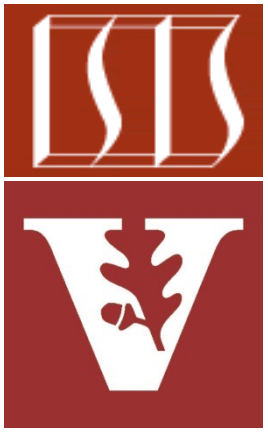
CS 253: Parallel Functional Programming with Java & Android : Overview (Part 3)

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



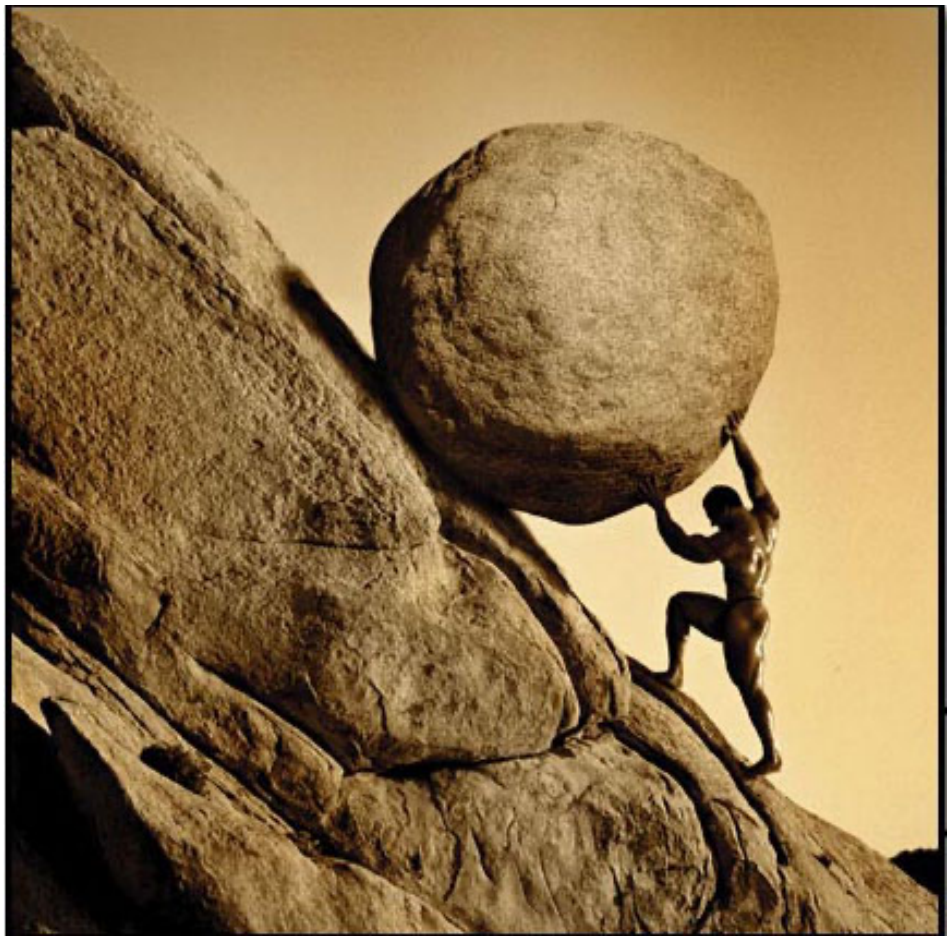
Professor of Computer Science

Institute for Software Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Lesson

