How to find "Good First Issues"

Quipper Limited

@ohbarye at Roppongi.js#4

If you:

- want to contribute OSS
- are struggling to find a repository or an issue to contribute
- (would rather you preferred major OSS 😇)

This talk would be helpful to you.

Content

A tip to find "Good First Issues" with a script to make an issue list.

- This talk focuses on just a way to find them.
- See OSS貢献超入門 if you want to know next steps.

Issue List

First of all, see the "good first issue" list.

	A	B	C	D	E
1	owner	name	stars	title	url
2	vuejs	vue	103892	warn if $set is used on a property that already exist	https://github.com/vuejs/vue/issues/8129
3	vuejs	vue	103892	Creating a component named map should warn the user (as with button)	https://github.com/vuejs/vue/issues/8025
4	vuejs	vue	103892	Make the warning messages more explicit	https://github.com/vuejs/vue/issues/7764
5	vuejs	vue	103892	2.5.14 introduces TypeError bind is not a function	https://github.com/vuejs/vue/issues/8017
6	facebook	react	103700	Nicer Formatting of SSR Validation	https://github.com/facebook/react/issues/10085
7	facebook	react	103700	Ignore <noscript> content on the client and don't warn about mismatches	https://github.com/facebook/react/issues/11423
8	facebook	react	103700	Children of option tags flattened when select has value	https://github.com/facebook/react/issues/11602
9	facebook	react	103700	False positive getDefaultProps warning when mixing development and production versions	https://github.com/facebook/react/issues/9999
10	nodejs	node	50510	Inconsistent documentation: hostname vs host	https://github.com/nodejs/node/issues/20892
11	nodejs	node	50510	Debugging: name every function	https://github.com/nodejs/node/issues/8913
12	nodejs	node	50510	dns.js not properly iterating through DNS array	https://github.com/nodejs/node/issues/21391

BTW, what is "good first issue"?

	A	B	C	D	E
1	**owner**	**name**	**stars**	**title**	**url**
2	vuejs	vue	103892	warn if $set is used on a property that already exist	https://github.com/vuejs/vue/issues/8129
3	vuejs	vue	103892	Creating a component named map should warn the user (as with button)	https://github.com/vuejs/vue/issues/8025
4	vuejs	vue	103892	Make the warning messages more explicit	https://github.com/vuejs/vue/issues/7764
5	vuejs	vue	103892	2.5.14 introduces TypeError bind is not a function	https://github.com/vuejs/vue/issues/8017
6	facebook	react	103700	Nicer Formatting of SSR Validation	https://github.com/facebook/react/issues/10085
7	facebook	react	103700	Ignore <noscript> content on the client and don't warn about mismatches	https://github.com/facebook/react/issues/11423
8	facebook	react	103700	Children of option tags flattened when select has value	https://github.com/facebook/react/issues/11602
9	facebook	react	103700	False positive getDefaultProps warning when mixing development and production versions	https://github.com/facebook/react/issues/9999
10	nodejs	node	50510	Inconsistent documentation: hostname vs host	https://github.com/nodejs/node/issues/20892
11	nodejs	node	50510	Debugging: name every function	https://github.com/nodejs/node/issues/8913
12	nodejs	node	50510	dns.js not properly iterating through DNS array	https://github.com/nodejs/node/issues/21391

Apply the help wanted and good first issue labels to issues in your repository to highlight opportunities for people to contribute to your project.

https://help.github.com/articles/helping-new-contributors-find-your-project-with-labels/

How to make the list

To make it, I wrote a simple script in
https://github.com/ohbarye/goofi.

