

**ANGULAR PROJECT SEED**

For this project, whe have a given repo named ITProject-ERP-Frontend, with a GitHub remote URL named <https://github.com/it-academyproject/ITProject-ERP-Frontend.git>...

We will use localProjectFolderName as a generic alias for this example local folder's name.

We will use repoURL as a generic alias for this example remote repo's URL.

We will use featureBranchName as a generic

Remember to use git checkout branchName to switch branches if needed.

Given a GitHub remote repo with files **.gitignore** and **README.md** already in it...

wanting to set up a fresh Angular “ERP” project inside that repo...

follow next steps in console:

1. Clone the repo from remote to local: git clone repoURL

2. Access that folder in local: cd localProjectFolderName

\* (optional) check the branch we're in is "main": git branch

3. Pull Dev branch, for safety reasons: git pull origin Dev

\* (optional) check the branch. Although Dev was pulled, it doesn't show yet: git branch

4. Switch to Dev branch: git checkout Dev (console give us feedback of branch switched)

5. (optional) check that branches Dev and main exist: git branch (now Dev is shown)

Make sure we're on Dev branch. Checkout branch if needed.

6. Create new feature branch with ID as stated in SCRUM:

git checkout -b featureBranchName

\* (optional) check the branch we're in is featureBranchName: git branch

Checkout branch if needed.

7. Copy **.gitignore** and **README** files in a backup folder outside the localProjectFolderName



8. Erase **.gitignore** and **README** inside localProjectFolderName, since Angular CLI will fail to init project in that folder because those 2 files already exist.

\* (optional) check previous deletion was registered in git: git status

A warning message related deletion of **.gitignore** and **README** files should arise.

Don't worry. We will recover those 2 original files from the bckup folder later



9. Finally, init Angular project without overriding existing git folder.

To do so, make sure you're in localProjectFolderName. Then enter following command...

ng new localProjectFolderName --skip-git --directory=./

This command wil init Angular project in localProjectFolderName, not in a subfolder, and it will be done without overriding original git file. When configurating the project, select: SPA (Routing yes) and SASS

\* (optional) check **.gitignore** and **README** are modified and all the rest of Angular's installation appear as new fiels: git status

10. Now, erase Angular's **.gitignore** and **README** from localProjectFolderName

\* (optional) check **.gitignore** and **README** were deleted: git status

11. And recover old original Angular's **.gitignore** and **README** files from backup folder. Remember? And paste them in localProjectFolderName



12. Check all the recovery process went fine: git status

Now, no warnings related to **.gitignore** and **README** should appear.



13. Modify **.gitignore**, to ignore **node\_modules**, when pushing to GitHub.

NOTE: Remember to use command npm install, to download node-modules as stated in package.json of your Angular project, qhen you're cloning a repo to local. But never push node-modules to GitHiub, since the folder is too big sized.

So just add this lines to top of **.gitignore** file:

# dependencies

/node\_modules

NOTE: In Visual Studio Code, ignored files and olders apperar in gray color in project tree

14. Pull Dev branch state before pushing featureBranchName: git pull origin Dev

15. Add changes (we did in .gitignore) to Stage. Then Commit with a message.

Example message: "project init + node-modules ignored"

\* (optional) check there's no warning: git status

16. Create a folder named **documentation** in localProjectFolderName

Now, add a provisional documentation of the seeding. In that **documentation** folder, we'll save all other documentations. Use open office extension (.docx)

RECOMENDED: Follow this naming patter: concept.documentation.docx

Example: seed.documentation.docx, nav.documentation.docx, ngrx.documentation.docx...

17. Add changes (we did in **documentation** folder) to Stage. Then Commit with a message.

Example message: "documentation folder added"

\* (optional) check there's no warning: git status

18. Make and paste current documentation as (.docx) to previus folder:

19. Add changes (we did in seed.documentation.docx file) to Stage.

Then Commit with a message. Example message: "seed.documentation added"

\* (optional) check there's no warning: git status

20. Pull Dev branch state before pushing featureBranchName: git pull origin Dev

21. Push featureBranchName: git push origin featureBranchName

22. Ask for a **pull request** in GitHub

- In the repoURL, select your branch and click on the **New pull request** button

- Select **Dev** as a base branch (because we want to merge our changes with Dev) and select your featureBranchName as the branch to compare.

- Click on the **Create Pull Request** button

(optional) You can copy/paste the URL to pass to your partners the info of the changes that

you're requesting to merge

- Once some of your partners code-reviewed your changes, you can merge with **Dev** branch using the **Merge pull request** button