

Syntax for GitHub's form schema

In this article

About GitHub's form schema

Keys

Further reading

You can use GitHub's form schema to configure forms for supported features.

Note: GitHub's form schema is currently in beta and subject to change.

About GitHub's form schema

You can use GitHub's form schema to configure forms for supported features. For more information, see "[Configuring issue templates for your repository](#)."

A form is a set of elements for requesting user input. You can configure a form by creating a YAML form definition, which is an array of form elements. Each form element is a set of key-value pairs that determine the type of the element, the properties of the element, and the constraints you want to apply to the element. For some keys, the value is another set of key-value pairs.

For example, the following form definition includes four form elements: a text area for providing the user's operating system, a dropdown menu for choosing the software version the user is running, a checkbox to acknowledge the Code of Conduct, and Markdown that thanks the user for completing the form.

YAML



```
- type: textarea
  attributes:
    label: Operating System
    description: What operating system are you using?
    placeholder: "Example: macOS Big Sur"
    value: operating system
  validations:
    required: true
- type: dropdown
  attributes:
    label: Version
    description: What version of our software are you running?
    multiple: false
    options:
      - 1.0.2 (Default)
      - 1.0.3 (Edge)
    default: 0
  validations:
    required: true
- type: checkboxes
  attributes:
    label: Code of Conduct
```

```
description: The Code of Conduct helps create a safe space for everyone. We
require
  that everyone agrees to it.
options:
  - label: I agree to follow this project's [Code of Conduct](link/to/coc)
    required: true
- type: markdown
  attributes:
    value: "Thanks for completing our form!"
```

Keys

For each form element, you can set the following keys.

Key	Description	Required	Type	Default	Valid values
type	The type of element that you want to define.	✓	String	—	<ul style="list-style-type: none">checkboxesdropdowninputmarkdowntextarea
id	The identifier for the element, except when type is set to markdown . Can only use alpha-numeric characters, -, , and _ . Must be unique in the form definition. If provided, the id is the canonical identifier for the field in URL query parameter prefills.	×	String	—	—
attributes	A set of key-value pairs that define the properties of the element.	✓	Map	—	—
validations	A set of key-value pairs that set constraints on the element.	×	Map	—	—

You can choose from the following types of form elements. Each type has unique attributes and validations.

Type	Description
markdown	Markdown text that is displayed in the form to

provide extra context to the user, but is **not submitted**.

<code>textarea</code>	A multi-line text field.
<code>input</code>	A single-line text field.
<code>dropdown</code>	A dropdown menu.
<code>checkboxes</code>	A set of checkboxes.

markdown

You can use a `markdown` element to display Markdown in your form that provides extra context to the user, but is not submitted.

Attributes for markdown

For the value of the `attributes` key, you can set the following keys.

Key	Description	Required	Type	Default	Valid values
<code>value</code>	The text that is rendered. Markdown formatting is supported.	✓	String	—	—

Tips: YAML processing will treat the hash symbol as a comment. To insert Markdown headers, wrap your text in quotes.

For multi-line text, you can use the pipe operator.

Example of markdown

YAML 

```
body:
- type: markdown
  attributes:
    value: "## Thank you for contributing to our project!"
- type: markdown
  attributes:
    value: |
      Thanks for taking the time to fill out this bug report.
```

textarea

You can use a `textarea` element to add a multi-line text field to your form. Contributors can also attach files in `textarea` fields.

Attributes for textarea

For the value of the `attributes` key, you can set the following keys.

Key	Description	Required	Type	Default	Valid values
<code>label</code>	A brief	✓	String	—	—

	description of the expected user input, which is also displayed in the form.				
description	A description of the text area to provide context or guidance, which is displayed in the form.	×	String	Empty String	—
placeholder	A semi-opaque placeholder that renders in the text area when empty.	×	String	Empty String	—
value	Text that is pre-filled in the text area.	×	String	—	—
render	If a value is provided, submitted text will be formatted into a codeblock. When this key is provided, the text area will not expand for file attachments or Markdown editing.	×	String	—	Languages known to GitHub. For more information, see the languages YAML file .

Validations for `textarea` [↗](#)


For the value of the `validations` key, you can set the following keys.

