

# LocalizeTranslator 2023

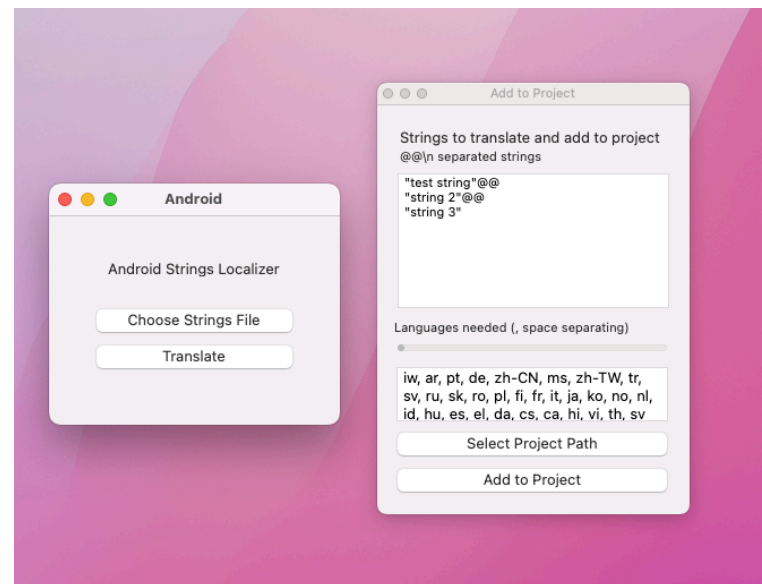
There are 4 new features in LocalizeTranslator 2023 that were not available in the Mac App Store version, LocalizeTranslator 2018.

- Android Strings Localization
- Simple Strings Localization (make many localized versions of a txt file)
- Website Localization (localize a static HTML file into many localized HTML files)
- Get Token (a window in LocalizeTranslator to skip the command line when fetching an API token)

## Android Strings Localization

This feature was made as simple as possible. There are only two buttons on this window: “Choose Strings File” and “Translate.” The other necessary components to the Android translator can be found on the “Add to Project” window. This window appears from the Tools menu to the “Localize and Add Strings to Project” option. On this window, you set the path to the Android project folder (“Select Project Path” button) and you select the languages you would like to translate the strings into (“Languages needed”).

Once you have set your Android project path and selected the language codes you want to translate with, go back to the “Android Strings Localizer” window from the Tools menu. Choose your main strings xml file. Next, click



“Translate.” Your main strings xml file has now been translated to many xml files in other languages, each with a containing folder for the Android project’s indexing.

## **Simple Strings Localization**

Go to the Tools menu and choose the Custom File Localizer option. Here, simply select any .txt file, set the language codes you want to translate into, and click the Translate and Export button. A dialog will ask you what folder to save your translations to, and a new .ltx file will show up for every language code you selected with the language code as the name for the file.

## **Website Localization**

This is the hardest-to-use tool in LocalizeTranslator. It was designed to take a static HTML file, for static websites and translate it into many different localized HTML files. Apache server has a mod to serve a specific HTML file for the language of the client. This means that for my index.html, I can use the Website Localizer to make many index.html files, each with an additional extension for the Apache language code.

To use this tool, go to the Tools menu, and choose Website Translation. From here, select the path of your static HTML files. To use this tool, for simplicity, make a new folder, and for each HTML file you want to translate, make a folder in this new folder for each one of the files. Next, for each folder for each file, click on each listed HTML file in the list of HTML files on the Web Page Translator screen. For each file that you click on, click the Choose Mapping Files Path button and select the folder that you made for each HTML file. When the web page is translated, all of the different language translations for the index.html file will go in the index.html Mapping Files folder.

For each HTML file, not only will there be a Mapping Files folder, but there will also be a Terms file. The Terms file contains a list of (:\\n)-separated strings. You can extract these strings manually, or you can click the Define Terms button for each HTML file you want to translate.

For example, in your index.html file, to describe all the strings that need to be translated, this list might contain the top bar menu of the website and the title of each menu item, the footer of the website and each the text of each link in the footer, each header section on the page, each paragraph section on the page, and so on. Take each one of these phrases and add it to a .txt file with each phrase separated by “:\\n”

Next, add this .txt file into the Mapping Files folder for the HTML file that the .txt file contains the terms for. Use the Simple Strings Translator to translate this .txt file into many .ltx files that are all saved in the Mapping Files folder. Once each HTML file has a terms file and translated versions of the terms file, now you can use the Save Translated Files button in Web Page Translator. This will take each HTML file and each terms translation file and mix them all together to make many translated HTML files for each specified HTML file. Each translated HTML file will contain the Apache server file extension for the language mod.

Support Email: [chris@playr.us](mailto:chris@playr.us)