Using GitHub in Android Studio

Overview

We will learn how to:

- configure your Android Studio for using GitHub
- create an AS project (incorporating an SQLite database) based on a GitHub repository
- create a GitHub repository and push your new project to the repository

Git and GitHub

- git is a very popular distributed version control system (VCS), commonly used for collaborative development of software systems
- Many other VCS systems exist, e.g., Mercurial, Azure DevOps, Subversion, and others
- It was originally created in 2005 by Linus Torvalds for maintaining Linux kernel development

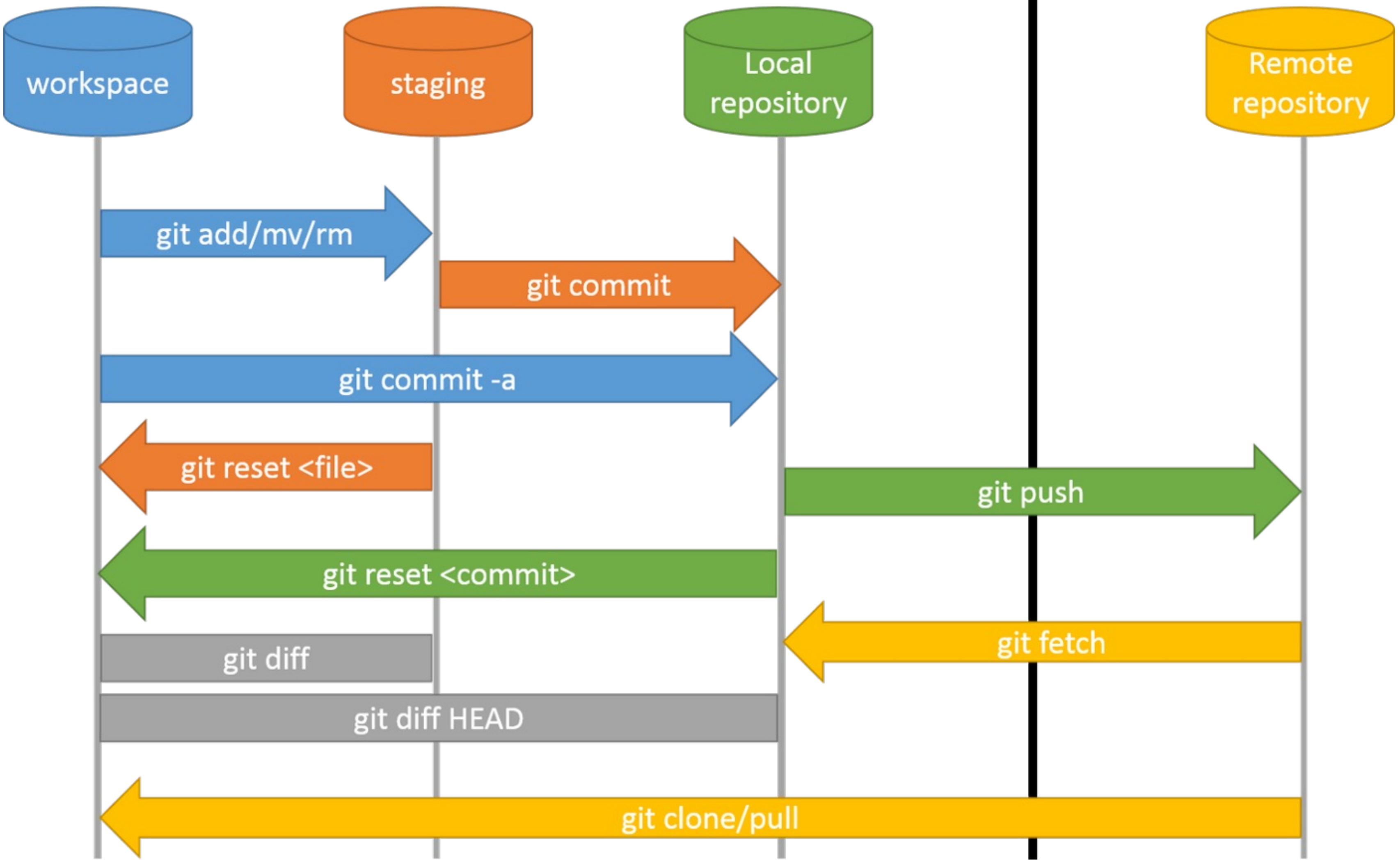
Git and GitHub

- You probably know how to use git from previous UGA courses, such as CSCI 1302
- GitHub is a hosting site for collaborative software development
- It uses git to maintain software repositories
- GitHub offers many other types of collaborative software development functions, as well (e.g., bug tracking, project management, team administration, and many others)

Atlassian, a GitHub Competitor

- Atlassian, is a well-known, direct competitor of GitHub
- Atlassian offers Bitbucket, which is similar to GitHub (https://bitbucket.org/)
- Atlassian also offers Jira, a system to manage software defects and manage projects (https://www.atlassian.com/software/jira)

Git Basics



From: medium.com/@mehulgala77 github-fundamentals-clone-fetch-push-pull-fork-16d79bb16b79

