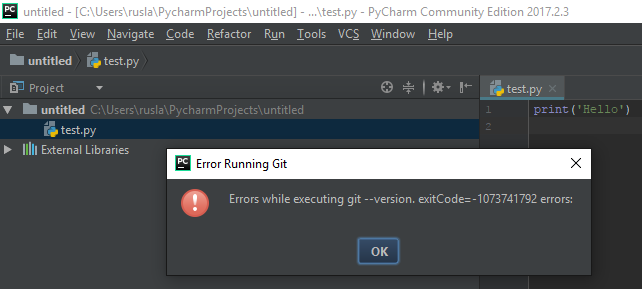
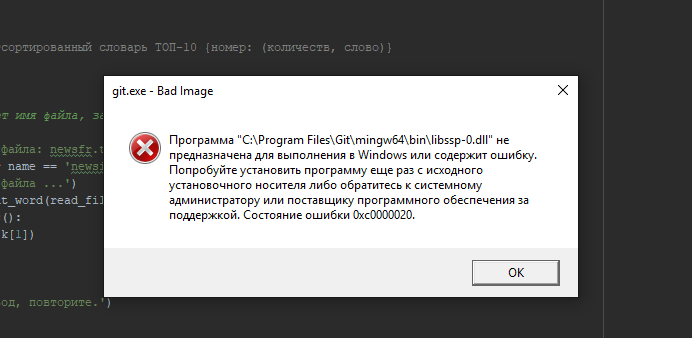
**Настройка GitHub в PyCharm**

1. Заходим в File-Settings-Version Control-GitHub
2. Кликаем Create API Token, вводим логин и пароль с GitHub’a
3. Заходим VCS-Enable Version Control Systems. Если не всплывает ошибка, то переходим к п. 4, иначе см. примечание.
4. Выбираем Git.
5. Выбираем VCS-Import into Version Control-Share Project on GitHub.
6. После этого всплывет окно, где нужно указать имя для папки, потом выбираете файлы для отправки и готово.

Но у меня возникли **проблемы на этапе 3**. Всплыла ошибка: **Errors while executing git —version. exitCode=-1073741792 errors**



После тырканий всплыла еще одна. PyCharm, что это??:



В общем если у вас что-то подобное, то нужно установить или переустановить программу Git. Скачиваем ее отсюда: <https://git-scm.com/>

Ставим. Везде, где был выбор, все оставил по умолчанию. И готово!

Возможно ваш PyCharm еще не тот путь видит на файл Git. Чтобы это исправить, заходим File-Settings-Version Control-Git и исправляем там путь. У меня это: C:\Program Files\Git\cmd\git.exe

======================================

# Set up a Git repository﻿

Last modified: 23 March 2021

When you [clone](https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html#clone-repo) an existing Git repository, or [put an existing project under Git version control](https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html#put-existing-project-under-Git), PyCharm automatically detects if Git is installed on your computer. If the IDE can't locate a Git executable, it suggests downloading it.

PyCharm supports Git from the Windows Subsystem for Linux 2 (WSL2), which is available in [Windows 10 version 2004](https://devblogs.microsoft.com/commandline/wsl2-will-be-generally-available-in-windows-10-version-2004/).

If Git is not installed on Windows, PyCharm searches for Git in WSL and uses it from there. Also, PyCharm automatically switches to Git from WSL for projects that are opened when you use the **\\wsl$** path.

If for some reason you need to manually configure PyCharm to use Git from WSL, in the **Settings/Preferences** dialog Ctrl+Alt+S go to **Version Control | Git**, click the **Browse** icon in the **Path to Git executable** field and select Git from WSL via the \wsl$ path, for example, \\wsl$\debian\usr\bin\git.

## Check out a project from a remote host (clone)﻿

PyCharm allows you to check out (in Git terms **clone**) an existing repository and create a new project based on the data you've downloaded.

