Translator's Manual

In this project you will be translating subtitles, which are then submitted to TMI's YouTube channel.

Before starting it is advisable to make you aware of the whole process so you would be able to deliver translations of high quality.

The original subtitle files you will receive are generated automatically by advanced speech to text conversion software. This software is able to learn from processed videos and deliver accurate transcripts. But, it does not analyze the semantics and therefore, depending on speaker's pronunciation and expression, sometimes unexpected punctuation or wrongly recognized words may occur. These are very rare situations, but you have to be aware of that and be ready to confront such questionable elements with the original video to find the correct form.
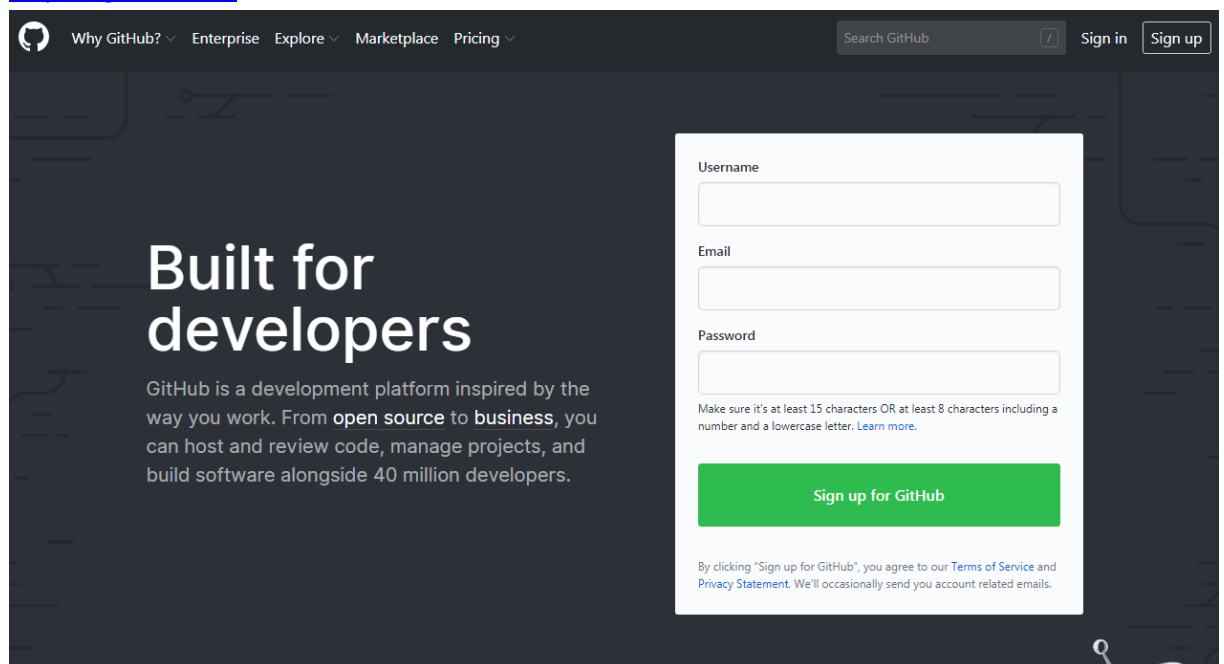
You will be using specialized CAT (*Computer Aided Translation*) software, which will facilitate the translation process and preserve the structure of translated files. This software also has the ability of learning from already translated phrases, so the more you translate the more the software will be able to provide reliable automated translations.

The subtitle files you will work on cannot be modified manually since this can damage their structure and so make them unusable for YouTube. This is another reason why the CAT software is used. It deals with these specific files preserving their format and presents only the text to be translated.

Prepare your working environment

1. First of all sign up for GitHub – the repository we will be working with at:
   https://github.com

   Go through registration process – it is very easy and quick. Don't forget to click the confirmation link, that you will obtain in an e-mail from GitHub. Then contact your Project

Manager and provide him/her your username or e-mail address you used for signing up for GitHub. Having this information the Project Manager will add you to the translation project.

