', 'Fri'] # Optional, 7 days a week by
↪default, specific the days of the week in which to run the validator
```

Note: This is a special use case. The schedule event runs on an interval. When used with stale, it will search for issues and/or pull request that are n days old. See a full example »

Supported Events:

```
'schedule.repository'
```

2.3.17 Title

```
- do: title
  no_empty:
      enabled: true # Cannot be empty when true. A bit redundant in this case since␣
→GitHub don't really allow it. :-)
      message: 'Custom message...'
  must_include:
      regex: 'doc|feat|fix|chore'
      regex_flag: 'none' # Optional. Specify the flag for Regex. default is 'i', to␣
→disable default use 'none'
      message: 'Custom message...'
  must_exclude:
      regex: 'DO NOT MERGE|WIP'
      regex_flag: 'none' # Optional. Specify the flag for Regex. default is 'i', to␣
→disable default use 'none'
      message: 'Custom message...'
  begins_with:
      match: ['doc','feat','fix','chore']
      message: 'Some message...'
  ends_with:
      match: 'A String' # or array of strings
      message: 'Come message...'
      # all of the message sub-option is optional
  jira:
    regex: '[A-Z][A-Z0-9]+-\d+'
    regex_flag: none
    message: 'The Jira ticket does not exist'
```

```
- do: title
  and:
    - must_include:
        regex: 'ISSUE-\d+'
        message: 'Custom message...'
    - must_include:
        regex: 'type:.+'
        message: 'Custom message...'
  or:
    - must_include:
        regex: 'feat|chore|fix'
        message: 'Custom message...'
    - must_include:
        regex: 'major|minor|patch'
        message: 'Custom message...'
```

you can also nest `and` and `or` options

```
- do: title
  and:
    - or:
      - must_include:
          regex: 'feat|fix|chore'
          message: 'Custom message...'
      - must_include:
          regex: 'major|minor|patch'
          message: 'Custom message...'
    - must_include:
```

(continues on next page)

(continued from previous page)

```
        regex: 'ISSUE-\d+'
        message: 'Custom message...'
```

Supported Events:

```
'pull_request.*', 'pull_request_review.*', 'issues.*'
```

2.4 Options

These Options are used with validators.

Options List

2.4.1 Boolean

`boolean` can be used to validate if the input is exactly *true* or *false*. This does not pass truthy values.

For example, if the *pull_request.draft* value is *false*:

```
{
  "action": "opened",
  "number": 2,
  "pull_request": {
    "draft": false
```

This will pass validation, because the value of *draft* exactly matches *false*:

```
- do: payload
  pull_request:
    draft:
      boolean:
        match: false
        message: 'Custom message...' # this is optional, a default message is used␣
↪when not specified.
```

Table 1: Supported Params

Param	Description	Required	Default Message
match	Bool value to check for	Yes	
message	Message to show if the validation fails	No	The [INPUT NAME] must be [match]

Supported Filters:

```
'payload'
```

2.4.2 BeginsWith

`begins_with` can be used to validate inputs if begin with given string

```
- do: milestone
  begins_with:
     match: 'A String' # array of strings
     message: 'Some message...'
```

Table 2: Supported Params

Param	Description	Required	Default Message
match	message to validate input with	Yes	
message	Message to show if the validation fails	No	[INPUT NAME] must begins with [MATCH]

Supported Validators:

```
'changeset', 'content', 'description', 'label', 'milestone', 'title'
```

2.4.3 EndsWith

`ends_with` can be used to validate inputs if end with given string

```
- do: milestone
  ends_with:
     match: 'A String' # array of strings
     message: 'Some message...'
```

Table 3: Supported Params

Param	Description	Required	Default Message
match	message to validate input with	Yes	
message	Message to show if the validation fails	No	[INPUT NAME] must ends with [MATCH]

Supported Validators:

```
'changeset', 'content', 'description', 'label', 'milestone', 'title'
```

2.4.4 Jira

`jira` can be used to validate inputs string in Atlassian Jira with APIs

---

Note: This option work in self-hosted version only.

Table 4: Required Environment Variables

Env	Description	Required	Default
JIRA_PASSWORD	Password to authenticate with JIRA	Yes	
JIRA_USERNAME	Username to authenticate with JIRA	Yes	
JIRA_HOST	Host to authenticate with JIRA	yes	
JIRA_PROTOCOL	Protocol to establish connection with JIRA	no	https
JIRA_VERSION	JIRA API Version to use	no	2
JIRA_STRICT_SSL	Force SSL while establishing connection	no	yes

