

Crowdin library usage

Project initialization

Adding folders to the remote project

- `crowdin:translations:add_dirs[locale]`
- `bundle exec rake crowdin:translations:add_dirs[locale]`

Adding files to the remote project

- `crowdin:translations:add_files[locale]`
- `bundle exec rake crowdin:translations:add_files[locale]`

Setting up the remote project with a single command

- `crowdin:translations:setup[locales]`
- `bundle exec rake crowdin:translations:setup[locales]`

Warning: These commands must be run only once (at the moment of creation). To avoid translation progress losses better use: `sync` and `update_files` when remote project is already initialized

Warning: `setup` command does the following in background:

- removes **/download** folder if present
- removes remote folders
- adds remote folders
- adds remote files

Translations fetching

Export translations (*build latest changes on Crowdin*)

- `crowdin:translations:export[locale]`
- `bundle exec rake crowdin:translations:export[locale]`

Download translations

- `crowdin:translations:download[locale]`
- `bundle exec crowdin:translations:download[locale]`

Note: Run `export` command each time before `download`

Note: `download` command will fetch the remote files, create a local folder **config/locales/download** and store the fetched files there. Files can be manually copied to **config/locales/<locale>/...**

Files synchronization

Synchronize folders

- `crowdin:translations:sync_dirs[locale]`
- `bundle exec rake crowdin:translations:sync_dirs[locale]`

Synchronize files

- `crowdin:translations:sync_files[locale]`
- `bundle exec rake crowdin:translations:sync_files[locale]`

Synchronize folders and files with a single command

- `crowdin:translations:sync[locale]`
 - `bundle exec rake crowdin:translations:sync[locale]`
-

Translation keys update

Update all remote files (*keeps translation progress*)

- `crowdin:translations:update_files[locale]`
- `bundle exec rake crowdin:translations:update_files[locale]`

Update specific remote files

- `crowdin:translations:update_files[locale] -- --file <path to file>`
- `bundle exec rake crowdin:translations:update_files[locale] -- --file <path to file>`

*Note: Adds the missing keys to the remote project and keeps the **Crowdin** translation progress*

Translations status

- `crowdin:translations:status[locale]`
- `bundle exec rake crowdin:translations:status[locale]`

Note: This command will return the information about translation progress

Example of usage

Starting point. Phase 1

- **Initialization**
 - 1st option
 - `bundle exec rake crowdin:translations:add_dirs[locale]`
 - `bundle exec rake crowdin:translations:add_files[locale]`
 - 2nd option
 - `bundle exec rake crowdin:translations:setup[locale]`

Phase 2

Working with translations is a continuous process. It is a matter of time when some member of the development team will add a new translation file or a new translation key to an existing file, or even both of them at the same time. Hence one needs to synchronize periodically the local files with the remote ones so that the *Crowdin* translator sees the new files/strings and performs the translation process.

- `bundle exec rake crowdin:translations:sync[locale]`

Note: The `sync` and `update_files` commands must be used to let the translator keep track of the new translation files and keys.

Note: The `status` command may be used to keep track of translator's progress.

Phase 3

This is not actually a phase. Whenever is needed one can run the `export` and `download` commands and fetch the translations from **Crowdin**.

What can be improved?

- Add validation for `add_dirs`, `add_files` and `setup` commands. Skip the execution if remote project had been already initialized