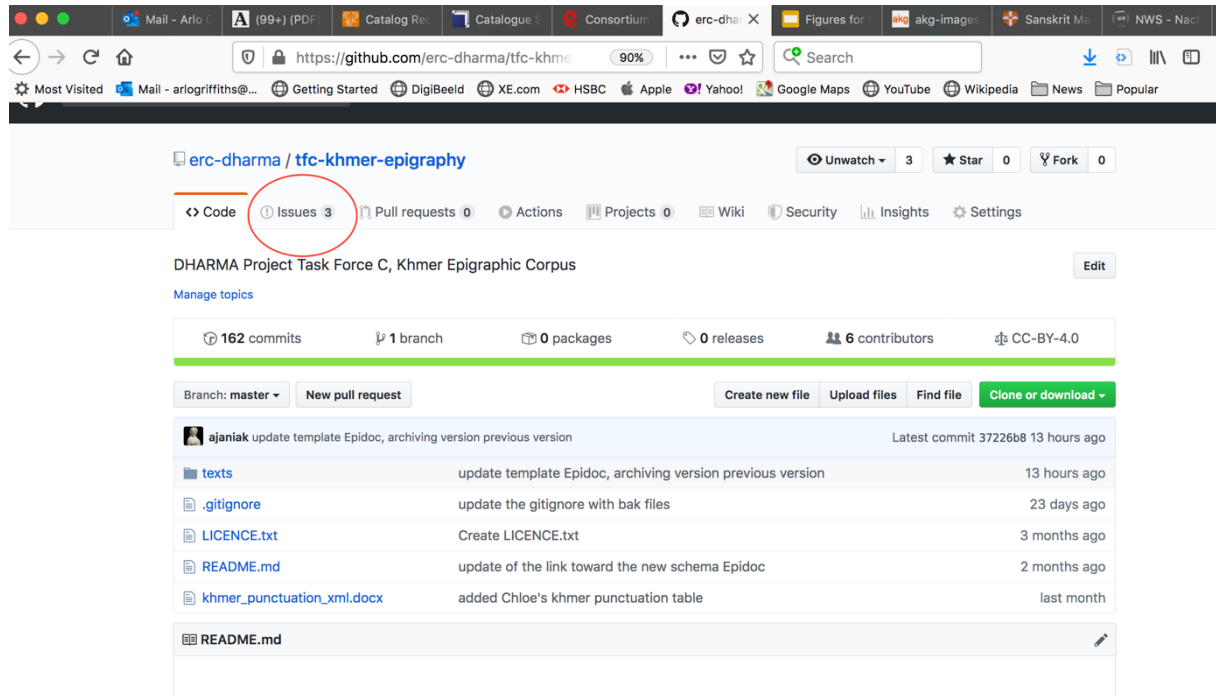
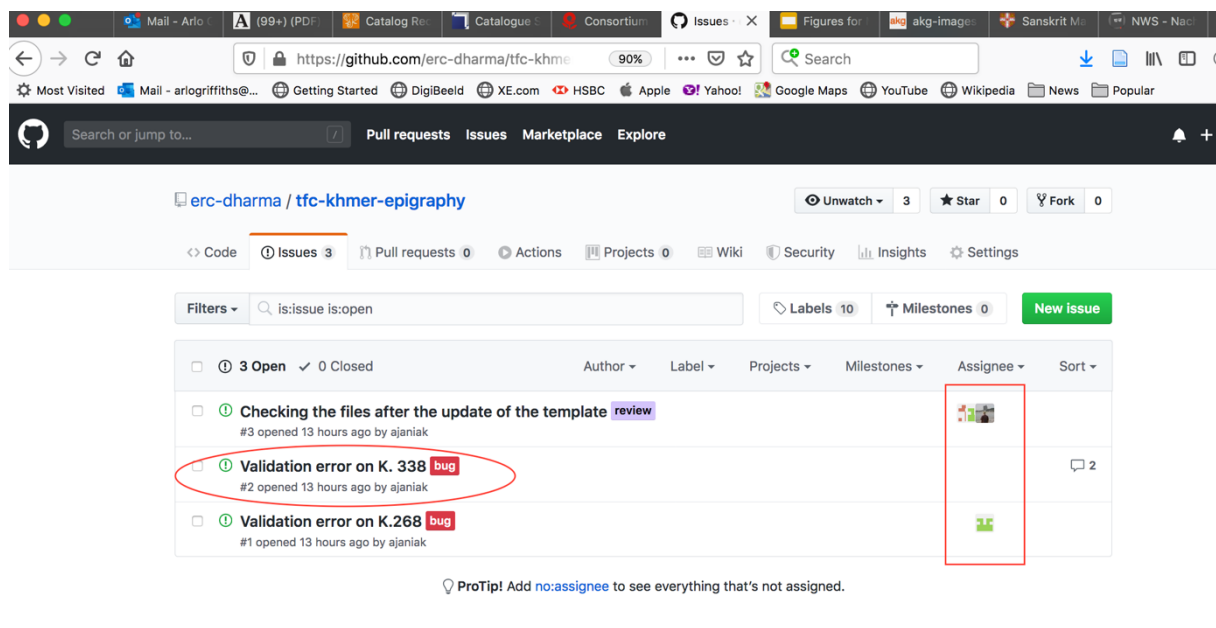