```
- do: headRef
  jira:
    regex: '[A-Z][A-Z0-9]+-\d+'
    regex_flag: none
    message: 'The Jira ticket does not valid'
```

Table 5: Supported Params

Param	Description	Required	Default Message
regex	Regex enabled message to validate input with	Yes	
message	Message to show if the validation fails	No	[INPUT NAME] does not include [REGEX]
regex_flag	Regex flag to be used with regex param to validate inputs	No	i

Supported Validators:

```
'commit', 'description', 'headRef', 'label', 'milestone', 'title'
```

2.4.5 Max

`max` can be used to validate inputs length that is no more than given integer.

```
- do: assignee
  max:
    count: 2 # There should not be more than 2 assignees
    message: 'test string' # this is optional
```

Table 6: Supported Params

Param	Description	Required	Default Message
count	number to validate input's length	Yes	
message	Message to show if the validation fails	No	[INPUT NAME] count is more than [COUNT]

Supported Validators:

```
'approvals', 'assignee', 'changeset', 'label'
```

2.4.6 Min

`min` can be used to validate inputs length that isn't less than given integer.

```
- do: assignee
  min:
    count: 2 # There should be more than 2 assignees
    message: 'test string' # this is optional
```

Table 7: Supported Params

Param	Description	Required	Default Message
count	number to validate input's length	Yes	
message	Message to show if the validation fails	No	[INPUT NAME] count is less than [COUNT]

Supported Validators:

```
'approvals', 'assignee', 'changeset', 'label', 'size'
```

2.4.7 MustInclude

`must_include` can be used to validate input that includes the given regex supported message.

```
- do: headRef
  must_include:
    regex: '^(feature|hotfix)\/.+$'
    message: |
        Your pull request doesn't adhere to the branch naming convention described <a
→href="some link">there</a>!k
```

You can also use an array of regex matchers. If any of them match, the validation will pass.

```
- do: headRef
  must_include:
    regex:
      - "^feature"
      - "^hotfix"
      - "^fix"
    message: |
        Your pull request doesn't adhere to the branch naming convention described <a
→href="some link">there</a>!k
```

Table 8: Supported Params

Param	Description	Required	Default Message
regex	Regex or array enabled message to validate input with	Yes	
message	Message to show if the validation fails	No	[INPUT NAME] does not include [REGEX]
regex_flag	Regex flag to be used with regex param to validate inputs	No	i

Supported Validators:-

```
'baseRef', 'headRef', 'changeset', 'commit', 'content', 'description', 'label',
↪'milestone', 'project', 'title'
```

2.4.8 MustExclude

`must_exclude` can be used to validate input that excludes the given regex supported message.

```
- do: headRef
  must_exclude:
    regex: '^(feature|hotfix)\/.+$'
    message: |
        Your pull request doesn't adhere to the branch naming convention described <a␣
↪href="some link">there</a>!k
```

You can also use an array of regex matchers. If any of them match, the validation will fail.

```
- do: headRef
  must_exclude:
    regex:
      - "^bug"
      - "^breaking"
      - "^test"
    message: |
        Your pull request doesn't adhere to the branch naming convention described <a␣
↪href="some link">there</a>!k
```

Table 9: Supported Params

Param	Description	Required	Default Message
regex	Regex or array enabled message to validate input with	Yes	
message	Message to show if the validation fails	No	[INPUT NAME] does not include [REGEX]
regex_flag	Regex flag to be used with regex param to validate inputs	No	i

Supported Validators:-

```
'baseRef', 'headRef', 'changeset', 'content', 'description', 'label', 'milestone',
↪'title'
```

2.4.9 NoEmpty

`no_empty` can be used to validate if input is not empty

```
- do: description
  no_empty:
    enabled: false # Cannot be empty when true.
    message: 'Custom message...' # this is optional, a default message is used when␣
↪not specified.
```

Table 10: Supported Params

Param	Description	Required	Default Message
enabled	Bool value to enable/disable the option	Yes	
message	Message to show if the validation fails	No	The [INPUT NAME] can't be empty

Supported Validators:

```
'changeset', 'description', 'label', 'milestone', 'title'
```

2.4.10 Required

`required` can be used to validate if input meets the conditions given with params

