



# **Drupal Contributions - A Pragmatic Approach to the Issue Queue**



Who am I ??



# AmyJune Hinline

She/Her/Hers

Open Source Community Ambassador at

Kanopi Studios

**@volkswagenchick**



- Mother and Palliative Care Nurse
- Hobbies include: comics, geocaching, mushroom hunting, and volkswagening





# Kanopi Studios

We create open source websites that make an impact.

[www.kanopistudios.com](http://www.kanopistudios.com)



Who are you ??





Everyone!!



# What are we going to talk about??

- Why contribute
- Benefits of contributing
- How to start contributing
- Keeping the contributors we already have
- Ways to foster new contributors
- Anatomy of an issue

If there is time after the demo, we'll look at the new GitLab



# Why contribute





# Why Contribute

- If you depend on open source, open source depends on you
- Your contributions are valued
- You Get What You Give (and Then Some)
- Contributing makes you a more integrated part of the Drupal community



Come for the code,  
stay for the  
community



# Benefits of contributing



# Benefits of contributing

## More than warm and fuzzies

- Contributing makes you feel good
- Move the Drupal project closer to perfection
- Drupal street cred
- The more you contribute, the more you learn



# Why individuals give back...

## Build your resume

- Work on a team
  - Willingness to collaborate
  - Issues parallel working on tickets
- Write better code
  - Learn from other's comments
  - Learn coding standards



# Why Drupal shops should give back...

## Why should agencies give back?

- Drupal depends on sharing
  - Good corporate citizenship
- Talent recruitment
  - Potential employees view profile pages
  - People grow to love the community and attending events



# How to start contributing

# Ways to contribute besides the issue queue

- Documentation
- External documentation guides
- Speaking at camps
- Translations
- Writing themes and modules
- Promote the platform





# Ways to contribute in the issue queue...

- Report an issue
- Update issues
- Write patches
- Provide feedback
- Test patches

<https://www.drupal.org/contributor-tasks>



**Keeping the contributors  
we already have**

# Why people leave...

- Burnout
- Work/Life balance
- Lack of recognition
- Lack of employer incentives
- Huge learning curves between releases



# Fighting burnout

Find folks new roles

- Leadership
  - Maintain projects
  - Run initiatives
- Mentoring
  - Workshops
  - Speaking at events



# Ways to foster new contributors

# How do we recruit new contributors?

**Mentor!!**

**Mentor!!**

**Mentor!!**



Let's talk issue queue!!





Find an issue...



# The issue queue... Start small and work your way up.

**Patching can be daunting**

Start with :

- Triage
- Tagging
- Test and review code
- Apply patches



# There's so many to choose from!?!


## Choose based on

- Skill ability
- Area of interest
- Actively maintained

**8.x-1.7**  released 31 August 2018  
✓ Recommended by the project's maintainer.  
Important bug fix release for sites which used Schema.org Metatag.  
↓ [tar.gz \(138.08 KB\)](#) | [zip \(391.73 KB\)](#)

Development version: [8.x-1.x-dev](#) updated 31 Aug 2018 at 15:18 UTC

Testing result: **PHP 7.2 & MySQL 5.5, D8.6 143 pass** [all results](#)

**7.x-1.25**  released 4 April 2018  
✓ Recommended by the project's maintainer.  
Fixes a glaring bug on entity forms.  
↓ [tar.gz \(229.27 KB\)](#) | [zip \(330.54 KB\)](#)

Development version: [7.x-1.x-dev](#) updated 11 Apr 2018 at 13:48 UTC

Testing result: **PHP 7.1 & MySQL 5.5, D7 271 pass** [all results](#)



# Issues to steer clear of

As a beginner, pass on

- Issues with over 100 comments
- Issues that have changed status many times
- Projects that many open issues
- Multiple issues being closed with “won’t fix”



# Report an issue..



# What kind of issues are worth reporting?

**All of them!**

- UI errors
- Documentation
- Accessibility
- Bugs

You don't have to know how to fix the issue to report it.



# Search the queue...

## Does it already exist?

- Update any changes
- Report similar findings
- Add steps to replicate



# Anatomy of an issue...

A good ticket should provide correct metadata:

- Descriptive and concise title
- Category
- Priority
- Status
- Version
- Component
- Tags





# Anatomy of an issue...

A good ticket should provide correct metadata:

## ▼ Issue metadata

**Title \***

[regression] Table Drag handles no longer respond to up/down arrow keys

**Project \***

Drupal core (3060)

**Category \***

Bug report

**Priority \***

Major

**Status \***

Needs review

**Version \***

8.7.x-dev

**Component \***

javascript

**Assigned \***

Unassigned

Descriptions of the [Priority](#) and [Status](#) values can be found in the [Issue queue handbook](#).

**Issue tags**

accessibility, JavaScript, regression, Triaged for D8 major current state, Triaged D8 major, Vienna2017, badcamp 2018

Before adding tags [read the issue tag guidelines](#). Do NOT use tags for adding random keywords or duplicating any other fields. Separate terms with a comma, not a space.



# Anatomy of an issue...

Dreditor provides an issue summary template:

- Problem/motivation
  - Steps to reproduce
  - Expected results
  - Actual results
- Proposed resolution
- Remaining tasks
- Ui and API changes
- Data model changes



# Anatomy of an issue...

```
<h3 id="summary-problem-motivation">Problem/Motivation</h3>
```

```
<h3 id="summary-proposed-resolution">Proposed resolution</h3>
```

```
<h3 id="summary-remaining-tasks">Remaining tasks</h3>
```

```
<h3 id="summary-ui-changes">User interface changes</h3>
```

```
<h3 id="summary-api-changes">API changes</h3>
```

```
<h3 id="summary-data-model-changes">Data model changes</h3>
```

```
<h3 id="summary-release-notes">Release notes snippet</h3>
```



A good issue is worth its weight in latinum...



