



Yantra Launcher translator's guide

Index

[Step 1: Communicate your intentions](#)

[Step 2: Create a GitHub Account](#)

[Step 3: Fork the Yantra Launcher Project](#)

[How do I fork a project?](#)

[Where to find your copy?](#)

[Step 4a: Check if Git is installed](#)

[On Windows 🪟:](#)

[On macOS 🍏:](#)

[On Linux 🐧:](#)

[Step 4b: Installing Git](#)

[On Windows 🪟:](#)

[On macOS 🍏:](#)

[On Linux 🐧:](#)

[Debian/Ubuntu-based distributions](#)

[Fedora](#)

[CentOS or RHEL](#)

[Verifying Installation](#)

[Step 5: Clone Your Forked Repository](#)

[Find Your Repository URL](#)

[Cloning the Repository](#)

[Verify the Cloning Process](#)

[Step 6: Add a New Language Translation](#)

[Step 7: Resolve conflicts](#)

[How to manage and resolve conflicts](#)

[Syncing Your Fork: If there are updates in the original project](#)

[Handling Conflicts](#)

Best Practice

Step 8: Translate Correctly the Strings

Step 9: Push Changes and Open a Pull Request

Disclaimer

Commit Your Changes:

Push Your Changes:

Open a Pull Request:

Step 1: Communicate your intentions

Before diving into translating the Yantra Launcher to your language, it's important to let the community know about your plans. This helps in coordinating efforts and ensures that your work complements the project's needs without duplicating efforts.



1. **Join the [Community Discord](#):** If you haven't already, join the Yantra Launcher community on Discord. This platform is where developers and contributors collaborate and share updates about the project.
2. **Introduce Yourself:** Once in the Discord, find the general text channel. Introduce yourself briefly, mentioning your interest in contributing to the Yantra Launcher by translating it into your language.
3. **Communicate Your Intentions:** Share your intentions with the community. Mention that you plan to translate the Yantra Launcher and ask if anyone else is working on the same language. This is also a good time to inquire about any guidelines or requirements for translations.

4. **Collaborate:** Be open to collaborating with others. If someone is already working on translations for your language, ask if you can help. If not, let others know how they can support your efforts.

NOTE: If you don't want to use Discord for whatever reasons, implement steps 2-4 by opening an issue on the [Github Repository](#) of Yantra Launcher. The contributors will collaborate with you keeping the community in mind!

Before you embark on the translation journey, remember that working in isolation without communicating with the community or following established best practices can lead to unintended outcomes. Specifically, there's a risk that your hard work might not align with the project's standards or directions, which could result in your contributions being declined or not integrated into the repository.

To avoid such scenarios, we strongly encourage you to stay engaged with the community, seek feedback, and adhere to any provided guidelines or standards for translation. This collaborative approach not only enriches your contribution but also enhances the project's quality and coherence.

By working together, following the project's best practices, and keeping open lines of communication, we can ensure that your valuable contributions are recognized, appreciated, and incorporated, making the Yantra Launcher more accessible to users worldwide.

Remember that the community of Yantra Launcher is strongly supportive and grateful for any kind of contribution you want to provide to the project and no question is too stupid or obvious to ask to the ones that are more experienced: Don't be afraid!

Step 2: Create a GitHub Account



If you do not already have a GitHub account, you will need to create one to contribute to the project.

Go to [GitHub's website](#).

Click on the "Sign up" button and follow the instructions to create your account.