```
- do: approvals
  required:
    reviewers: [ user1, user2 ] # list of github usernames required to review
    owners: true # Optional boolean. When true, the file .github/CODEOWNERS is read
↪and owners made required reviewers
    assignees: true # Optional boolean. When true, PR assignees are made required
↪reviewers.
    requested_reviewers: true # Optional boolean. When true, all the requested
↪reviewer's approval is required
    message: 'Custom message...'
```

Table 11: Supported Params

Param	Description	Required	Default Message
reviewers	An array value for github users/teams to be required to do the validation	No	[]
owners	The file .github/CODEOWNERS is read and owners made required reviewers	No	[]
assignees	PR assignees are made required reviewers.	No	[]
requested_reviewers	All the requested reviewer's approval is required	No	[]
message	Message to show if the validation fails	No	[INPUT NAME] does not include [REGEX]

Supported Validators:

```
'approvals'
```

2.5 Operators

These operators can help create more complex logic of validations

Operator List

2.5.1 And

`And`, `Or`, and `Not` can be used to create more complex validation/filter check

```
filter:
  - do: and
    filter:
      - do: author
        must_include: 'user-1'
      - do: repository
        visibility: public
validate:
  - do: and
    validate:
      - do: title
        begins_with: '[WIP]'
      - do: label
        must_include: 'Ready to Merge'
```

you can also create nested `And`, `Or`, and `Not`

```
filter:
  - do: and
    filter:
      - do: or
        filter:
          - do: author
            must_include: 'user-1'
          - do: author
            must_include: 'user-2'
      - do: repository
        visibility: public
validate:
  - do: and
    validate:
      - do: or
        validate:
          - do: title
            begins_with: 'feat:'
          - do: label
            must_include: 'feature'
      - do: label
        must_include: 'Ready to Merge'
```

2.5.2 Or

`And`, `Or`, and `Not` can be used to create more complex validation/filter check

```
filter:
  - do: or
    filter:
      - do: author
        must_include: 'user-1'
      - do: repository
        visibility: public
validate:
```

```yaml
    - do: or
      validate:
        - do: title
          begins_with: '[WIP]'
        - do: label
          must_include: 'Ready to Merge'
```

you can also create nested `And`, `Or`, and `Not`

```yaml
filter:
  - do: and
    filter:
      - do: or
        filter:
          - do: author
            must_include: 'user-1'
          - do: author
            must_include: 'user-2'
      - do: repository
        visibility: public
validate:
  - do: and
    validate:
      - do: or
        validate:
          - do: title
            begins_with: '[WIP]'
          - do: label
            must_include: '[WIP]'
      - do: label
        must_include: 'DO NOT MERGE'
```

2.5.3 Not

`And`, `Or`, and `Not` can be used to create more complex validation/filter check

```yaml
filter:
  - do: not
    filter:
      - do: author
        must_include: 'user-1'
      - do: repository
        visibility: public
validate:
  - do: not
    validate:
      - do: title
        begins_with: '[WIP]'
      - do: label
        must_include: 'Ready to Merge'
```

you can also create nested `And`, `Or`, and `Not`

```yaml
filter:
  - do: not
```

```
    filter:
      - do: or
        filter:
          - do: author
            must_include: 'user-1'
          - do: author
            must_include: 'user-2'
validate:
  - do: and
    validate:
      - do: not
        validate:
          - do: title
            begins_with: 'feat:'
          - do: label
            must_include: 'feature'
      - do: label
        must_include: 'Ready to Merge'
```

2.6 Actions

Actions that mergeable is currently able to perform.

Hint: Don't see an action that should be on here? Let us know by creating an issue on github

---

2.6.1 Assign

```
- do: assign
  assignees: [ 'shine2lay', 'jusx', '@author' ] # only array accepted, use @author
→for PR/Issue author
```

Supported Events:

```
'pull_request.*', 'issues.*'
```

2.6.2 Check

Note: The logic for whether checks will be added by default is as follows: 1. If no action is provided in either pass, fail or error , add *checks* as default (to be backward compatible) 2. If only actions other than *checks* is provided, don't add check as default (to support cases where checks are not wanted) 3. If checks is a part of the actions provided, all pass , fail and error cases will have *checks* (to prevent case where a check is hanged and never finish processing)

```
- do: checks # default pass case
  status: 'success' # Can be: success, failure, neutral, cancelled, timed_out, or
→action_required
  payload:
```

```
    title: 'Mergeable Run have been Completed!'
    summary: "All the validators have returned 'pass'! \n Here are some stats of the␣
→run: \n {{validationCount}} validations were ran"
```

You can pass in Handlebars template to show the details result of the run.