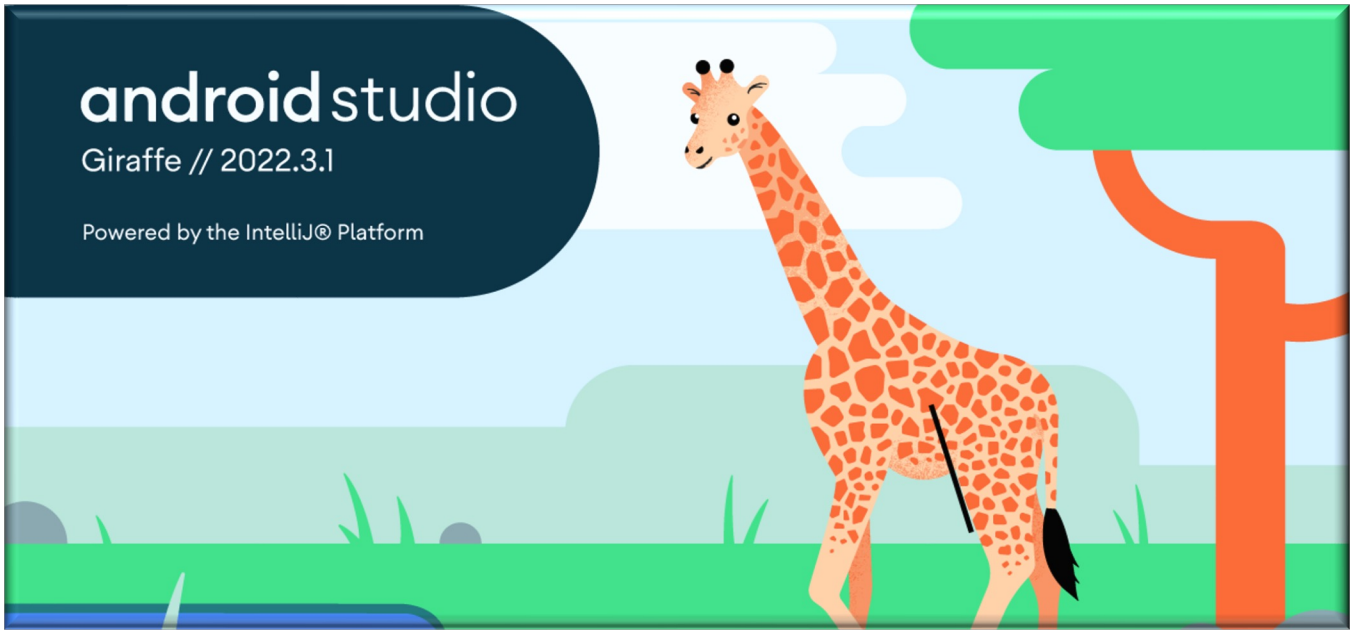
- Understand the course topics & logistics
  - Course philosophy
  - Course contents
  - Structure of the lecture material
  - Overview of the assignments & assessments
  - Setting up the Java & Android IDE on Android Studio
  - Setting up GitLab et al.
  - Accessing Android & Java source code



Setting Up the Android & Java IDE on Android Studio

Installing Java/Android Developer Tools

- To use Android, you need to install the latest release of Android Studio



Select the version of Android Studio that's right for your Mac:

