STARTING A GITHUB REPO

VERSION CONTROL FOR ALL YOUR JAVA NEEDS

VOCAB REMINDERS

**git** A powerful, open source Version Control solution. Hopefully you

already have it installed or this is gonna be a really short trip.

**GitHub** A great place online to store your git repositories for collaboration

and backup purpose.

**repository (repo)** How git keeps your code including all its versions (if you follow

these directions)

**.gitignore** A file that tells git what files you never want to include—files that

are sensitive or unnecessary.

LET’S MAKE THIS HAPPEN!

1. Start your project in IntelliJ and write some code as usual.
2. In IntelliJ, right-click on the project folder (not src) and choose New > File. Name the file .gitignore and paste the .gitignore contents pinned on Slack.
3. Go to **www.github.com** (and log in if you haven’t already).
4. Click the + button near the top right, then click **New repository**.
5. Give the repo a descriptive but short name. If you’re doing a lab, to make it easy to find later you might call is **Lab##description**, like **Lab02AdventureGame**. (This name does not have to match your project name although that’s not a bad habit to get into.)
6. Type a one-sentence description (future employers may look at this!) then click **Create repository**.
   * You could create a readme here but then you’d need to pull this repo. To keep life simple we’ll make it in your project directory.
   * Likewise, you could choose a .gitignore here but it doesn’t exclude everything we’ll want to, so use the one from class.
   * With a free account you can’t choose a private repo.
7. Open up Terminal (Mac) or GitBash (Windows).
8. At the command prompt, use the **cd** command to navigate to your project folder. The specifics of this are going to vary based on where you keep your workspace. For many users, it’s going to be **cd IdeaProjects/projectname** where projectname is the IntelliJ project name.
9. Type the following commands. (Note that you can copy most of these commands from the repo webpage and paste them into your command prompt with command-V (Mac) or right-click, Paste (Windows).)

|  |  |  |
| --- | --- | --- |
| **Command** | **Explanation** | |
| **git init** | | Create the repo here (in the current folder). This will pick up your .gitignore too. |
| **echo "# *projname*" >> README.md** | | Create a readme file with the comment projname (where projname is the name if your GitHub repo) |
| **git add .** | | Add everything in the project folder. Git will ignore things based on the .gitignore you already have, but it will get your source code and other relevant files. |
| **git commit -m "first commit"** | | Commit the staged files with a comment that this was the first commit. |
| **git remote add origin**  [**git@github.com**](mailto:git@github.com)**:**  **pguenthe/*projname*.git** | | Connect this local repo with the repo you created on GitHub. Make sure *projname* is the name of your GitHub repo—**this command is a good one to copy and paste from the GitHub page!** |
| **git push –u origin master** | | Copy the repo (including your newest commit) up to GitHub |

1. In the future, when you’ve made changes to project files, get back into the command prompt, navigate back to the project folder and use these commands:

|  |  |
| --- | --- |
| **Command** | **Explanation** |
| **git add src/*filename*** or **git add .** | Add the relevant files. This will vary based on what you’ve modified. |
| **git commit -m "commit reasons"** | Commit the staged files with a comment that explains to teammates and/or your future self what you changed or why. |
| **git push –u origin master** | Copy the repo (including your newest commit) up to GitHub |