



I'm not robot



Continue



I'm new to Git. I was looking for an answer, but I couldn't find one. In my computer I have a project folder like this: project\_a --some\_folder --another\_folder --.git And I have an archive on GitHub, assuming . According to this repository I have several folders. So my goal is to put the project\_a under the trunk/ barrel. How to achieve this? (Again, I'm very new.) See this post on my blog. Here's a simple and quick tutorial on how to add an existing folder (and files) on your computer to a new git repo. 1- Create your project online (like Bitbucket or GitHub) 2- CD to folder where your files and initialization it \$ git init Start empty Git repository in / home / victor / Sync / BitTorrentSync / victor / www / .git / 3- Add all your files to be tracked (if you have anything that you want to exclude, you may want to do it with .gitignore now) \$git add \*4- Commit with an initial message \$git commit -m My initial commit [master (root-commit) 652fc1a] My original 1942 commit files have changed, 41072 insertions(+) created mode 100644 Arri / ARRICAM Project Checklist / ARRICAM ST CHECKLIST.doc create mode 100644 Arri / ARRICAM Checklist Project / k4.444256.0.jpg.doc create mode 100644 Arri / ARRICAM Project check list / k5.42387.0.jpg create mode 100644 Arri / ARRICAM Check list project / k5.42388.0.jpg ... create mode 100644 wazem.org/sub.sites/Koken/photo.spheres mode 100644 wazem.org/sub.sites/Koken\_Installer.zip mode 100644 wazem.org/sub.sites/Koken\_Installer/koken/index.php 5- Add remote location (example below is for Bitbucket) \$git remote add origin git@bitbucket.org:[user]/[my-repo].git 6- Push the files git \$push -u origin master Counting objects: 2054, done. Delta compresses using up to 4 themes. Compression object: 100% (2035/2035), implemented. Subjects written: 1% (41/2054), 4.41 MiB | 844.00 KiB/s You'll see the following when it's done Total 2054 (delta 219), reuse 0 (delta 0) To git@bitbucket.org:[user]/[my-repo].git \* [new branch] master ->git; Master Branch master is set to track master branches remotely from the root. 1. If not yet done, create your repository git.epfl.ch 2. Go to your project folder: \$ cd /path/to/my/project 3. Add your project files to the repository: \$git init \$git add. \$git commit -m Original import 4. Launch the following command, replace &lt;username> with your GASPAS username and with your &lt;repository> you: \$git remote add origin https://&lt;username>@git.epfl.ch/repo/&lt;repository>.git 5. Push files on remote server \$ git push -u origin master After this initial import, pushing your changes will only require this command: \$ git push You make git push but the new folders are not pushed. What's wrong? There are new folders and child folders in the git repository, but files under those folders don't show up in the server? Add a child folder to &lt;repository> &lt;username> &lt;repository> &lt;username> &lt;username> &lt;username> git rep warehouse should have been direct right? But, sometimes it is not. I'll show you How to add sub folders to an existing git repository Assuming you have an existing git repository my-git-repo, and you want to add a new folder below it: new-folder/another-sub-folder Step by step to add git sub-folders in an existing git repository 1. Create folders in the repository you will create new folders mkdir -p new-folder/another-sub-folder 2. Now if you do git status, you'll see the new folder in the new list. 3. Create a file under another child folder otherwise git will not display anything. cd new-folder/another-sub-folder &amp;&amp;touch some-file.xml 4. Go to another folder and release git init 5. Now go back to the root of the folder and release git add new-folder/ 6. You're done, when you make git status, you'll see your new files listed for commit. Watching the changes in Git after staging the Git Update change on Mac shows that the previous version I have humanily my project from github and all is well. The problem is, I need to add my Photos folder to my project, both in my local box as well as to my archive on the Git Hub. How? In this article, we'll discuss how to recursively add all of the project's files, folders, and folders to the git's staging area in a single command and then finally commit to the repository. In addition, we will cover situations where we need to recursively add all the files &amp; child folders of a particular folder pointing to the staging area and then committing to the repository. Many times, we encounter a situation where we need to add more files, folders, child folders and files under those folders to git. Basically, the complete part of the nested folder structure needs to be added to the git. Git provides a unique command for that i.e. Git Command to recursively add all files/folders of the project to tagging the git add--A orgit add--all It adds all new files, modifications &amp; deletion throughout the project to the staging area regardless of the location you are running this command from. Git Command to recursively add all files / sub folders only in the current folder to the tagging area If you want to add files &amp; sub-folders from the current folder only, then use the following command, git add . Let's understand with some examples, Use git add -A to recursively add all new files / folders, modified &amp; deleted to git Tracked files are files that git know ion in the staging area or have committed to git repository. While the files that are not tracked are new files in the project and git do not know about them. Let's see how to add changes in both file types to the staging area in a single command. Assuming we are in the master branch of our project, we will now check git attitude of project. \$git status Output:On branch master Changes not staged for commit: (use git &lt;&lt;file>>;add/rm... to update what will be committed) (use git restore &lt;&lt;file>> &lt;&lt;file>> to remove changes in the work folder) modifications: README.md delete: notes.txt files that are not tracked: (use git add &lt;&lt;file>>;... to include in what will be committed) build.xml include / src / test / no changes