```
- do: checks # default fail case
  status: 'failure' # Can be: success, failure, neutral, cancelled, timed_out, or␣
→action_required
  payload:
    title: 'Mergeable Run have been Completed!'
    summary: |
        ### Status: {{toUpperCase validationStatus}}
            Here are some stats of the run:
            {{validationCount}} validations were ran.
            {{passCount}} PASSED
            {{failCount}} FAILED
    text: "{{#each validationSuites}}\n
        #### {{{statusIcon status}}} Validator: {{toUpperCase name}}\n
        {{#each validations }} * {{{statusIcon status}}} ***{{ description }}}***\n
            Input : {{{details.input}}}\n
            Settings : {{{displaySettings details.settings}}}\n
            {{/each}}\n
        {{/each}}"
```

```
- do: checks # default error case
  status: 'action_required' # Can be: success, failure, neutral, cancelled, timed_out,
→ or action_required
  payload:
    title: 'Mergeable found some errors!'
    summary: |
        ### Status: {{toUpperCase validationStatus}}
        Some or All of the validators have returned 'error' status, please check␣
→below for details
        Here are some stats of the run: \n {{validationCount}} validations were ran.
        {{passCount}} ***PASSED***
        {{failCount}} ***FAILED***
        {{errorCount}} ***ERRORED***
    text: "{{#each validationSuites}}
        #### {{{statusIcon status}}} Validator: {{toUpperCase name}}
        Status {{toUpperCase status}}
        {{#each validations }} * {{{statusIcon status}}} ***{{ description }}}***
            Input : {{{details.input}}}
            Settings : {{{displaySettings details.settings}}}
            {{#if details.error}}
            Error : {{{details.error}}}
            {{/if}}
            {{/each}}
        {{/each}}"
```

Note: if any of required fields `title`, `summary` or `status` is missing, default values will be used

Note: checks will automatically re-run if the base branch has a modified config file

Supported Events:

The *pull_request.closed* event is not supported since it does not have meaningful use in the context of GitHub check API.

```
'pull_request.assigned', 'pull_request.auto_merge_disabled', 'pull_request.auto_merge_
↪enabled', 'pull_request.converted_to_draft', 'pull_request.demilestoned', 'pull_
↪request.dequeued', 'pull_request.edited', 'pull_request.enqueued', 'pull_request.
↪labeled', 'pull_request.locked', 'pull_request.milestoned', 'pull_request.opened',
↪'pull_request.push_synchronize', 'pull_request.ready_for_review', 'pull_request.
↪reopened', 'pull_request.review_request_removed', 'pull_request.review_requested',
↪'pull_request.synchronize', 'pull_request.unassigned', 'pull_request.unlabeled',
↪'pull_request.unlocked', 'pull_request_review.dismissed', 'pull_request_review.
↪edited', 'pull_request_review.submitted'
```

2.6.3 Close

```
- do: close
```

Supported Events:

```
'schedule.repository', 'pull_request.*', 'issues.*'
```

2.6.4 Comment

```
- do: comment
  payload:
    body: >
      Your very long comment can go here.
  leave_old_comment: true # Optional, by default old comments are deleted, if true,
↪old comments will be left alone
```

Supported Events:

```
'schedule.repository', 'pull_request.*', 'issues.*'
```

2.6.5 Merge

```
- do: merge
  merge_method: 'merge' # Optional, default is 'merge'. Other options : 'rebase',
↪'squash'
  # template variables for next two items come from result of https://docs.github.com/
↪en/rest/reference/pulls#get-a-pull-request
  # use triple curly braces to avoid html escaping
  commit_title: '{{{ title }}} (#{{{ number }}})' # Optional, override commit title
  commit_message: '{{{ body }}}' # Optional, override commit message
```

Supported Events:

```
'pull_request.*', 'pull_request_review.*', 'status.*', 'check_suite.*'
```

2.6.6 Labels

You can add, remove, and delete labels with one action.

> **Warning:** Using multiple `do:    labels` in the same operation is not idempotent. Use only one `labels` action per block for accurate labeling.

You can add new labels, preserving existing ones

```
- do: labels
  add: 'Ready for Review'
```

You can delete existing labels

```
- do: labels
  delete: [ 'Ready for Review', 'Triage' ]
```

You can replace all current labels with new ones

```
- do: labels
  replace: [ 'Triage', 'Needs Deploy' ]
```

You can also use any combination of these options. They can be listed in any order, but the action is always evaluated in the order of `replace → add → delete`.