Step 3: Fork the Yantra Launcher Project

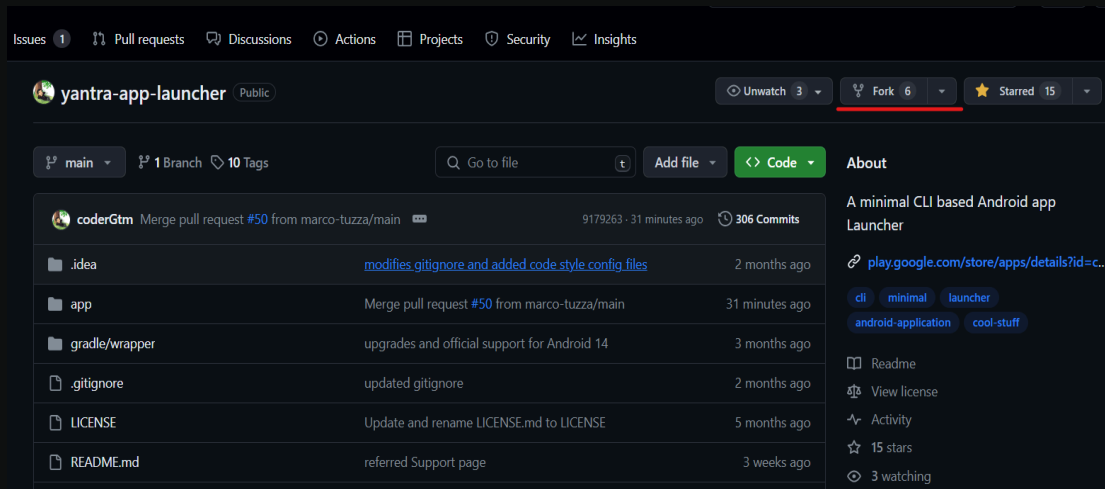


Forking a project creates a copy of the project in your GitHub account, allowing you to make changes without affecting the original project.

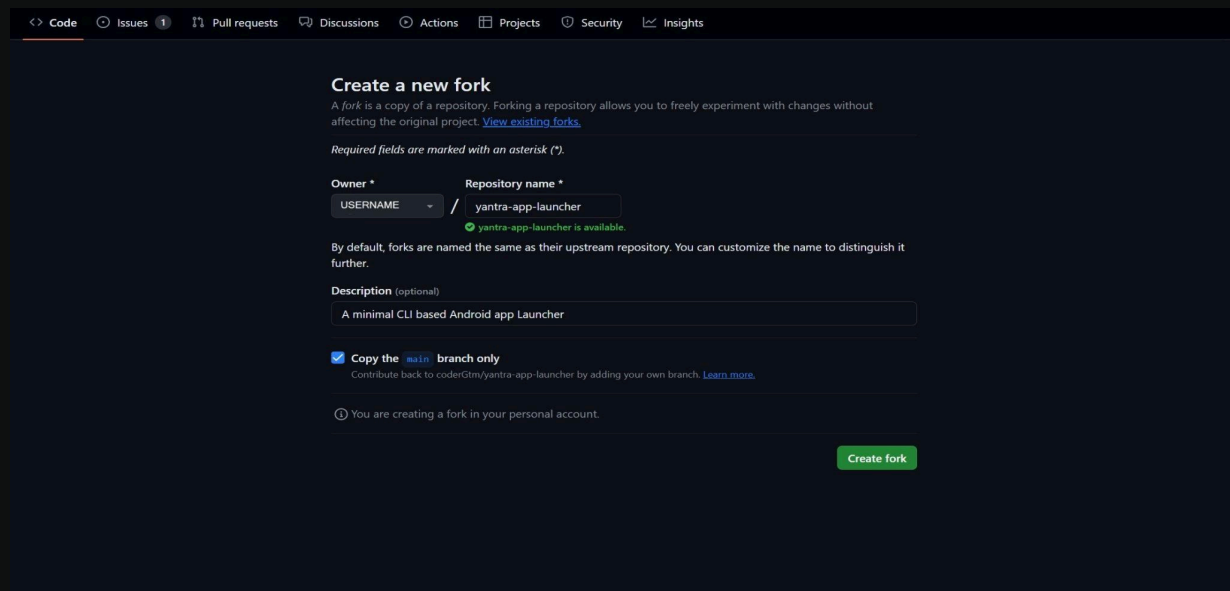
How do I fork a project?