Android Studio Giraffe | 2022.3.1

Mac with Intel chip — *android-studio-2022.3.1.18-mac.dmg*

Mac with Apple chip — *android-studio-2022.3.1.18-mac_arm.dmg*

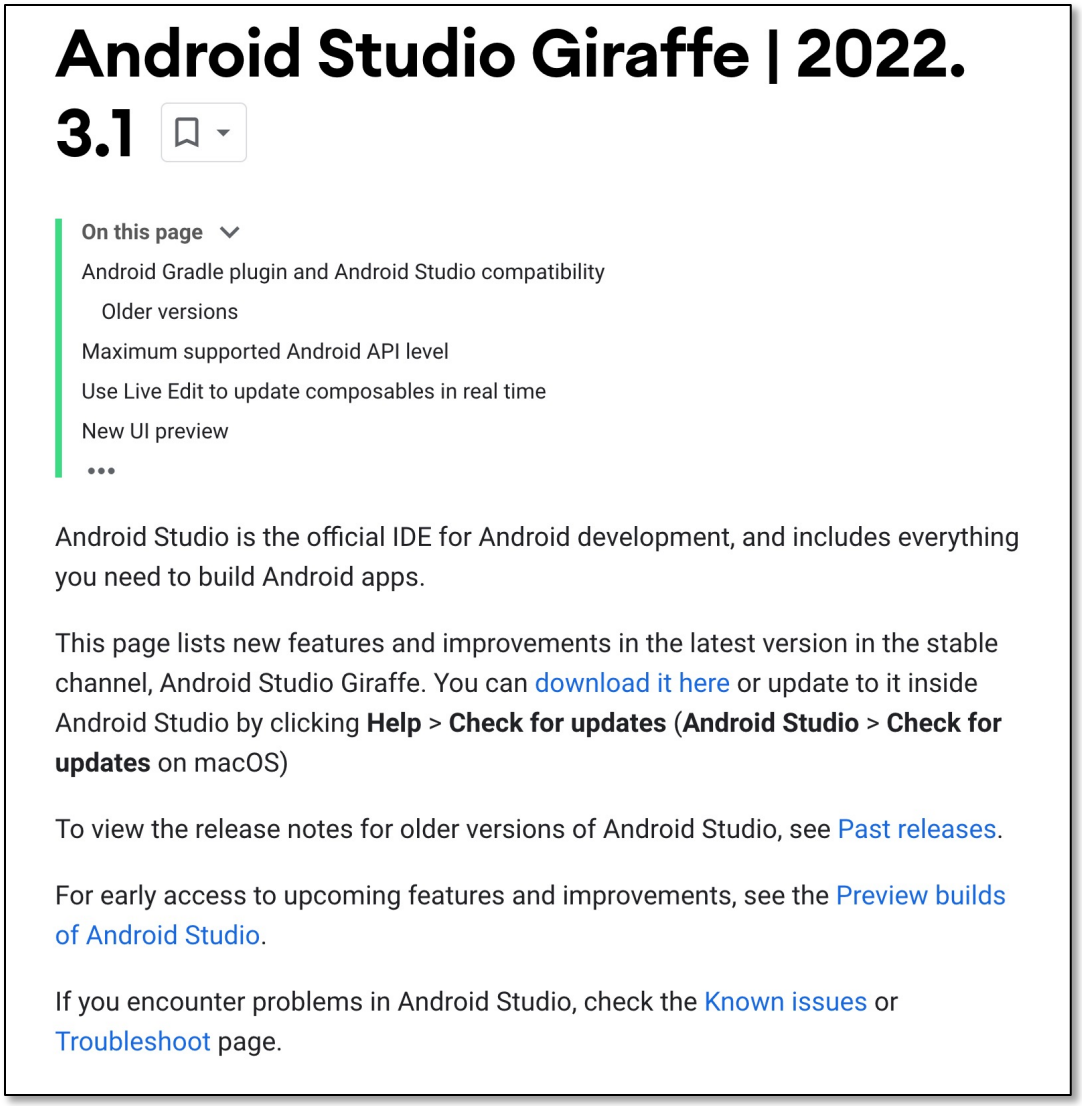
See developer.android.com/studio

Installing Java/Android Developer Tools

- Installation steps

Installing Java/Android Developer Tools

- Installation steps
  - Download & install the latest version of Android Studio



Android Studio Giraffe | 2022.3.1

On this page

Android Studio is the official IDE for Android development, and includes everything you need to build Android apps.

This page lists new features and improvements in the latest version in the stable channel, Android Studio Giraffe. You can download it here or update to it inside Android Studio by clicking **Help** > **Check for updates** (**Android Studio** > **Check for updates** on macOS)

To view the release notes for older versions of Android Studio, see Past releases.

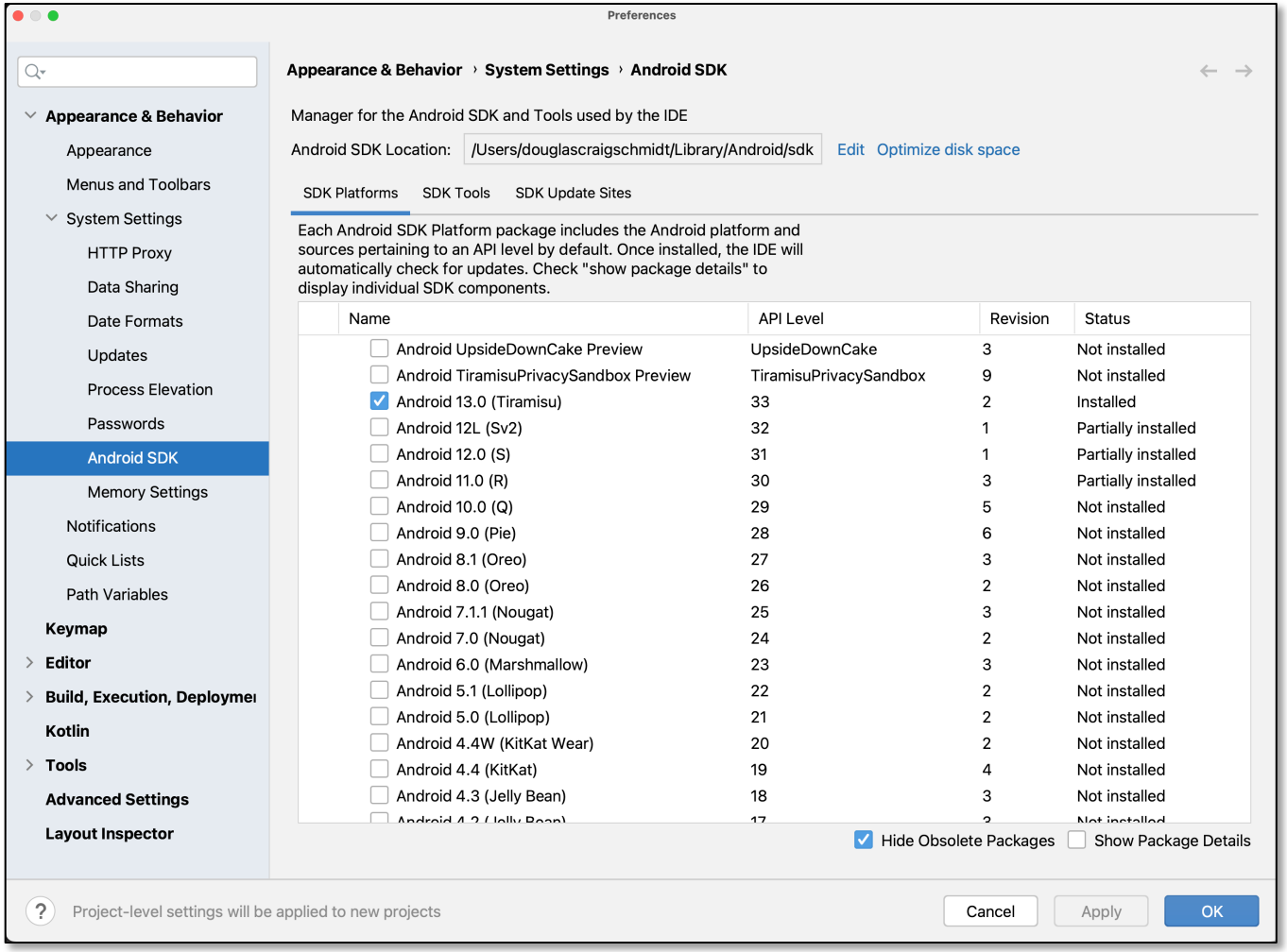
For early access to upcoming features and improvements, see the Preview builds of Android Studio.