```
- do: labels
  replace: [ 'New Task', 'Not Useful' ]
  add: [ 'Work in Progress', 'Needs Deploy' ]
  delete: 'Not Useful'

# result: [ 'New Task', 'Work in Progress', 'Needs Deploy' ]
```

`labels` and `mode`

> **Warning:** Using `labels` with `mode` is deprecated and will be removed in v3. Use the `add`, `replace`, and `delete` options above for labeling.

You can also add a set of the labels on an item

```
- do: labels
  # if label doesn't exist, it'll be created
  labels: [ 'Triage' ] # Only arrays are accepted
  mode: 'add' # Optional , default is 'add'. Other options : 'replace', 'delete'
```

You can also replace all of the labels on an item with a given set of labels

```
- do: labels
  # if label doesn't exist, it'll be created
  labels: [ 'Triage' ] # Only arrays are accepted
  mode: 'replace' # Replaces all of the labels with the above array of labels
```

You can also delete existing labels on an item and specify glob patterns when the mode is *delete*.

```
- do: labels
  # if label doesn't exist, it'll be created
  labels: [ 'feature-*' ] # All labels beginning with 'feature-' will be removed
  mode: 'delete'
```

Note that the glob functionality is powered by the minimatch library. Please see their documentation for details on how glob patterns are handled and possible discrepancies with glob handling in other tools.

Supported Events:

```
'schedule.repository', 'pull_request.*', 'issues.*'
```

2.6.7 Request Review

You can request specific reviews from specific reviewers, teams, or both

```
- do: request_review
  reviewers: ['name1', 'name2']
  teams: ['developers'] # team names without organization
```

Supported Events:

```
'pull_request.*'
```

2.7 Reusable Configuration

YML has a feature called **'Anchor<https://blog.daemonl.com/2016/02/yaml.html>'_** that allows you to create reusable parts in the config

2.8 Organisation-wide defaults

You can specify a default configuration to be applied across your GitHub organisation. This can help reduce how many configuration files you need to maintain and make it easier to get started with Mergeable.

To add a default configuration:

1. Create a repository called `.github` in your organisation.

2. Create a file with the path `.github/mergeable.yml` in this repository.

The final path of the file (including the repo name) should be `<YOUR_ORG>/.github/.github/mergeable.yml`

Mergeable will now use this file as the default when it cannot find one in a given repository or PR. It determines the file to use in the following order:

1. A `mergeable.yml` inside the PR if the PR originates from the same repository. If it originates from a fork it is not loaded for security reasons. See #406 for more details.

2. A `mergeable.yml` inside the repository the PR is for.

3. A `mergeable.yml` at `<YOUR_ORG>/.github/.github/mergeable.yml`.

Note: If config file is changed in base branch, all the PR against the base branch will be re ran using the changed config file Warning! this feature require mergeable have `read` access to contents and needs to be listening for `push` event

2.8.1 Why the weird default file path?

The Probots library that Mergeable uses automatically searches for config files in a repo named `.github` within the organisation.

The double nesting of the `<YOUR_ORG>/.github/.github/mergeable.yml` default file is unfortunately necessary. The GitHub app permissions model only lets you specify a single path for your probot to access, so it must be the same as in regular repositories.

Recipes

3.1 Work In Progress

Prevent accidental merging of Pull Requests that are work in progress by labeling it WIP or prefixing the title with the abbreviation.

```
version: 2
mergeable:
  - when: pull_request.*
    validate:
      - do: title
        must_exclude:
          regex: ^\[WIP\]
      - do: label
        must_exclude:
          regex: 'wip'
```

3.2 No Empty Description

Ensure all Pull Requests have a description so that reviewers have context.

```
version: 2
mergeable:
  - when: pull_request.*
    validate:
      - do: description
        no_empty:
          enabled: true
          message: Description matter and should not be empty. Provide detail with
↪**what** was changed, **why** it was changed, and **how** it was changed.
```

3.3 Dependent Files

Certain files are related and you want to ensure that they are updated as part of the PR (i.e. if package.json is updated, so should yarn.lock and package-lock.json)

```
version: 2
mergeable:
  - when: pull_request.*
    validate:
      - do: dependent
        changed:
          file: 'package.json'  # also supports globs expressions
          required: ['package-lock.json', 'yarn.lock'] # alias: `files` for backward
→compatibility
```

3.4 Must Include Milestone

Ensure that all Pull Requests have a milestone associated. Mergeable will also detect when you are **'closing an issue<https://help.github.com/en/github/managing-your-work-on-github/linking-a-pull-request-to-an-issue>'_** that is associated with the specified milestone.