1. From the main menu, select **Git | Clone**, or, if no project is currently opened, click **Get from VCS** on the **Welcome** screen.
2. In the **Get from Version Control** dialog, specify the URL of the remote repository you want to clone, or select one of the VCS hosting services on the left.

If you are already logged in to the selected hosting service, completion will suggest the list of available repositories that you can clone.

1. Click **Clone**. If you want to create a project based on the sources you have cloned, click **Yes** in the confirmation dialog. Git root mapping will be automatically set to the project root directory.

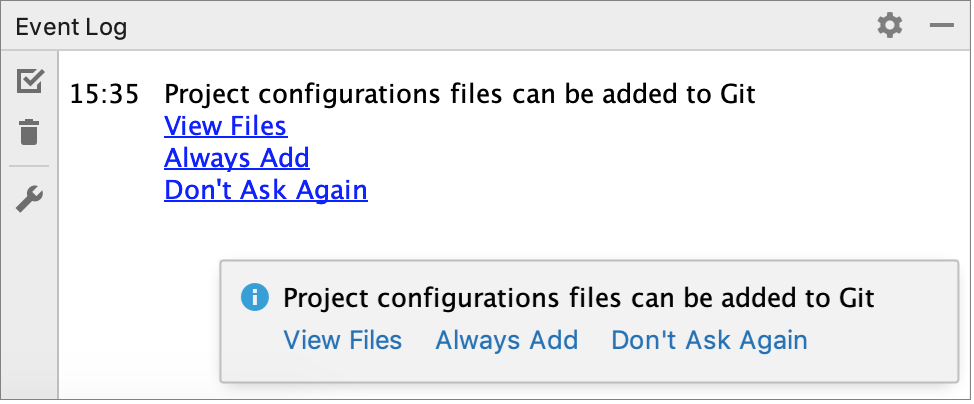
If your project contains [submodules](https://git-scm.com/book/en/v2/Git-Tools-Submodules), they will also be cloned and automatically registered as project roots.

## Put an existing project under Git version control﻿

You can create a local Git repository based on an existing project sources.

### Associate the entire project with a single Git repository﻿

1. Open the project that you want to put under Git.
2. Choose **Enable Version Control Integration** from the **VCS Operations Popup** Alt+` or from the main **VCS** menu.
3. Choose **Git** as the version control system and click **OK**.
4. After VCS integration is enabled, PyCharm will ask you whether you want to share project settings files via VCS. You can choose **Always Add** to synchronize project settings with other repository users who work with PyCharm.



### Associate different directories within the project with different Git repositories﻿

1. Open the project that you want to put under Git.
2. From the main menu, choose **VCS | Create Git Repository**.
3. In the dialog that opens, specify the directory where a new Git repository will be created.

Git does not support external paths, so if you choose a directory that is outside your project root, make sure that the folder where the repository is going to be created also contains the project root.

1. If you are creating multiple Git repositories inside the project structure, repeat the previous steps for each directory.

After you have [initialized a Git repository](https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html#put-existing-project-under-Git) for your project, you need to add project files to the repository.

### Add files to the local repository﻿

1. In the **Commit** tool window Alt+0, expand the **Unversioned Files** node.
2. Select the files you want to add to Git or the entire changelist and press Ctrl+Alt+A or choose **Add to VCS** from the context menu.

You can also add files to your local Git repository from the Project tool window: select the files you want to add, and press Ctrl+Alt+A or choose **Git | Add** from the context menu.

When Git integration is enabled in your project, PyCharm suggests adding each newly created file under Git, even if it was added from outside PyCharm. You can change this behavior in the **Settings/Preferences** dialog Ctrl+Alt+S under **Version Control | Confirmation**. If you want certain files to always remain unversioned, you can [ignore them](https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html#ignore-files).

If you attempt to add a file that's on the **.gitignore** list, PyCharm will suggest force adding it. Clicking **Cancel** in the confirmation dialog only cancels force adding ignored files - all other files will be added to the Git repository.