If you encounter problems in Android Studio, check the Known issues or Troubleshoot page.

See developer.android.com/studio

Add Components to the SDK

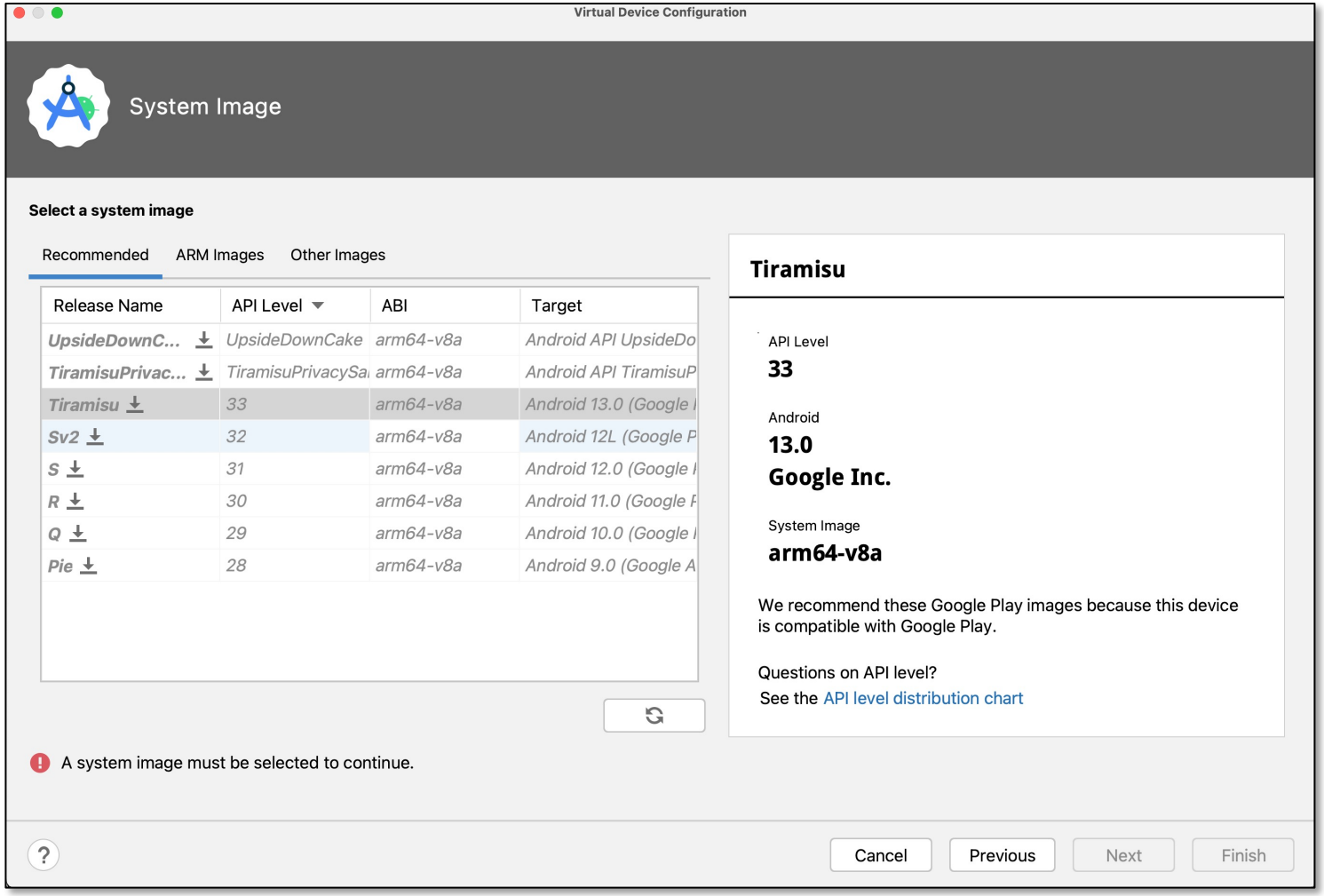
- Launch the Android Studio SDK Manager
  - Select Android 13, API level 33