```
version: 2
mergeable:
  - when: pull_request.*
    validate:
      - do: milestone
        must_include:
          regex: Release 1
```

3.5 Small PR Size

Size: Ensure that PRs don't exceed a certain size in terms of lines changed (excluding file patterns specified with `ignore`).

```
version: 2
mergeable:
  - when: pull_request.*
    validate:
      - do: size
        ignore: ['ignore_me.js', 'ignore_this_directory/*', '**/ignore_this_prefix*.js
→']
        lines:
          max:
            count: 500
            message: Change is very large. Should be under 500 lines of addtions and
→deletions.
```

3.6 Comment If Guidelines not met

Automatically create a comment when a new issue is openened to remind the author when the title does not follow
conventions or is missing a label.

```
version: 2
mergeable:
  - when: issues.opened
    validate:
      - do: title
        begins_with:
          match: ['AUTH', 'SOCIAL', 'CORE']
      - do: label
        must_include:
          regex: bug|enhancement
    fail:
      - do: comment
        payload:
          body: >
            The following problems were found with this issue:
              - Title must begin with `AUTH`, `SOCIAL` or `CORE`
              - The issue should either be labeled `bug` or `enhancement`
```

3.7 Check Stale PR and Issues

Detect issues and pull requests that are n days old (stale) and notify authors and collaborators by creating a comment.

```
version: 2
mergeable:
  - when: schedule.repository
    validate:
      - do: stale
        days: 20
        type: pull_request, issues
    pass:
      - do: comment
        payload:
          body: This is old. Is it still relevant?
```

3.8 Greet a new contributor

Add a comment on a pull request when it is created

```
version: 2
mergeable:
  - when: pull_request.opened
    name: "Greet a contributor"
    validate: []
    pass:
      - do: comment
        payload:
```

```
        body: >
            Thanks for creating a pull request! A maintainer will review your changes␣
→shortly. Please don't be discouraged if it takes a while.
```

3.9 Auto-merge pull requests once all checks pass

This recipe relies on the fact that the main branch has been protected and only allows merges when the required checks have passed or the required number of reviews/other conditions are met. This basically means that `mergeable` will merge the pull request as soon as it shows a green merged button on Github.

Notice the blank validator which ensures that the merge event happens as soon as Github allows `mergeable` to merge the pull request.

```
version: 2
mergeable:
  - when: pull_request.*, pull_request_review.*, status.*, check_suite.*
    name: "Automatically merge pull requests once it passes all checks"
    validate: []
    pass:
      - do: merge
        merge_method: "squash"
```

3.10 Approval check + title check if certain files are changed

Add 2 checks to the PR 1. Approval check - Checks whether the PR has been approved by certain people 2. Title should match a regex if certain files are changed. If no changes are made in those files, check should pass

```
version: 2
mergeable:
  - when: pull_request.*, pull_request_review.*
    name: 'Approval check'
    validate:
      - do: approvals
        min:
          count: 1
        limit:
          users: [ 'approverA', 'approverB' ]

  - when: pull_request.*, pull_request_review.*
    name: 'PR title check'
    validate:
      - do: or
        validate:
          - do: changeset
            must_exclude:
              regex: 'some/regex/for/those/certain/files/*'
          - do: and
            validate:
              - do: changeset
                must_include:
                  regex: 'some/regex/for/those/certain/files/*'
```

```
                    - do: title
                      begins_with:
                        match: [ 'some prefix' ]
```

3.11 Only run rules if PR is not a draft

Checks that the PR's draft state is false before running actions.

```
version: 2
mergeable:
  - when: pull_request.*, pull_request_review.*
    name: 'Draft check'
    validate:
      - do: payload
        pull_request:
          draft:
            boolean:
              match: false
    pass:
      - do: comment
        payload:
          body: This PR is NOT a draft!
    fail:
      - do: comment
        payload:
          body: This PR is STILL a draft!
```

3.12 Allow commits only if they contain a Issue ID (like an Azure DevOps Work Item)

Checks that the PR's draft state is false before running actions.