How to respond to issues with GitHub's Issue Tracker

Step 1: go to the repository in question and click on 'Issues'



Step 2: Identify the issue by its number, and see whether it has been assigned to anyone specific



Step 3: See what comments have been made on the issue

The screenshot displays a GitHub issue thread. The top comment by **ajaniak** (Member) from 13 hours ago discusses an XML encoding issue. It mentions that a certain element can only contain a `<g>` element and that a specific line in file K. 338 does not conform to the Epidoc TEI model. The comment includes a code snippet: `<unclear>tgap</unclear> <num value="1">I</num></unclear> .` and asks for resolution. A second comment by **arlogriffiths** (Member) from 12 hours ago asks "Would this be acceptable?" and provides a code snippet: `<unclear>tgap</unclear> <num value="1"><unclear>I</unclear></num>`. A third comment by **ajaniak** (Member, Author) from 12 hours ago responds "Yes, it would." The right sidebar shows metadata: Assignees (None), Labels (bug), Projects (None), Milestone (None), Notifications (Unsubscribe), and 2 participants.

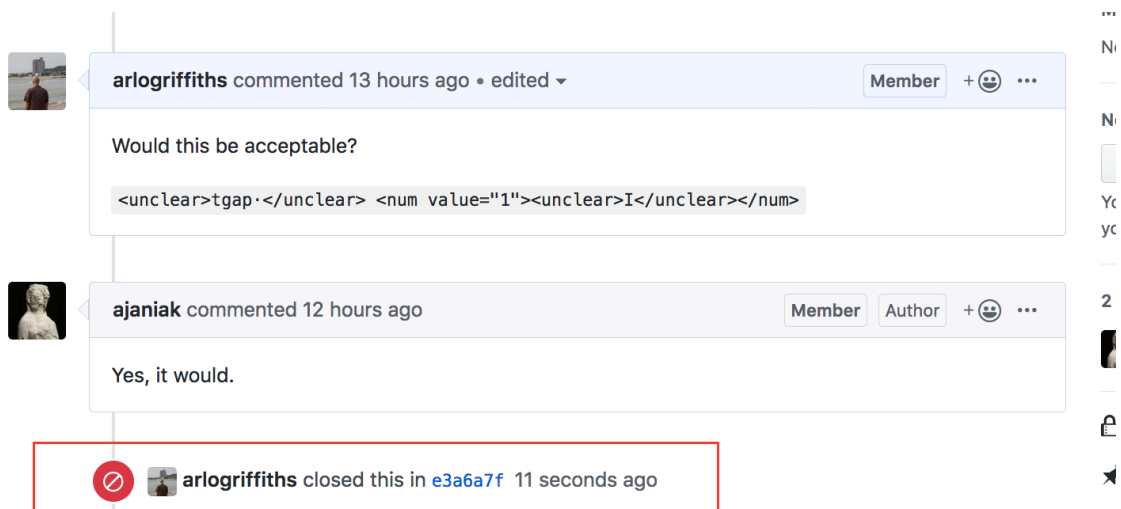
Step 4: Make the necessary changes in the relevant XML file, and record all changes made by adding a new `<change>` entry to `<revisionDesc>`

```
42 | <revisionDesc>
43 |   <change who="part:argr" when="2020-01-10" status="draft">Resolved issue #2, filled in respStmt, assigned copyright to Kunthea
44 | Chhom, and added two lines in revisionDesc</change>
45 |   <change who="part:axja" when="2020-01-09" status="draft">Update template</change>
46 |   <change who="part:kuch" when="2019" status="draft">Initial encoding of the text</change>
47 | <!-- add a <change> for each change to the file, the first change should replace this one and record your creation of a file using this template. -->
48 | </revisionDesc>
```

Step 5: Save the file, add and commit it to the repository, applying a commit message that makes explicit which issue number (#) has been treated, and push the commit.

```
MacBook-Pro-20:tfc-nusantara-epigraphy arlogriffiths$ git pull
Already up-to-date.
MacBook-Pro-20:tfc-nusantara-epigraphy arlogriffiths$ cd ../tfc-khmer-epigraphy/
MacBook-Pro-20:tfc-khmer-epigraphy arlogriffiths$ git pull
Already up-to-date.
MacBook-Pro-20:tfc-khmer-epigraphy arlogriffiths$ git add -A
MacBook-Pro-20:tfc-khmer-epigraphy arlogriffiths$ git commit -m "resolved #2"
[master e3a6a7f] resolved #2
 1 file changed, 12 insertions(+), 9 deletions(-)
MacBook-Pro-20:tfc-khmer-epigraphy arlogriffiths$ git push
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 744 bytes | 0 bytes/s, done.
Total 5 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To github.com:erc-dharma/tfc-khmer-epigraphy.git
 37226b8..e3a6a7f master -> master
MacBook-Pro-20:tfc-khmer-epigraphy arlogriffiths$
```

Step 6: This automatically communicated to GitHub that the issue has been resolved, as you will almost instantly see on GitHub's interface.



It is, however, also possible to use a different commit message and close on issue manually on GitHub's interface.