See developer.android.com/studio/intro/update.html

Add Components to the SDK

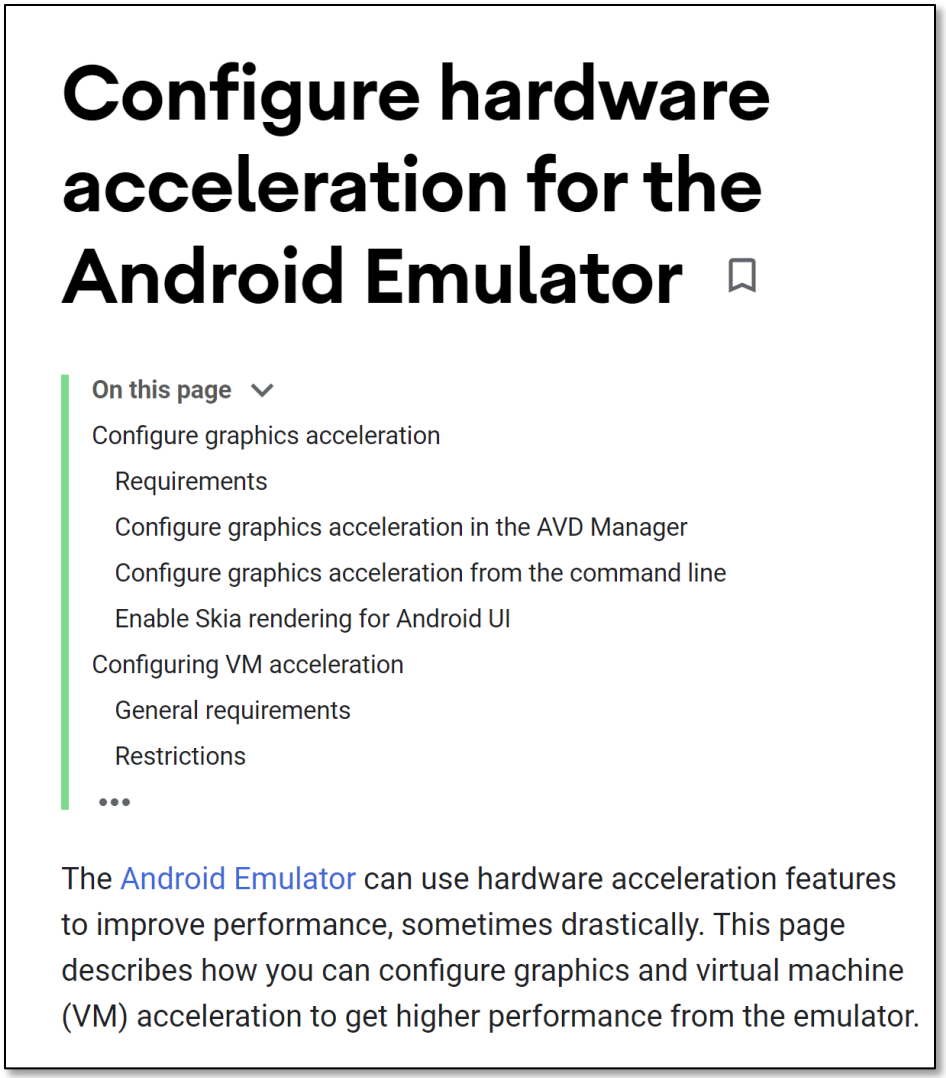
- Launch the Android Studio Virtual Device Manager
  - Create an Android API 33 emulator



developer.android.com/tools/devices/managing-avds.html

Intel HAXM Virtualization Driver

- **Requirements**
  - Intel virtualization extensions (VT, VT-x, vmx)
  - AMD virtualization extensions (AMD-v, SVM) [only supported on Linux]
  - Download an x86 emulator image
- **Windows & Mac OSX**
  - <sdk>/extras/intel/Hardware_Accelerated_Execution_Manager/IntelHAXM.exe/dmg
- **Linux**
  - Install KVM & pass "-enable-kvm" flag to emulator when starting



Configure hardware acceleration for the Android Emulator

On this page

Configure graphics acceleration

- Requirements
- Configure graphics acceleration in the AVD Manager
- Configure graphics acceleration from the command line
- Enable Skia rendering for Android UI