```
version: 2
mergeable:
  - when: pull_request.*
    validate:
      - do: commit
        message:
            regex: '^(AB#[0-9]{1,})' #check if all commit messages begin with an AzDO␣
→Work Item
    pass:
      - do: comment
        payload:
          body: >
            <h2>Successfully checked for Azure Work Item IDs in commits</h2>
            <h3>All commits in your PR have Azure Board Work Item IDs. Ready for␣
→Review!</h3>
            :+1:
      - do: labels
        add: 'Ready for Review'
    fail:
```

```
      - do: comment
        payload:
          body: >
            :warning:
            <h2>Azure Boards Work Item IDs missing in commits</h2>
            <h3>Some commits messages were found not having the Azure Boards Work
→Item ID (AB#1234).</h3>
            <h3>We will close this PR for now.</h3>
            <h3>To resolve, please do one of the following</h3>
            <ul>
              <li>Identify your Azure Boards Work Item ID and <a href="https://gist.
→github.com/nepsilon/156387acf9e1e72d48fa35c4fabef0b4">amend your commits</a>. Then
→re-open the PR</li>
              <li>In case you do not have a Work Item ID to reference, please discuss
→with your reviewer(s) for alternate options</li>
            </ul>
      - do: labels
        add: 'Non-Compliant'
      - do: close
```

Contributing

We need your help:

- Have an **idea** for a new **feature**? Please create a new issue and tell us!

- Fix a **bug**, implement a new **validator** or **action** and open a pull request!

Note: For development and testing, you'll want to read deploying.

Support

Found a bug? Have a question? Or just want to chat?

- Find us on Gitter.
- Create an Issue.

CHANGELOG

May 12, 2023: fix: Loading teams for team option of author filter/validator #713

May 11, 2023: fix: Send correct payload for changing labels #715

April 25, 2023: feat: Add author validator #710

March 13, 2023: fix: Replace delete with remove in changeset validator #705

March 7, 2023: fix: Extend checks supported events #700

March 1, 2023: feat:Added user information for commit validator #682

February 3, 2023: feat: Add team option to author filter #696

February 3, 2023: chore: Update node version for release workflow #699

February 3, 2023: feat: Add Not operator #695

September 28, 2022: feat: Add last comment validator #668

August 26, 2022: set fallback for *no_empty* options processor #657

August 25, 2022: fix: set fallback value for *description* payload #655

August 20, 2022: fix: supported events for *request_review* action #651

August 2, 2022: feat: Add newest_only commit validation setting #649

April 7, 2022: feat: Support adding deployment labels from values #631

March 22, 2022: fix: pass comment instance to removeErrorComments #626

February 24, 2022: fix: correct indentation on documentation #623

February 8, 2022: feat: Allow all option on must_include and must_include inside changeset validator #611

February 6, 2022: feat: Add commit_title and commit_message options to merge action so the merge commit can be customized based on PR content #612

December 12, 2021: feat: Add support for status events to baseRef validator #395

November 25, 2021: feat: Add more supported events to baseRef validator #395

November 12, 2021 : feat: Add baseRef filter #596

October 19, 2021 : feat: Add validator approval option to exclude users #594

October 12, 2021 : feat: Add boolean option for payload filter #583

September 27, 2021 : fix: use node version 14.17.6 for the release action #591

September 25, 2021 : feat: Support for objects in arrays in the payload filter #589

September 20, 2021 : fix: Check that getRouter is defined before attempting to access it #587

August 26, 2021 : fix: Await GithubAPI topics response #585

August 10, 2021 : feat: New labels API #577

August 6, 2021 : feat: Add team assignment to request_review action #574

August 6, 2021 : feat: Support must_include and must_exclude regex as an array #575

July 19, 2021 : feat: Add ignore_drafts option to ignore drafts in stale validator #565

July 12, 2021 : feat: Filter specific status of files in changeset #550

June 26, 2021 : feat: Add *payload* filter #398

March 31, 2021 : feat: add chart support to prometheus servicemonitor #535

March 30, 2021 : fix: codeowners team

March 25, 2021 : feat: Add rate limit related metrics/endpoint #526

February 25, 2021 : feat: add feature to limit approval to a list of users #509

February 25, 2021 : fix: minor bugs #508

February 25, 2021 : fix: Correct use of cache env

February 22, 2021 : feat: add *age* validator #367

February 22, 2021 : feat: add a info panel for 'limit' option in *approval* validator #509

February 21, 2021 : feat: *name* sub-option for *repository* filter

February 18, 2021 : fix: Scheduler support #499