2. Spend a while on exploring your GithHub account making initial settings like your name and avatar. It will be easier to communicate within the team then, when team members will use real names instead of GitHub usernames.
In order to do so, locate your avatar icon in the right top corner and click on that. Then, click on the *Settings* item from the dropdown menu.
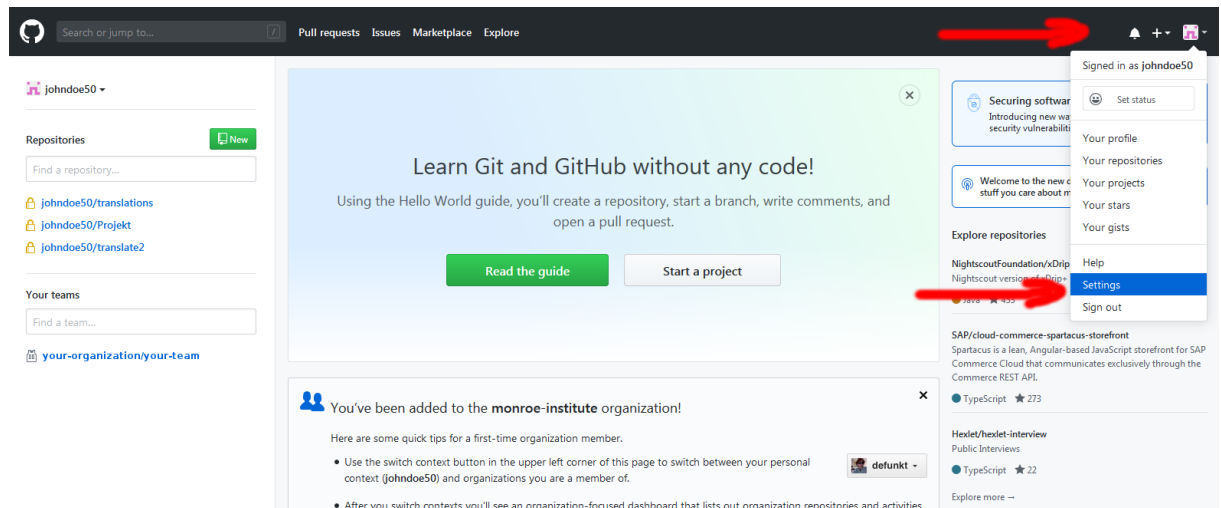
You will be moved to your account settings. Fill up *Name* field. If you want upload your photo and provide other information like your bio, URL, etc., do it as well. Having entered all necessary information scroll down the page to green button *Update profile* and confirm your changes.
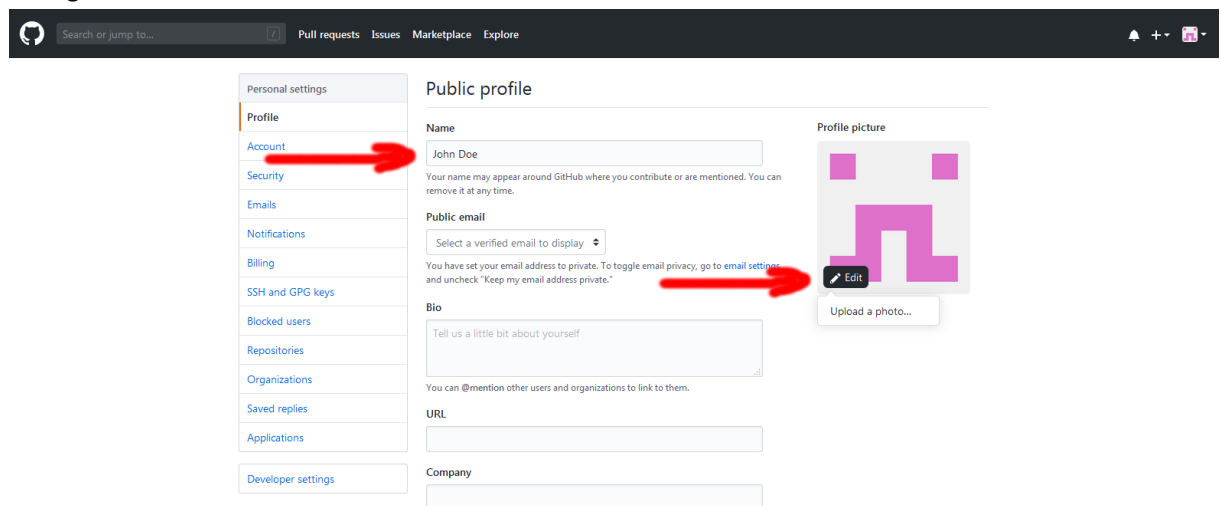
3. A good idea is to elevate the security of your account. So, click the *Security* link on the left pane and then locate a green button labeled *Enable two factor authentication*. Click on it and follow the instructions.