Preliminary Steps

- Create your account at github.com, if you don’t have one yet; select **Sign up** on top, right
- Provide your email and select a strong password
- You will have to setup a two-factor authentication
- Once you have a GitHub account, log in and create a Personal Access Token:
  - Click your user icon (top, right corner) and select **Settings**
  - In the new screen, on the left, scroll down to the bottom and select **Developer settings** and then, on the left, look for **Personal access tokens**, open the drop-down and then select **Tokens (classic)**

Preliminary Step

- Open **Generate new token** and pick **Generate new token (classic)**
- Pick an **Expiration** date to last at least until the end of the semester
- Enter a brief note about the token, e.g., **Repo access**
- **IMPORTANT**: Set the scopes for the new token as **repo**, **gist**, **read:org**, and **workflow** (mark the corresponding checkboxes)
- Then, click the **Generate token** button
- On a new screen, **copy your token string** to the clipboard (there is a convenient icon to the right of the generated key) and **save it in a file somewhere, as you will need the token to access your repositories**

Android Studio Configuration

- Open Android Studio, go to File->settings (Windows) or Android Studio->preferences (Mac) and select Version Control->GitHub
- Click the plus (+) on top menu and pick **Login with token…**
- Copy your token string into the text box and click **Add Account**
- You will now be able to access GitHub from Android Studio using your token
- Close this dialog by clicking **OK**

JobsTrackerSQLite App

- Open Android Studio and close the current project, if one is open
- On the welcome to Android Studio, click `Get from VCS` (you may need to open the overflow area – the three vertical dots on the right)

Alternatively, with a project still open, select

File -> New -> Project from Version Control -> GitHub

JobsTrackerSQLite App

- You may get a message **Git is not installed**

  If so, **Download and install** the plugin.
- In the popup window, enter the following URL of the repository:

  https://github.com/mobdevuga/jobstrackersqlite
- Then click on the **Clone** button and agree that you Trust the project
- Recently, I experienced problems with the compilation of projects initially checked out from github. If so, close the AndroidStudio and open the project again. Likely, this will clear out the compilation problems.

JobsTrackerSQLite App

- The project will be cloned into your local git repository and the project will also be opened in Android Studio
- You may need to wait for Android Studio to finish the gradle configuration and build the project
- The repository is read-only, so you will not be able to commit and push any changes
- Build the project and run it in the emulator.
- **You should examine the code carefully. It illustrates most of what you need for Project 4.**
- We will discuss this app in class, as well.

Share the Project to GitHub

By now, you should have an account on GitHub and a personal token available

- As a general rule, get an app to a reasonable stopping point
- When your app is functioning properly, you can share your app on GitHub
- From the VCS drop-down menu, select `Share project on GitHub`.

Share the Project to GitHub

- In a dialog box, select `Add account` and then pick `Log in with Token...`
- A new dialog box will open
- Copy your token string into the text box provided and your user information will be established; if you configured your GitHub user info ahead of time, this step will not be needed
- In the Share Project on GitHub dialog box, you will see you user information, now set

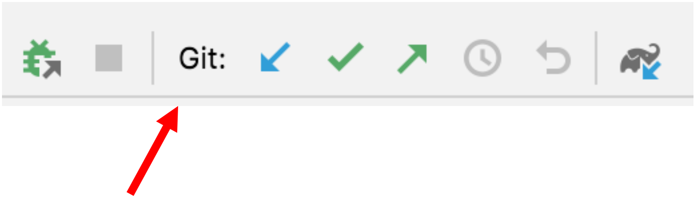
Share the Project to GitHub

- Check the repository name (it defaults to your AS project name) and make sure is what you want
- Click the `Share` button
- A new dialog box `Add Files For Initial Commit` with new project files will open
- Click the `Add` button
- A new repository will be created, and the AS project will be pushed to GitHub

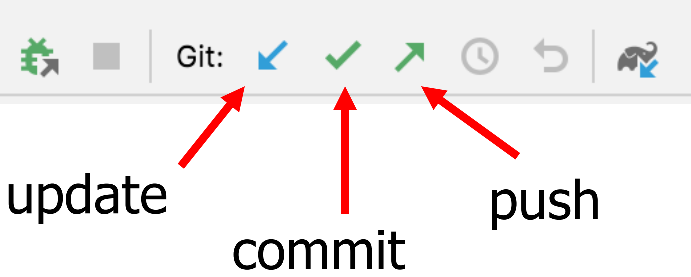
Commit and Push to GitHub

Working with a project already in GitHub

- When your app is functioning properly, you should commit and push your most recent changes to GitHub.
- Your project in AS should have the Git-related menu items:



Commit and Push to GitHub



- In the Android project panel, all files not currently under Git control will have their names in red font
- Click on the Commit button. You will be able to add new files to git control and/or commit any modified files.
- In the new window, at the top on left, you will see a checkbox with un-versioned files; you should check the box to indicate you want all files to be added to the git repository

Commit and Push to GitHub

- In the Commit Message panel, enter a simple comment about the most recent changes and click the Commit button
- Android Studio will perform code analysis and may likely find several issues, which are usually just warnings
- Once the analysis is finished and problems have been found, you can:
  - click on the `Review` button to review the problems; you may correct some or all of them and Redo the Commit steps (above)
  - click on the `Commit` button, if you trust that the reported problems are minor warnings

Commit and Push to GitHub

- Once your changes have been committed, you can push your local git repository to a remote repository
- Your teammate(s) may then update their project and their Android Studio will have the updated files