1. Navigate to the [GitHub page of the project](#).

2. Click on the "Fork" button in the upper right corner of the page.



3. A page titled 'Create a new fork' will appear in front of you. You can set the parameters as follows.
- Owner: Your username
 - Repository name: yantra-app-launcher
 - Description: The text that will be already present
 - Copy the *main* branch only: Selected

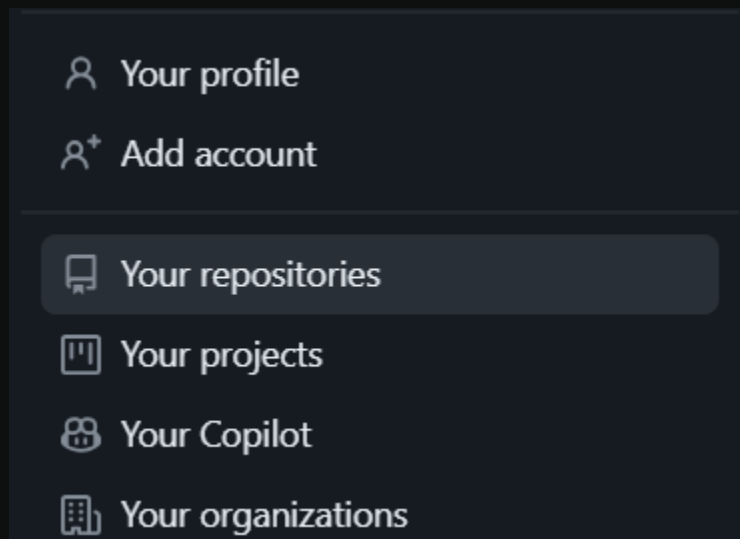


Congratulations on forking a project! This means you've created a personal copy of the original project, allowing you to explore its content, make changes, and experiment without affecting the main project.

Where to find your copy?

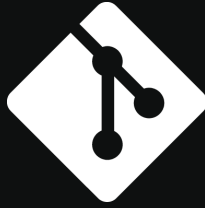
After performing the fork, the copy of the project will be located directly within your repositories on GitHub. To view it, follow these simple steps:

1. Log in to your GitHub account (if you're not).
2. Click on your profile picture in the top right corner of the screen and select "Your repositories" from the dropdown menu. This will take you to a page listing all your repositories.



3. Here, you will find the list of projects you have, including the new copy of the project you just forked. The name of the forked repository will be the same as the original project, but now it resides under your user profile.
4. Now that the project is in your repositories, you are free to explore and modify it as you wish, in this specific case, to include the translation in the language you want to integrate in the launcher.

Step 4a: Check if Git is installed



Before we proceed, let's check if Git is already installed on your machine. Here's how you can do it:

On Windows 🪟:

1. Open the Start menu and type "cmd" to open the Command Prompt.
2. In the Command Prompt, type `git --version` and press Enter.
3. If Git is installed, you will see the version number displayed. If you receive a message saying something like "git is not recognized as an internal or external command", Git is not installed.

On macOS 🍏:

1. Open the Terminal application. You can find it using Spotlight with **Cmd + Space** and typing "Terminal".
2. Type `git --version` and press Enter.
3. If Git is installed, the version number will be displayed. If the terminal says that git is not installed, you might be prompted to install it.

On Linux 🐧:

1. Open a terminal window.
2. Type `git --version` and press Enter.
3. If Git is installed, you'll see the version number. If not, you'll see a message indicating that Git is not installed.

Step 4b: Installing Git

If Git is not installed on your machine, follow the instructions below for your operating system:

On Windows 🪟:

1. Go to the official [Git website](#).
2. Click on "Download" for Windows. The download will start automatically.
3. Once downloaded, open the installer and follow the instructions to install Git. The default settings are usually fine for most users.

On macOS 🍏:

1. Go to the official [Git website](#) and download the macOS version.
2. Open the downloaded file and follow the installation instructions.

Alternatively, if you have [Homebrew](#) installed, you can open a terminal and run `brew install git`.

On Linux 🐧:

Debian/Ubuntu-based distributions

```
sudo apt update
sudo apt install git
```

Fedora

```
sudo dnf install git
```

CentOS or RHEL

```
sudo yum install git
```

Verifying Installation

After installation, reopen your terminal or command prompt, and type `git --version` to ensure Git was installed correctly. You should now see the version number displayed.

Now that you have Git installed, you're ready to start using it for version control and collaboration on your projects!

Step 5: Clone Your Forked Repository



After forking a repository on GitHub, the next step is to create a local clone of your fork. This allows you to work on the project from your own computer.