4. Download **OmegaT** software from:
https://omegat.org/download
Choose the version you need depending on the operating system installed on your computer. The *OmegaT* software is a Java application so *Java Runtime Environment (JRE)* should be installed on your computer as well. If you are not sure if you have JRE installed on your

machine just download the *OmegaT* installer with JRE. If you still have a problem deciding which version you should download use the *Download selector* link available on the top of the download page.

Caution: If you do not use English version of Operating System (especially Windows), you will see mixed content (English + your Operating System language) in some of dialog boxes to be displayed.

5. Install *OmegaT* on your machine.

6. Run *OmegaT*. The program will open with the interface in your operating system default language.

7. Go to menu *File->Download team file…* The *Team Project* dialog box will be presented to you:
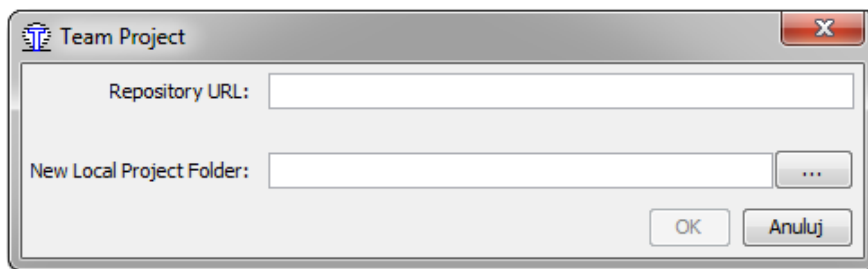
8. Put the repository path that you have received from your Project Manager into the *Repository URL* text field.
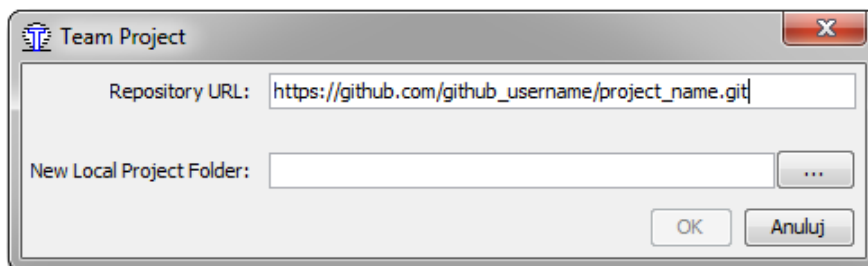

9. You will be asked for *Authentication* details (username and password) then click *OK*. Enter your GitHub username and password.
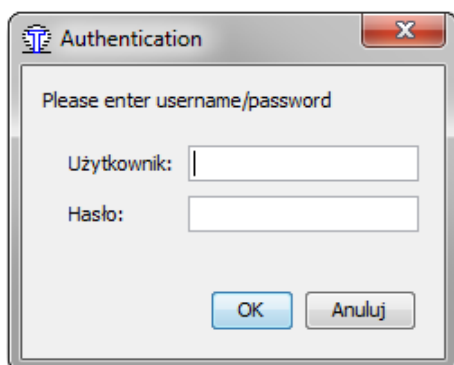

10. Click on the button with three dots in the *Team Project* dialog box and select a location for project files on your computer. It may be within your *Documents* folder in which you created the *Translations* subfolder before.