Key	Description	Required	Type	Default	Valid values
required	Prevents form submission until element is completed. Only for public repositories.	×	Boolean	false	—

Example of `textarea` [↗](#)

YAML

body:



```
- type: textarea
id: repro
attributes:
  label: Reproduction steps
  description: "How do you trigger this bug? Please walk us through it step by
step."
  value: |
    1.
    2.
    3.
    ...
  render: bash
validations:
  required: true
```

input

You can use an `input` element to add a single-line text field to your form.

Attributes for `input`

For the value of the `attributes` key, you can set the following keys.

Key	Description	Required	Type	Default	Valid values
label	A brief description of the expected user input, which is also displayed in the form.	✓	String	—	—
description	A description of the field to provide context or guidance, which is displayed in the form.	×	String	Empty String	—
placeholder	A semi-transparent placeholder that renders in the field when empty.	×	String	Empty String	—
value	Text that is pre-filled in the field.	×	String	—	—


Validations for `input`

For the value of the `validations` key, you can set the following keys.

Key	Description	Required	Type	Default	Valid values
required	Prevents form submission until element is completed.	×	Boolean	false	—

Only for public repositories.

Example of `input` [↗](#)

YAML 

```
body:
- type: input
  id: prevalence
  attributes:
    label: Bug prevalence
    description: "How often do you or others encounter this bug?"
    placeholder: "Example: Whenever I visit the personal account page (1-2 times a week)"
  validations:
    required: true
```

dropdown [↗](#)

You can use a `dropdown` element to add a dropdown menu in your form.

Attributes for `dropdown` [↗](#)

For the value of the `attributes` key, you can set the following keys.

Key	Description	Required	Type	Default	Valid values
label	A brief description of the expected user input, which is displayed in the form.	✓	String	—	—
description	A description of the dropdown to provide extra context or guidance, which is displayed in the form.	✗	String	Empty String	—
multiple	Determines if the user can select more than one option.	✗	Boolean	false	—
options	An array of options the user can choose from. Cannot be empty and all choices must be distinct.	✓	String array	—	—
default	Index of the	✗	Integer	—	—

preselected option in the `options` array. When a default option is specified, you cannot include "None" or "n/a" as options.

Validations for `dropdown` [↗](#)

For the value of the `validations` key, you can set the following keys.

Key	Description	Required	Type	Default	Valid values
<code>required</code>	Prevents form submission until element is completed. Only for public repositories.	X	Boolean	false	—

Example of `dropdown` [↗](#)

YAML



```
body:
- type: dropdown
  id: download
  attributes:
    label: How did you download the software?
    options:
      - Built from source
      - Homebrew
      - MacPorts
      - apt-get
  default: 0
  validations:
    required: true
```

checkboxes [↗](#)

You can use the `checkboxes` element to add a set of checkboxes to your form.

Attributes for `checkboxes` [↗](#)

For the value of the `attributes` key, you can set the following keys.

Key	Description	Required	Type	Default	Valid values
<code>label</code>	A brief description of the expected user input, which is displayed in the form.	✓	String	—	—

description	A description of the set of checkboxes, which is displayed in the form. Supports Markdown formatting.	×	String	Empty String	—
options	An array of checkboxes that the user can select. For syntax, see below.	✓	Array	—	—

For each value in the `options` array, you can set the following keys.

Key	Description	Required	Type	Default	Options
label	The identifier for the option, which is displayed in the form. Markdown is supported for bold or italic text formatting, and hyperlinks.	✓	String	—	—
required	Prevents form submission until element is completed. Only for public repositories.	×	Boolean	false	—

Validations for checkboxes [↗](#)

For the value of the `validations` key, you can set the following keys.

Key	Description	Required	Type	Default	Valid values
required	Prevents form submission until element is completed. Only for public repositories.	×	Boolean	false	—

Example of checkboxes [↗](#)

YAML [↗](#)

```
body:
- type: checkboxes
  id: operating-systems
```



```
attributes:  
  label: Which operating systems have you used?  
  description: You may select more than one.  
  options:  
    - label: macOS  
    - label: Windows  
    - label: Linux
```

Further reading

- [YAML](#)

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)