Configuring VM acceleration

- General requirements
- Restrictions

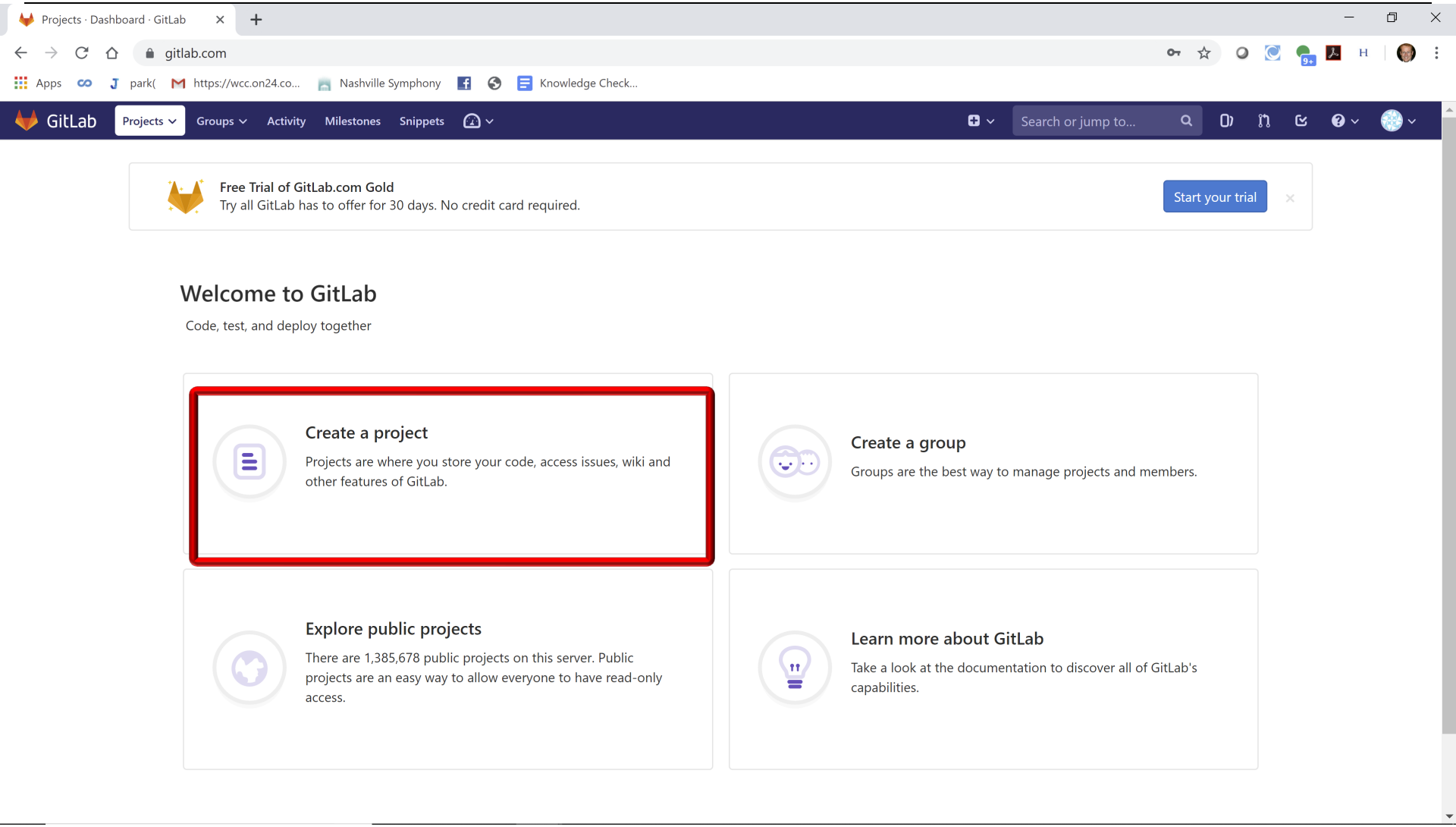
...

The Android Emulator can use hardware acceleration features to improve performance, sometimes drastically. This page describes how you can configure graphics and virtual machine (VM) acceleration to get higher performance from the emulator.