1. Call GitHub GraphQL API
2. Format its response
3. Create CSV

GraphQL Query

```
 2    search(first: 100, query: "language:javascript good-first-issues:>1 stars:>500", type: REPOSITORY) {
 3      repositoryCount
 4      pageInfo {
 5        startCursor
 6        endCursor
 7        hasNextPage
 8      }
 9      nodes {
10        ... on Repository {
11          owner {
12            login
13          }
14          name
15          url
16          issues(first: 100, labels: ["good first issue"], states: OPEN, orderBy: {field: UPDATED_AT, direction: DESC}) {
17            totalCount
18            nodes {
19              title
20              url
21            }
22          }
23          stargazers {
24            totalCount
25          }
26        }
27      }
28    }
```

"good-first-issues:>1"

```
search(first: 100,
       query: "language:javascript good-first-issues:>1 stars:>500",
       type: REPOSITORY) {
```

ref: GitHub help

"stars:>500"

```
search(first: 100,
       query: "language:javascript good-first-issues:>1 stars:>500",
       type: REPOSITORY) {
```

ref: GitHub help

Response

```
{
  "data": {
    "search": {
      "repositoryCount": 197,
      "pageInfo": {
        "startCursor": "Y3Vyc29yOjE=",
        "endCursor": "Y3Vyc29yOjEwMA==",
        "hasNextPage": true
      },
      "nodes": [
        {
          "owner": {
            "login": "vuejs"
          },
          "name": "vue",
          "url": "https://github.com/vuejs/vue",
          "issues": {
            "totalCount": 4,
            "nodes": [
              {
                "title": "warn if $set is used on a property that already exist",
                "url": "https://github.com/vuejs/vue/issues/8129"
              },
```

Once we can easily get data via the GitHub API, it's not a hard task to format the data.

```javascript
writeIssues(repository) {
  const owner = repository.owner.login;
  const name = repository.name;
  const stars = repository.stargazers.totalCount;

  repository.issues.nodes.forEach((issue) => {
    const title = issue.title;
    const url = issue.url;
    this.writer.write({owner, name, stars, title, url});
  });
}
```

Now that we have the candidate list, all we have to do is check issues one by one.

	A	B	C	D	E
1	owner	name	stars	title	url
2	vuejs	vue	103892	warn if $set is used on a property that already exist	https://github.com/vuejs/vue/issues/8129
3	vuejs	vue	103892	Creating a component named map should warn the user (as with button)	https://github.com/vuejs/vue/issues/8025
4	vuejs	vue	103892	Make the warning messages more explicit	https://github.com/vuejs/vue/issues/7764
5	vuejs	vue	103892	2.5.14 introduces TypeError bind is not a function	https://github.com/vuejs/vue/issues/8017
6	facebook	react	103700	Nicer Formatting of SSR Validation	https://github.com/facebook/react/issues/10085
7	facebook	react	103700	Ignore <noscript> content on the client and don't warn about mismatches	https://github.com/facebook/react/issues/11423
8	facebook	react	103700	Children of option tags flattened when select has value	https://github.com/facebook/react/issues/11602
9	facebook	react	103700	False positive getDefaultProps warning when mixing development and production versions	https://github.com/facebook/react/issues/9999
10	nodejs	node	50510	Inconsistent documentation: hostname vs host	https://github.com/nodejs/node/issues/20892
11	nodejs	node	50510	Debugging: name every function	https://github.com/nodejs/node/issues/8913
12	nodejs	node	50510	dns.js not properly iterating through DNS array	https://github.com/nodejs/node/issues/21391

Another way

Use Jasper

Create a stream to find "good first issues"

NOTE

I used to find issues with this way, but...

- It cannot sort repositories by stars count
- It cannot reject wrong usage of "good first issue"
- It notifies me every when an issue is updated

In short, that was too noisy.

Are they really nice ways? 🤔

At least, I could have contributed some popular repositories even though I was not familiar with them.

e.g. Node.js

Let's find your "good first issues"
And contribute! 💪

Who?

- @ohbarye http://ohbarye.me/
- Web Developer / Engineering Manager
- Working for Quipper

7/19 (Thu) 19:30-22:00

https://techplay.jp/event/680406

スタディサプリ

Product Meetup #1