Find Your Repository URL

First, you need to find the URL of your forked repository on GitHub:

1. Navigate to your forked repository on GitHub.
2. Look for the "Code" button near the top of the page. Click on it.
3. You'll see a URL (it should look something like `https://github.com/yourusername/repositoryname.git`). You can click the clipboard icon to copy this URL.

Cloning the Repository

Next, you'll use Git from your command line or terminal to clone the repository:

1. Open your command line (Command Prompt on Windows, Terminal on macOS and Linux).
2. Navigate to the directory where you want to clone the repository.
3. Type the following command and replace the URL with the one you just copied:

```
git clone https://github.com/yourusername/yantra-app-launcher
```

4. Press Enter. Git will start cloning the repository to your local machine.

Verify the Cloning Process

After the cloning process completes, navigate into the cloned directory. It should appear like this:

Name	Status	Date modified	Type	Size
.git		03/03/2024 11:41	File folder	
.idea		03/03/2024 11:41	File folder	
app		03/03/2024 11:41	File folder	
gradle		03/03/2024 11:41	File folder	
.gitignore		03/03/2024 11:41	Git Ignore Source File	1 KB
build.gradle		03/03/2024 11:41	Gradle Source File	1 KB
gradle.properties		03/03/2024 11:41	Properties Source File	2 KB
gradlew		03/03/2024 11:41	File	6 KB
gradlew.bat		03/03/2024 11:41	Windows Batch File	3 KB
LICENSE		03/03/2024 11:41	File	3 KB
README.md		03/03/2024 11:41	Markdown Source File	2 KB
settings.gradle		03/03/2024 11:41	Gradle Source File	1 KB
support.md		03/03/2024 11:41	Markdown Source File	1 KB

Step 6: Add a New Language Translation



Once the project is cloned in your computer open it in your file explorer and navigate to `yantra-app-launcher\app\src\main\res`

Here you will find the “values” directories, containing the strings translated in the supported languages.

IF AND ONLY IF the values-language directory of your language is not present, follow these steps:

1. Inside the directory `yantra-app-launcher\app\src\main\res` add a new directory with the name “values-[ISO 639 language code of your language](#)”

Ex:

- Italian: values-it
- German: values-de
- Portugues: values-pt

2. Inside the new directory create an empty file called “strings.xml”
3. Copy what’s inside the file `yantra-app-launcher\app\src\main\res\values\strings.xml` and paste it inside the new empty file you just created so now your file will be filled with all the original english strings ready to be translated in your language.

Step 7: Resolve conflicts



Before you start translating the Yantra Launcher into your language, it's essential to ensure that your forked project is up-to-date with the original project, without any conflicts. Conflicts can occur when

the original project has been updated since you made your fork, and your changes clash with the new updates.

How to manage and resolve conflicts

Check for Conflicts: Regularly check your forked repository for any potential conflicts with the original project. GitHub provides tools and indicators to help you identify when your fork is behind or has conflicts with the base repository.

Syncing Your Fork: If there are updates in the original project

1. Navigate to your fork on GitHub.
2. Look for the "Fetch upstream" option. Click on it, then select "Fetch and merge" to sync your fork with the original repository. This action tries to merge the latest changes from the original repository into your fork.

Handling Conflicts

1. If the automatic merge has conflicts that GitHub cannot resolve, you will need to resolve these manually. GitHub will guide you through the process of reviewing the conflicts.
2. In some cases, if you haven't started translating or your changes are not significant, you might find it easier to discard your changes. You can do this by selecting "Discard changes" during the conflict resolution process. This will revert your files to match the original repository's latest version.
3. Alternatively, if you need to keep your changes, you will have to manually edit the files to resolve the conflicts. After editing, commit your changes to finalize the resolution process.

Best Practice

Always ensure your fork is synchronized with the original repository before you start working on translations. This minimizes the risk of conflicts and simplifies the process of integrating your work later on.

Once your fork is up-to-date and free of conflicts, you're ready to proceed with translating the Yantra Launcher into your language, confident that your contributions will smoothly integrate with the project.

Remember: Regular communication with the community and adherence to best practices are key to a successful contribution. By keeping your fork synchronized and resolving any conflicts promptly, you ensure that your efforts are aligned with the project's overall direction and contribute to a seamless collaborative environment.