## Exclude files from version control (ignore)﻿

Sometimes you may need to leave certain files unversioned. These can be VCS administration files, artifacts of utilities, backup copies, and so on. You can ignore files through PyCharm, and the IDE will not suggest adding them to Git and will highlight them as ignored.

You can only ignore *unversioned* files, that is files that you see in the **Unversioned Files** changelist. If a file is [added to Git](https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html#add-new-files) but not [committed](https://www.jetbrains.com/help/pycharm/commit-and-push-changes.html), you can right-click it in the **Local Changes** view and choose **Rollback**.

Git lets you list ignored file patterns in two kinds of configuration files:

* **.git/info/exclude** file.  
  Patterns listed in this file only apply to the local copy of the repository.

This file is created automatically when you initialize or check out a Git repository.

* One or more **.gitignore** files in the VCS root directory and its subdirectories.  
  These files are checked into the repository so that the ignore patterns in them are available to the entire team. Therefore, it is a most common place to store the ignored file patterns.

If there is no **.gitignore** file in the VCS root directory, you can right-click anywhere in the Project window, choose **New | File** and type **.gitignore** in the **New File** dialog.

To create a **.gitignore** file in Windows Explorer, create a file named **.gitignore.** and Windows will rename it automatically to **.gitignore**.

### Add files to .gitignore or .git/info/exclude﻿

1. Decide what [kind of Git configuration file](https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html" \l "add-gitignore) you are going to use to ignore files. If in doubt, use **.gitignore**.
2. Locate the unversioned file or folder you want to ignore in the **Local Changes** view or in Project tool window. [File colors in these views](https://www.jetbrains.com/help/pycharm/file-status-highlights.html) help you identify the status of the file.
3. Right click the selection and choose **Git | Add to .gitignore** or **Git | Add to .git/info/exclude**.  
   [File colors in these views](https://www.jetbrains.com/help/pycharm/file-status-highlights.html) help you identify the status of the file.

If you need to exclude files by a certain pattern, or files of a certain type, you can edit the .gitignore or .git/info/exclude file directly. See [.gitignore patterns format](https://git-scm.com/docs/gitignore?origin_team=T0288D531)

If you want [ignored files](https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html#ignore-files) to be also displayed in the **Local Changes** view, click  on the toolbar and select **Show Ignored Files**.

## Check project status﻿

PyCharm allows you to check the status of your local working copy compared to the repository version of the project. It uses [specific colors](https://www.jetbrains.com/help/pycharm/file-status-highlights.html) to let you see which files have been modified, which new files have been added to the VCS, and which files are not being tracked by Git.

Open the **Local Changes** view.

* The **Default** changelist shows all files that have been modified since you last synchronized with the remote repository (highlighted in blue), and all new files that have been added to the VCS but have not been committed yet (highlighted in green).
* The **Unversioned Files** changelist shows all files that have been added to your project, but that are not being tracked by Git.

For more info on changelists, see [Group changes into different changelists](https://www.jetbrains.com/help/pycharm/work-on-several-features-simultaneously.html#changelists).

### Track changes to a file in the editor﻿

You can also track changes to a file as you modify it in the editor. All changes are highlighted with **change markers** that appear in the gutter next to the modified lines, and show the type of changes introduced since you last [synchronized with the repository](https://www.jetbrains.com/help/pycharm/sync-with-a-remote-repository.html). When you commit changes to the repository, change markers disappear.

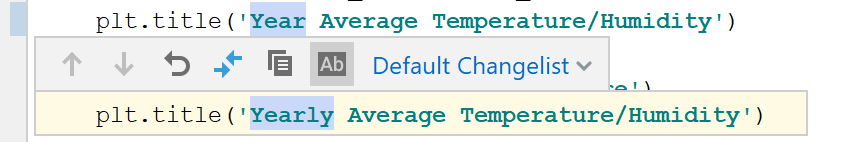
The changes you introduce to the text are color-coded:

* marker for a newly added line line added.
* marker for a modified line line changed.
* marker for a deleted line line deleted.