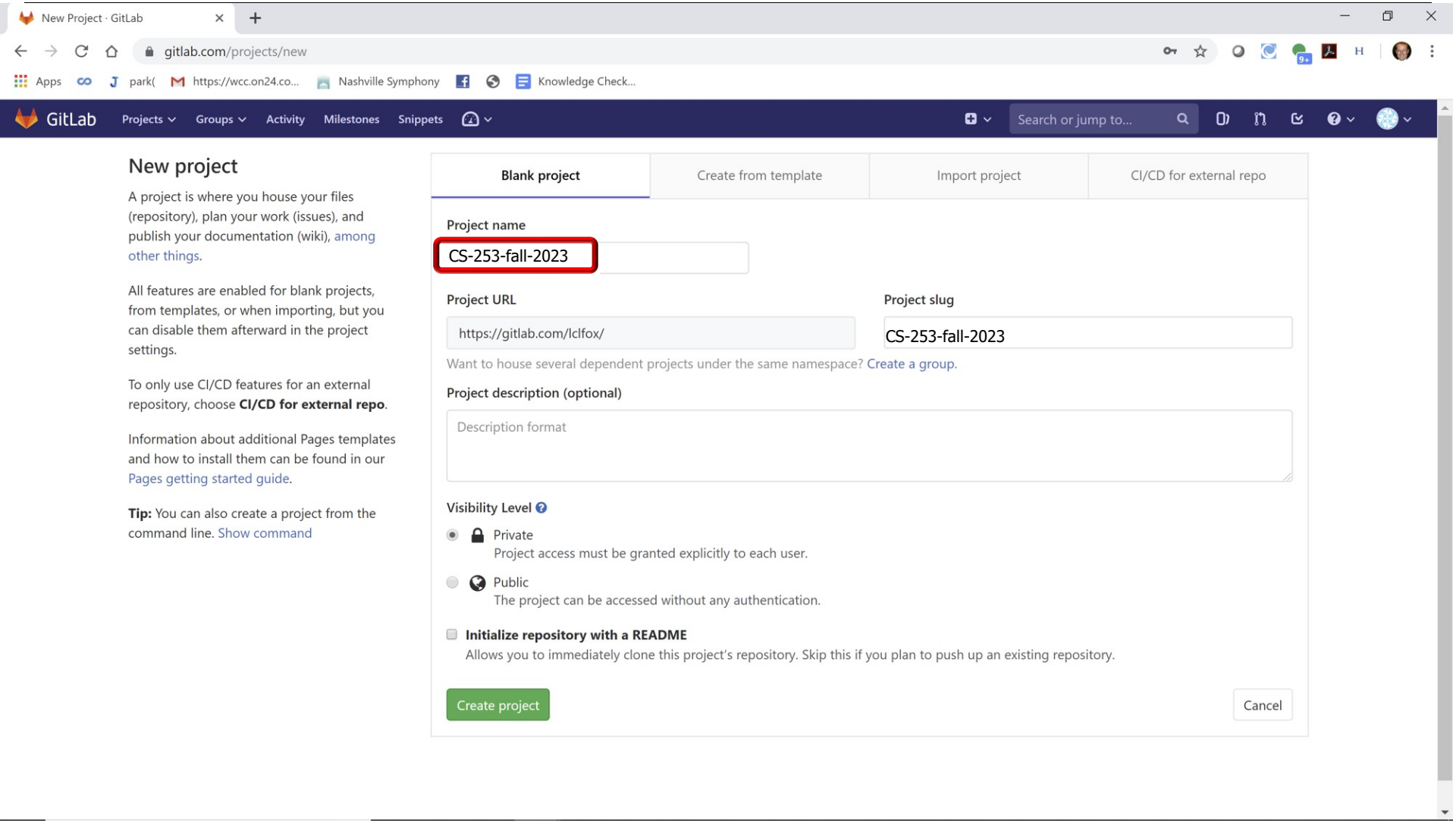
developer.android.com/studio/run/emulator-acceleration

Setting Up GitLab et al.

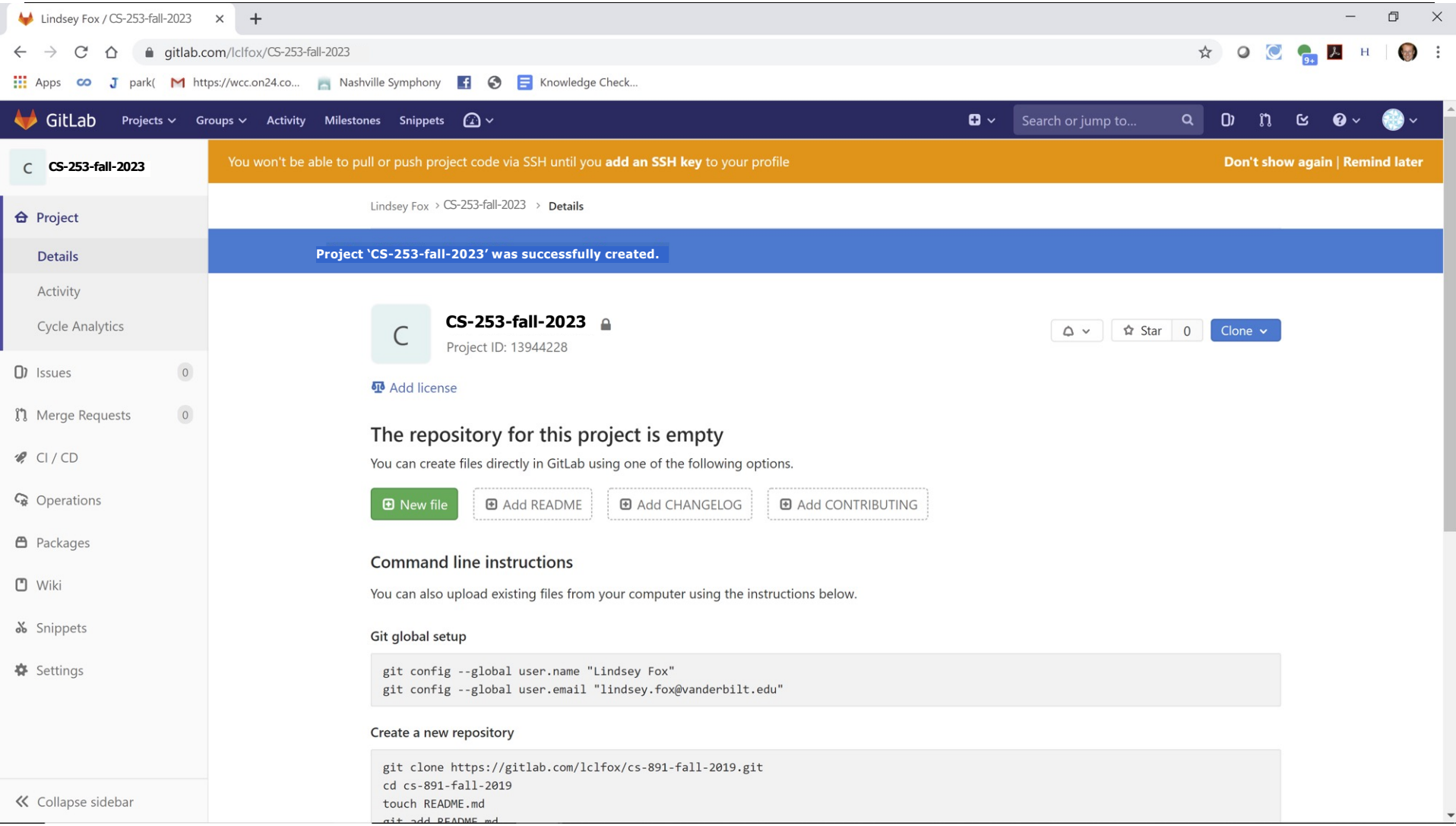
Setting Up GitLab et al.