Tip: After choosing the folder (*Translations* as in the example above) and before saving your choice, at the end of location path in *Folder name* line, type the backslash character and then type the specific name of your project, for example *Bob interview*.
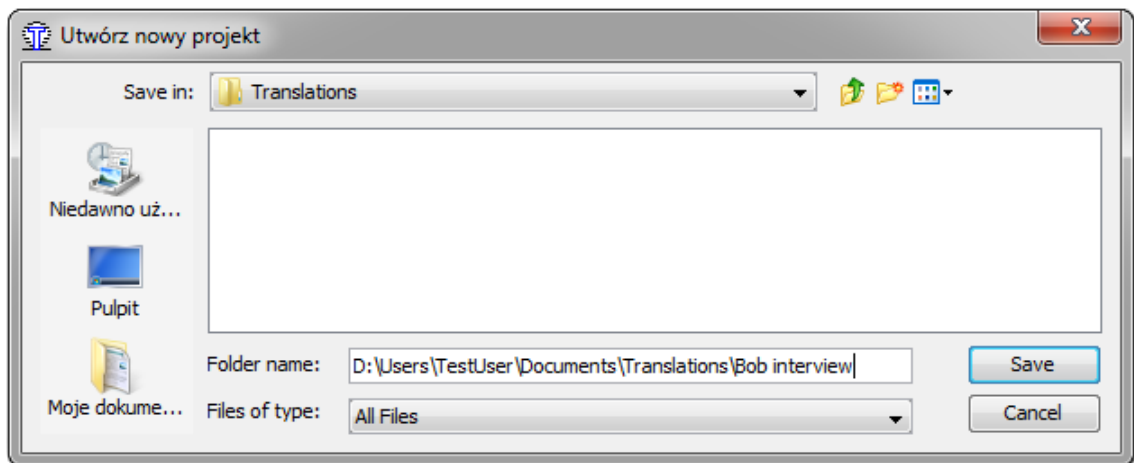
11. Click the *Save* button. All the information for setting a new Team Project has just been entered. You can click the *OK* button in *Team Project* dialog box.
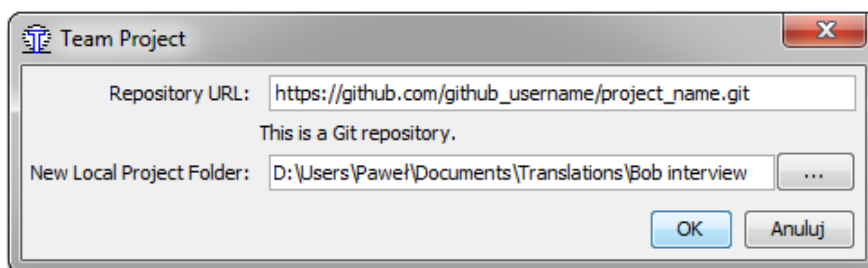
12. On the next screen, just select the file you want to work on and click the *Close* button.
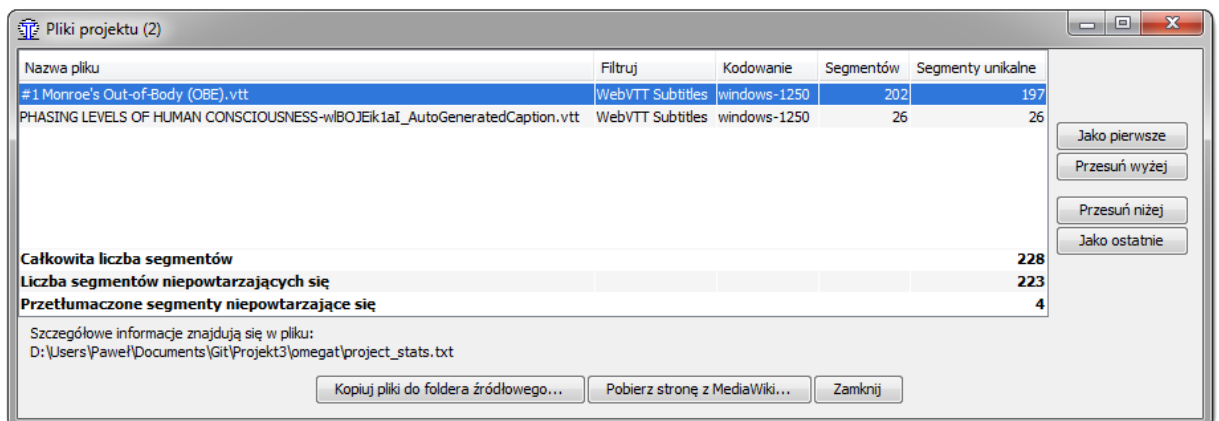
13. Your setup is complete. Now, you can close the project from the menu *File.*

Learn OmegaT

With the application running and no project open a *Quick Start* tutorial is displayed in the left panel. Read it carefully. Also watch following videos that will familiarize you better with the *OmegaT* application:

1. https://youtu.be/WMXr6aJTpcU

Start translating with OmegaT

To start working on a recently configured *Team Project,* click on menu *File* and then *Open recent project*.

You will be working in a multiuser environment with a central repository of files. So, each time you finish your work, *OmegaT* will commit your work to the repository (you can do it manually as well: *File->Commit source files* and *File->Commit target files*). During committing, the application checks if any differences between your local file copy and the repository file does exist. When someone else has worked on the same file at the same time you will see the *Merge* resolving screen. This is standard behavior of such a type of working environment that prevents losing changes made in files. This screen will present you differences between the most recent copy of the file you are committing and the repository file. Just decide which version of the translated phrase should be kept.