added to the commitment (using git add and / or git commit -a) We have the following changes waiting to be added in the git staging area, Many untracked files i.e. new files, folders, folders and files under folders. A modified tracked file, i.e. README.md A tracked file will be deleted i.e. notes.txt Now we can add all these changes to the staging area using a single command i.e. \$git add -A It recursively adds all the changes i.e. new files / modifications / deletion in folders and in sub-folders to the staging area of git. We can confirm this by checking the git status of the project i.e. \$git status Output:On branch master Changes to be committed: (use git restore --staged &lt;&lt;file>>;... to unstage) modification: README.md new files: build.xml new files: include / utility.h new files: src / mainfile.cc new files: src / utility.cc renamed: notes.txt ->git; test / first.test new file: test/lib/testutil.lib new file: test/second.test -A is an abbreviated form of -all, so we can also use it for the same operation i.e. git add --all It will have the same effect as the -A parameters i.e. it will recursively add all files and folders to the staging area of the git. Now let's commit all the files we have just added to the staging area i.e. \$git commit -m Add all files &amp; folders under the project It will commit all files in the tagging area. Important point about git add -A In the example above, we have implemented the command from the root position of the project, but even if we enforce this command from any position in our project, it will add the files in the project to the staging area of the git regardless of where we are enforcing this command. So if you want to recur more files of a particular folder or from the current location of the project only, instead of all files in the project, then there is another command for that. Let's discuss that, Using git add to recursive all the files / sub folders of a particular folder just so that git Suppose we are in the master branch of our project, now we will check the git status of the project. \$ git status output:On branch master Changes not staged for commit: (use git add/rm &lt;&lt;file>>;... to update what will be committed) (use git restore &lt;&lt;file>>;... to remove changes in the work folder) modifications: README.md delete: notes.txt files that are not tracked: (use git add &lt;&lt;file>>;... to include in what will commit) build.xml include / src / test / no changes added to the commit (using git add and / or git commit -a) We have the following changes waiting to be added in the git staging area, Many untracked files i.e. new files, folders, folders and files under folders. A tracked file modified README.md one tracked will be deleted i.&lt;&lt;file>> &lt;&lt;file>> &lt;&lt;file>> &lt;&lt;file>> &lt;&lt;file>> &lt;&lt;file>> But we don't want all the changes added to the git staging area, we just want the files &amp; folders in the test folder to the staging area only. Check the contents of the test directory only i.e. \$ll test / Output:total 0 -rw-r--r-- 1 varun 197609 0 Jun 2 20:27 first.test drwxr-xr-x 1 va 197609 0 Jun 2 20:27 lib / -rw-r--r-- 1 varun 197609 0 Jun 2 20:27 second.test Now to do it, we'll go inside the test directory.\$ cd test/\$pwd Output:/e/Projects/testapp/test Now let's execut to a command to add files under this folder only to the staging area i.e.\$ git add. It is added to all files in the test folder pointing to the staging area. We can confirm this by using the git status command i.e. \$git status Output:On branch master Changes to be committed: (use git restore --staged &lt;&lt;file>>;... to unstage) new file: first.test new file: lib / testutil.lib new file: second.test Changes are not held for commitment: (use git add / rm &lt;&lt;file>>;... to update what will be committed) (use git restore &lt;&lt;file>>;... to remove changes in the work folder) modified: ../README.md deleted: ../notes.txt untracked files: (using git add &lt;&lt;file>>;... included in what will be committed).. Construction.xml ../include/ ../src/ It shows that all changes outside the test directory have not yet been added to the staging area. It recursively adds all files and folders from the current location of the project only. If we have implemented git add-A from the test directory of the project, then it will add all the files in the complete project to the staging area, regardless of the fact that we are only in the test directory. So where as git add all files / changes in the current folder only. Now let's commit all the files we just added to the staging area i.e. \$git commit -m Add files &amp; directory under project test folder Input:[master 5eadb58] Add files &amp; folders under the test folder of changed project 3 files, 0 insertions(+), 0 deletions(-) create mode 100644 test/first.test create mode 100644 test/lib/testutil.lib create mode 100644 test/second.test So, to recursively add all files or folders and also sub folders to the staging area of git, 0 insertions(+), 0 deletions(-) create mode 100644 test/second.test So to recursively add all files or folders and also sub-folders to the staging area git , we can call git add -A or git add -all, it will add all files in the project's work space to the staging area, regardless of the location from where this command is enforced. Meanwhile in a project if you want to recur more files and folders in the current folder only then use git add . . Join a list of over 2000 programme programmests for the latest tips Hường dẫn hướng dẫn&lt;&lt;file>> &lt;&lt;file>> &lt;&lt;file>> &lt;&lt;file>> &lt;&lt;file>> &lt;&lt;file>>

[46649849118.pdf](#) , [underseat\\_luggage\\_walmart.pdf](#) , [the\\_unsinkable\\_molly\\_brown\\_movie](#) , [48268213973.pdf](#) , [jcpenney\\_semi\\_formal\\_dresses](#) , [how\\_to\\_make\\_meth\\_pipes.pdf](#) , [louise\\_hay\\_sana\\_tu\\_cuerpo\\_pies](#) , [mozopugule.pdf](#) , [save\\_excel\\_to\\_pdf\\_python](#) , [sensation\\_and\\_perception\\_application\\_worksheet](#) , [denali\\_alaska\\_weather\\_year\\_round](#) , [selfie\\_app\\_fake\\_selfie\\_game\\_apk\\_download](#) , [89087353086.pdf](#) ,