February 12, 2021 : feat: Implemented redis as a dependency to the helm-chart

February 10, 2021 : feat: global cache manager #502

February 10, 2021 : feat: Implement and/or filters support #496

February 10, 2021 : feat: New Author filter #496

January 28, 2021 : feat: global settings feature

January 28, 2021 : feat: Filters validators #496

January 21, 2021 : fix: request_review action failing when the reviewer is PR author #486

January 20, 2021 : fix: deep validation bug that is causing HttpError #488

January 19, 2021 : feat: Async Options #480

January 19, 2021 : feat: Support use organization-wide configuration as default #470

January 19, 2021 : feat: Jira Validator Option #482

January 17, 2021 : fix: *required-status-check* bug in merge action

January 15, 2021 : feat: add prometheus templates for easy alerting setup

January 14, 2021 : fix: validators not running in certain pull_request events #431

January 14, 2021 : fix: remove unsupported settings from title, description and milestone validator

January 14, 2021 : feat: Allow to have pending checks #454

January 13, 2021 : fix: GH Action workflows for pushing to dockerhub.

January 12, 2021 : feat: upgrade probot to v11.0.1

January 8, 2021 : fix: Prevent add comma on last list element

January 5, 2021 : fix: Shift fix in team slug pagination

January 4, 2021 : feat: GitHub actions #450

December 18, 2020 : feat: Better logs for failures in PR home page without going to details #446

December 17, 2020 : fix: Members in Org listing pagination bug #442

December 17, 2020 : feat: Add docker image build and push #427

December 14, 2020 : feat: Add branch validator #438

December 11, 2020 : feat: Helm Chart implemantation for kubernetes #435

December 11, 2020 : feat: Add env to control which configuration file to use #429

October 20, 2020 : feat: Add a config cache which can be enabled via MERGEABLE_ENABLE_CONFIG_CACHE env var #407

October 15, 2020 : feat: Add the ability to auto-merge pull requests #395

October 8, 2020 : feat: added BaseRef-validator to enforce stricter rules on certain branches #343

October 8, 2020 : feat: Do not load modified unsafe config files from forks #406

October 6, 2020 : fix: Size validator - do not ignore hidden files by default #401

October 6, 2020 : Do not attempt to merge a pull request if the status is blocked #389

October 6, 2020 : fix: Fix undefined error with blank validators #402

October 5, 2020 : fix Typo in header of labels action docs and corresponding rst file

October 4, 2020 : fix Typo in header of title validator docs

October 2, 2020 : Don't throw merge error if required status are are the cause of the error #389

September 24, 2020 : Add ability to delete or replace the labels on an issue #380

September 22, 2020 : Allow support for customizing the scheduler interval via an enviroment variable *MERGEABLE_SCHEDULER_INTERVAL* #383

September 17, 2020 : Add support for *schedule.repository* event for'labels' and *close* actions #377

September 17, 2020 : Fix the comment action to work correctly with the scheduler #376

September 16, 2020 : Allow specifying files to *match* for the *size* validator #371

September 16, 2020 : *stale* validator now supports optionally skipping items associated with a *project* or a *milestone* #375

September 16, 2020 : *stale* validator now supports labels for *match* or *ignore* #372

August 24, 2020: display files processed in *size* validator #366

August 17, 2020: fix Error string in *merge failed* error

July 28, 2020 : owners file now support teams and limit.owners option added in *approvals* validator #331

July 12, 2020 : Allow usage of special annotation @*author* in comments and checks #328

July 1, 2020 : When config file is added/modified in base branch, mergeable will trigger for all PR against the base branch #153

June 30, 2020 : Add *ignore_comment* option to *size* validator #245

June 17, 2020 : Added new validator *contents* #207

June 16, 2020 : Create an error comment if errors have occurred during execution of actions #312

June 5, 2020 : For missing fields in 'checks', default values will be used #233

May 30, 2020 : New Action *merge* added #201

May 29, 2020 : throw *UnSupportedSettingError* if provided setting is not valid #228

May 29, 2020 : Ability to Limit *stale* validator to certain days and time #221

May 23, 2020 : Allow PRs/Issues to be assigned to their author by using @*author* in the *assign* action

May 14, 2020 : Delete obsolete comments by default #157

May 12, 2020 : Limit so that only approval from team members will count, #236

May 6, 2020 : Ability to create multiple checks with `named` recipe, #225

May 5, 2020 : Added ability to configure config file name using `CONFIG_PATH` env variable, #223

April 22, 2020 : readthedoc documentation added, start of CHANGELOG