Write a patch...



# Do you know how to solve the issue?

## Write a patch:

- Download the repo
- Create a new branch
- Makes changes
- Compare the branches
- Upload the patch and write a comment



Review an issue...



# Reviewed and tested by the community (RTBC)...

**You don't have to know code to test a patch!!**

- Test the patch
  - Simplytest.me
  - Locally in git repo
- Provide useful information
- Attach screenshots
- Update the status





# Reviewed and tested by the community (RTBC)...

## Add new comment #3

▼ Issue metadata

**Title \***  
Switching to V2 API

**Project \***  
Drupal Chatbot (3002462)

**Category \*** Support request | **Priority \*** Normal | **Status \*** Active

**Version \*** 8.x-1.0 | **Component \*** Code | **Assigned \*** Unassigned

**Issue tags**

**Attribute this contribution**

I'm volunteering my own time

volkswagenchick at organization

When this issue is fixed, the project will be in a better state than when I created the issue.

Before adding tags [read the issue tags policy](#)

Random keywords or duplicating any other fields. Separate terms with a comma, not a space.

Not applicable

Learn more

- Active
- Needs work
- Needs review
- Reviewed & tested by the community
- Patch (to be ported)
- Fixed
- Postponed
- Postponed (maintainer needs more info)
- Closed (duplicate)
- Closed (won't fix)
- Closed (works as designed)
- Closed (cannot reproduce)
- Closed (outdated)



# Reviewed and tested by the community (RTBC)... Dreditor

Comment	File		
	<a href="#">interdiff-26-29.txt</a>	9.33 KB	<a href="#">REVIEW</a>
<b>#29</b> Elaman	<a href="#">3003707-29.patch</a>	15.58 KB	<a href="#">REVIEW</a> <a href="#">SIMPLYTEST.ME</a>
	8.x-3.x: <span>PHP 5.5 &amp; MySQL 5.5, D8.7 5 pass</span>		
	<a href="#">Add test / retest</a>		



# Reviewed and tested by the community (RTBC)... Dreditor

Comment selected code:

SAVE

CANCEL

```
19 19      ^ return hello string.
20 20      */
21 21      public function renderTaxonomyLinks() {
22 22      -   $markup = '';
23 23      -
24 24      // Load taxonomy menus.
25 25      -   $storage = \Drupal::entityTypeManager()->getStorage('taxonomy_menu');
26 26      -   $taxonomy_menus = $storage->loadMultiple();
27 27      +   $taxonomy_menus = $this->entityTypeManager()
28 28      +   ->getStorage('taxonomy_menu')
29 29      +   ->loadMultiple();
30 30      $links = [];
31 31
32 32      // Get taxonomy and create menu links from vocabularies.
33 33      @@ -32,7 +31,7 @@ class TaxonomyMenu extends ControllerBase {
34 34      return [
35 35      'type' => 'markup',
36 36      -   '#markup' => $this->t($markup),
37 37      +   '#markup' => '',
38 38      ];
39 39      }
```

PASTE

HIDE

CANCEL





Live demo!!!

# The Process..

- Find a problem
- Download the repo
- File an issue
- Create a new branch
- Make changes
- Create a patch
- Upload a patch



# The move to GitLab



# Why move the issue queue to GitLab?

## The need to modernize

- Adopt a standard workflow that will be familiar to others outside our community
- Preserve those unique elements of how we collaborate
- Leverage an expert partner



# Gitlab workflow...





# Resources

- <https://about.gitlab.com/2018/08/16/drupal-moves-to-gitlab/>
- <https://www.drupal.org/drupalorg/blog/a-visual-prototype-of-drupalorgs-integration-with-gitlab>
- <https://hbswk.hbs.edu/item/the-hidden-benefit-of-giving-back-to-open-source-software>
- <http://www.tothenew.com/blog/how-to-give-back-to-drupal-community/>
- <https://dri.es/contributing-back-to-drupal>





# Q & A



Thank you!!

# Issue and download repo

- Navigate to project page's issue queue and create a new issue
- Fill out metadata
  - Category: Task (usually)
  - Priority: Normal or minor
  - Status: remain the same, Active
  - Select latest branch (usually)
  - Component: Make a good judgement call
  - Assigned: Yourself (If you are going to work on it in the next couple of days)
- Save
- Take note of node # from issue just created
- Navigate to project page and select the version control
  - Select the desired version and show
  - Copy git clone link
    - example: `git clone --branch 7.x-1.x`  
<https://git.drupal.org/project/i18n.git>
    - `cd i18n`



# Creating a Patch

- Create a new branch
  - `git checkout -b [issue node number]-[comment number]`
- Make changes to new branch and save
- Commit changes
  - `git add .`
  - `git commit -m "useful commit message"`
- Create a diff
  - `git diff [old branch] [new branch] > [project name]-[descrip]-[node number]-[comment number]`
  - Example: `sticky_toolbar-capitalize-name-2981572-4.patch`



# Useful Terminal Commands

- Basic terminal commands
  - `cd` - change directory
  - `mkdir` - make a new directory or file
  - `pwd` - present working directory (where are you)
  - `open .` - opens file in Finder
- Git commands
  - `git clone`
  - `git branch`
  - List all branches
    - `git branch -a`
  - Change branches
    - `git branch checkout [name of branch]`
  - Create a new branch
    - `git branch checkout -b [name of branch]`
  - `git status`
  - `git add`
  - `git commit`

