

Proposal: Implement Translation Message Widget Backend

Written by: Szymon <fas: skrzepo>

Description:

- Widget that displays % translated statistics for various critical pieces of Fedora. (How do we define critical?) Goal of the widget would be to identify those parts of Fedora in most critical need of translation and try to encourage translators to pop over to those items in Zanata and submit some new translations.
- Widget that displays translation submissions in need of review. From what I understand thus far, translated strings must be approved by a reviewer on the translation team. The goal of this widget would be to draw more attention to strings that have been submitted but need review to get them in.
- Widget that displays a random untranslated string to tease the logged in user to submit a translation. This one could be tricky because we need to know the language the user can translate in order for the widget to be useful

src: <https://lists.fedoraproject.org/archives/list/hubs-devel@lists.fedoraproject.org/message/ZMGUOK63U76RTGCOFOBDFI6RDOLQSS6D/>

Sample Use Case:

In the HUBS widget it will pop prompting:

"Hey, you! Yeh, you there!

What languages do you speak? [dropdown of Fedora-supported langs]"

... "Hey, great, want to help us out? Here's a quick string that needs translation:

\$string

[translate this string] <= goes to zanata"

This widget could be useful outside of the translation hub too.

src: <https://lists.fedoraproject.org/archives/list/hubs-devel@lists.fedoraproject.org/message/ZMGUOK63U76RTGCOFOBDFI6RDOLQSS6D/>

Technical Background: Unfortunately Zanata has a fairly weak webhook and only reports on milestones <http://docs.zanata.org/en/release/user-guide/projects/project-settings/#w...>

...
Currently, webhooks events will be triggered when

- A document has reached 100% Translated
- A document has reached 100% Approved (by reviewer)

But Zanata does provide a python-client that is able to interface with the server and pretty much do everything you can on web

<https://github.com/zanata/zanata-python-client>

Implementation

My idea is that my program will query the zanata server every so often lets say every hour per project. The frequency of the fresh at the moment is arbitrary and can be adjusted. I will then store the strings needing to be translated stored in the db with project-id, project-version, msgid, language and date last refreshed from zanata.

Below I show how the data is received for two languages for the same project “abrt” The line “Language: <>” is not actually in the file but I can determine which language by which folder it's stored in since its labeled as the language. Potential problem is multiple version of the project look at section *potential problems*.

```
Language: AF
```

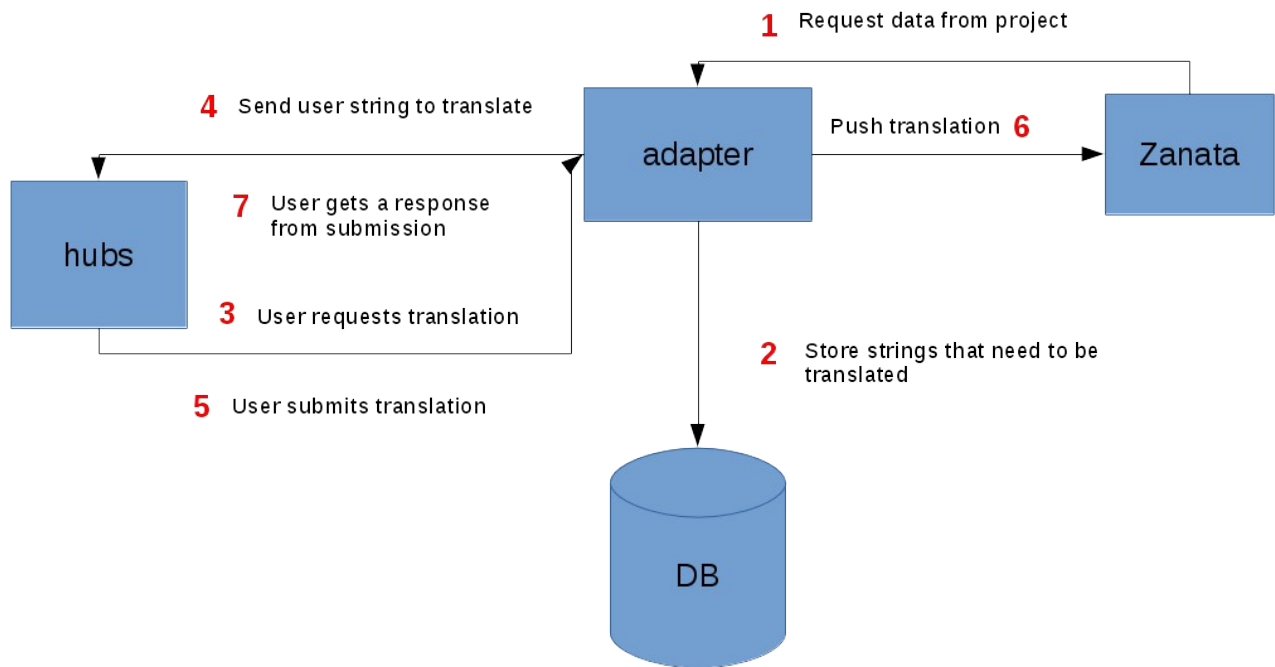
```
#: ../src/applet/abrt-applet.desktop.in.h:1  
msgid "Problem Reporting"  
msgstr ""
```

```
Language: PL
```

```
#: ../src/applet/abrt-applet.desktop.in.h:1  
msgid "Problem Reporting"  
msgstr "Zgłaszanie problemów"
```

The program can get a query from the widget saying get me the oldest (or newest??) string to translate in this language. The user then enters their translation and submits it. Then we need some way of notifying the user that the review was successfully sent or perhaps there was an issue sending the review.

Big assumption with this program is that we are translating from English to <language>



Potential Problems

Issues that may need more research

1. Interfacing FAS login into this program and making sure security is good
2. Submitting a translation that has been already completed but the server hasn't refreshed yet
 - a) Perhaps, before submitting I can just check that particular string has been translated in Zanata.
3. Multiple users submitting patch at the same time.
 - a) Maybe I need to add a tag in the db that says in review and prompt the user. "Sorry someone has sent a review before you"
4. Processing multiple project versions
 - a) The problem here is that when I request Zanata to pull the translation it throws it into <pwd>/PL/abrt.po

but abrt may have multiple versions like master, rhel-6, rhel-7, etc... so I might need to create directories for each version and pull the data there so it won't get overwritten

5. Performance of the db with concurrent connections.
 - a) Postgres with transactions??

Milestones

1. Be able to pull the project with various versions and store them into the db if translation is needed
2. Query the db manually for a string to translate in specified languages
3. Submit a translation to the db and handle the push
4. Write code that refreshes the db at a certain frequency

Misc Comments/Concerns

1. Should I get this implemented into fedmsg or as a separate service?