Step 8: Translate Correctly the Strings



1. **Modify Only the Text Within `<string>` Tags:** The content to be translated is contained within the `<string>` tags. Do not alter the name attribute or any part of the XML structure outside the text intended for translation.

```
<string name="username_prefix">Username prefix</string>
```

This is the only part that you must edit

2. **Do Not Translate Command Names:** Commands such as settings, help, or any other command name mentioned within the strings should remain in English to maintain the integrity of the code.

```
<string name="cmd_launch_title">launch [app name]</string>
```

Do not translate

Can be translated to enhance clarity and comprehension

3. **Preserve the Meaning and Tone:** Your translation should convey the original message's meaning as closely as possible. Keep the tone (formal, informal, technical) consistent with the source text.

English:

```
<string name="enjoy_with_face">Enjoy 😊👍</string>
```

Italian:

```
<string name="enjoy_with_face">Divertiti 😊👍</string>
```

4. **Keep Examples Unchanged:** If the original string includes examples, especially command usage examples, keep these examples exactly as they are in English, but with the needed translations. This ensures that the commands remain universally understandable and functional.

English:

```
\nExample: \'vibe 1000\' does a 1 second vibration.
```

Italian:

```
\nEsempio: \'vibe 1000\' per una vibrazione di 1 secondo.
```

5. **Escape Special Characters:** Pay attention to special characters that need to be escaped, such as apostrophes ('). In XML, you should escape an apostrophe with a backslash (\') to prevent parsing errors.

```
<string name="locating_app">Locating \'%1$s\'...</string>
```

6. **Preserve Placeholders and Special Formatting:** Often, strings contain placeholders (e.g., %1\$s, %2\$d) or special formatting tags (e.g., <![CDATA[...]]>). These should remain unchanged to ensure the software functions correctly.

```
<string name="weather_min_c_f"><![CDATA[=> Min: %1$d°C  
(%2$d°F)]]></string>
```

```
<string name="locating_app">Locating \'%1$s\'...</string>
```

Do not change

7. **Be Mindful of Context:** Understand the context in which the string is used within the application. This can significantly affect the translation choice. If possible, consult the application or screenshots to see how strings are presented.
8. **Consistency is Key:** Ensure consistency in terminology and style throughout your translation. This may involve creating a glossary or referring to an existing one if available.
9. **Seek Feedback:** Don't hesitate to consult with other translators or native speakers for feedback on your translations, especially for nuanced or technical phrases, we are here to support you if you have any doubt.

Step 9: Push Changes and Open a Pull Request



After translating the strings, you'll need to push your changes to your forked repository and open a Pull Request (PR) to the original project. Here's a simplified guide for users not proficient with Git:

Disclaimer

Navigating Git and GitHub might feel daunting, especially for those not familiar with coding or version control systems. The process of committing changes, pushing them, and opening a Pull Request can be challenging on the first attempt. If you find yourself struggling or have any doubts, please don't hesitate to seek help on the Yantra Launcher community Discord server. Our community is here to support you through this learning curve, so feel free to ask for assistance or clarification on any step.

Commit Your Changes:

1. Open your terminal or command prompt.
2. Navigate to your project's directory.
3. Use `git add .` to stage your changes.
4. Commit the changes with `git commit -m "Translation for strings.xml in [Your Language]"`, replacing [Your Language] with the actual language you translated to.

Push Your Changes:

1. Push the commit to your fork with `git push origin master` (assuming you are working on the master branch).

Open a Pull Request:

1. Go to your Yantra Launcher repository on GitHub.
2. You'll likely see a prompt to open a Pull Request if your fork has recent changes. Click on "Compare & pull request".
3. Add a title and description for your Pull Request. Clearly state that you've added a translation and mention any other contributors if you collaborated.
4. Submit the Pull Request.
5. After submitting your Pull Request, stay engaged by monitoring any feedback from the project's maintainers. Should they request changes, it's important to respond promptly and make the necessary edits to your submission.



Done! If your PR gets accepted it means you successfully translated the launcher in another language.

Remember: It's crucial to stay updated with the original project and resolve any conflicts before opening a Pull Request. This ensures a smoother integration process for your contributions.

Thanks for considering translating this project!