To customize the default colors for line statuses, in the **Settings/Preferences** dialog Ctrl+Alt+S, go to **Editor | Color Scheme | VCS**.

To disable VCS markers in the gutter, in the **Settings/Preferences** dialog Ctrl+Alt+S, go to **Editor | General** and deselect the **Highlight modified lines in the gutter** option.

You can manage changes using a toolbar that appears when you hover the mouse cursor over a change marker and then click it. The toolbar is displayed together with a frame showing the previous contents of the modified line:



You can roll back changes by clicking  and explore the differences between the current and the repository version of the current line by clicking .

Instead of reverting the whole file, you can copy any part of the contents of this popup and paste it into the editor.

## Add a remote repository﻿

If you [created a Git repository](https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html" \l "put-existing-project-under-Git) based on local sources, you need to add a remote repository to be able to collaborate on your Git project, as well as to eliminate the risks of storing all of your codebase locally. You [push](https://www.jetbrains.com/help/pycharm/commit-and-push-changes.html" \l "push) changes to a remote repository when you need to share your work and [pull](https://www.jetbrains.com/help/pycharm/sync-with-a-remote-repository.html" \l "pull) data from it to integrate changes made by other contributors into your local repository version.

If you have [cloned a remote Git repository](https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html" \l "clone-repo), for example from [GitHub](https://github.com/), the remote is configured automatically and you do not have to specify it when you want to [sync](https://www.jetbrains.com/help/pycharm/sync-with-a-remote-repository.html) with it. The default name Git gives to the remote you've cloned from is **origin**.

### Define a remote﻿

1. Create an empty repository on any Git hosting, such as [Bitbucket](https://bitbucket.org/) or [GitHub](https://github.com/). You can create a repository on GitHub without leaving PyCharm: see [Share a project on GitHub](https://www.jetbrains.com/help/pycharm/manage-projects-hosted-on-github.html" \l "share-on-GitHub).
2. Invoke the **Push** dialog when you are ready to push your commits by selecting **Git | Push** from the main menu, or press Ctrl+Shift+K.
3. If you haven't added any remotes so far, the **Define remote** link will appear instead of a remote name. Click it to add a remote.
4. In the dialog that opens, specify the remote name and the URL where it will be hosted, and click **OK**.

### Add a second remote﻿

In some cases, you also need to add a second remote repository. This may be useful, for example, if you have cloned a repository that you do not have write access to, and you are going to push changes to your own [fork](https://www.jetbrains.com/help/pycharm/contribute-to-projects.html" \l "fork) of the original project. Another common scenario is that you have cloned your own repository that is somebody else's project fork, and you need to synchronize with the original project and fetch changes from it.

1. From the main menu, choose **Git | Manage Remotes**. The **Git Remotes** dialog will open.
2. Click the **Add**  button on the toolbar or press Alt+Insert.
3. In the dialog that opens, specify the remote name and URL and click **OK**.

* To edit a remote (for example, to change the name of the original project that you have cloned), right-click the remote branch in the [Branches pane of the Git Log tool window](https://www.jetbrains.com/help/pycharm/log-tab.html" \l "BranchesPane), and select **Edit Remote** from the context menu.

You can also edit a remote from the [Push Dialog](https://www.jetbrains.com/help/pycharm/push-dialog-mercurial-git.html) by clicking its name.

* To remove a repository that is no longer valid, right-click it in the [Branches pane of the Git Log tool window](https://www.jetbrains.com/help/pycharm/log-tab.html#BranchesPane), and select **Remove Remote** from the context menu.

Learn more from this video:

* [Set up a Git repository](https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html#top)
* [Check out a project from a remote host (clone)](https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html#clone-repo)
* [Put an existing project under Git version control](https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html#put-existing-project-under-Git)
* [Exclude files from version control (ignore)](https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html#ignore-files)
* [Check project status](https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html#check_project_status)
* [Add a remote repository](https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html#add-remote)

Was this page helpful?

YesNo