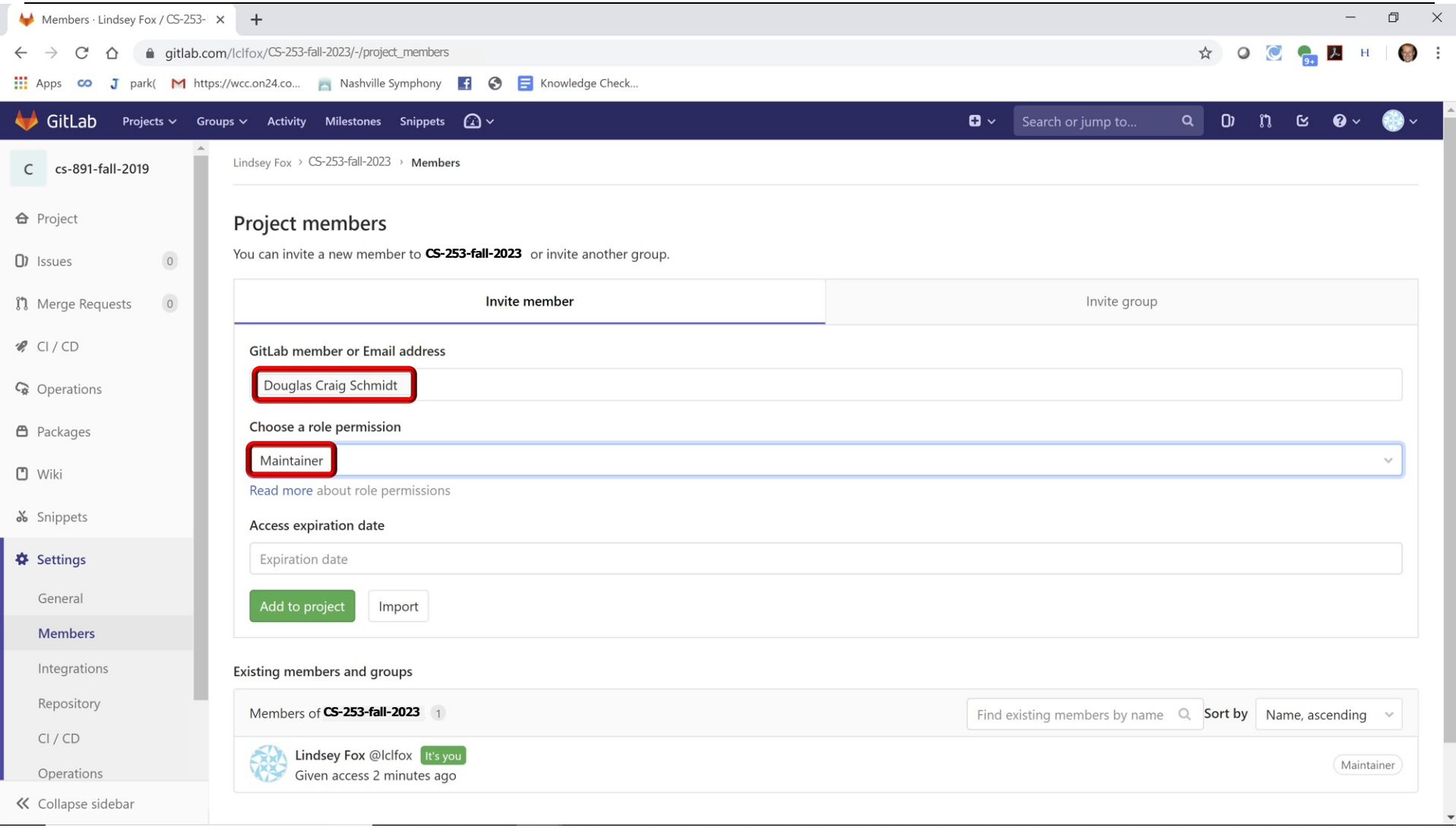
Setting Up GitLab et al.



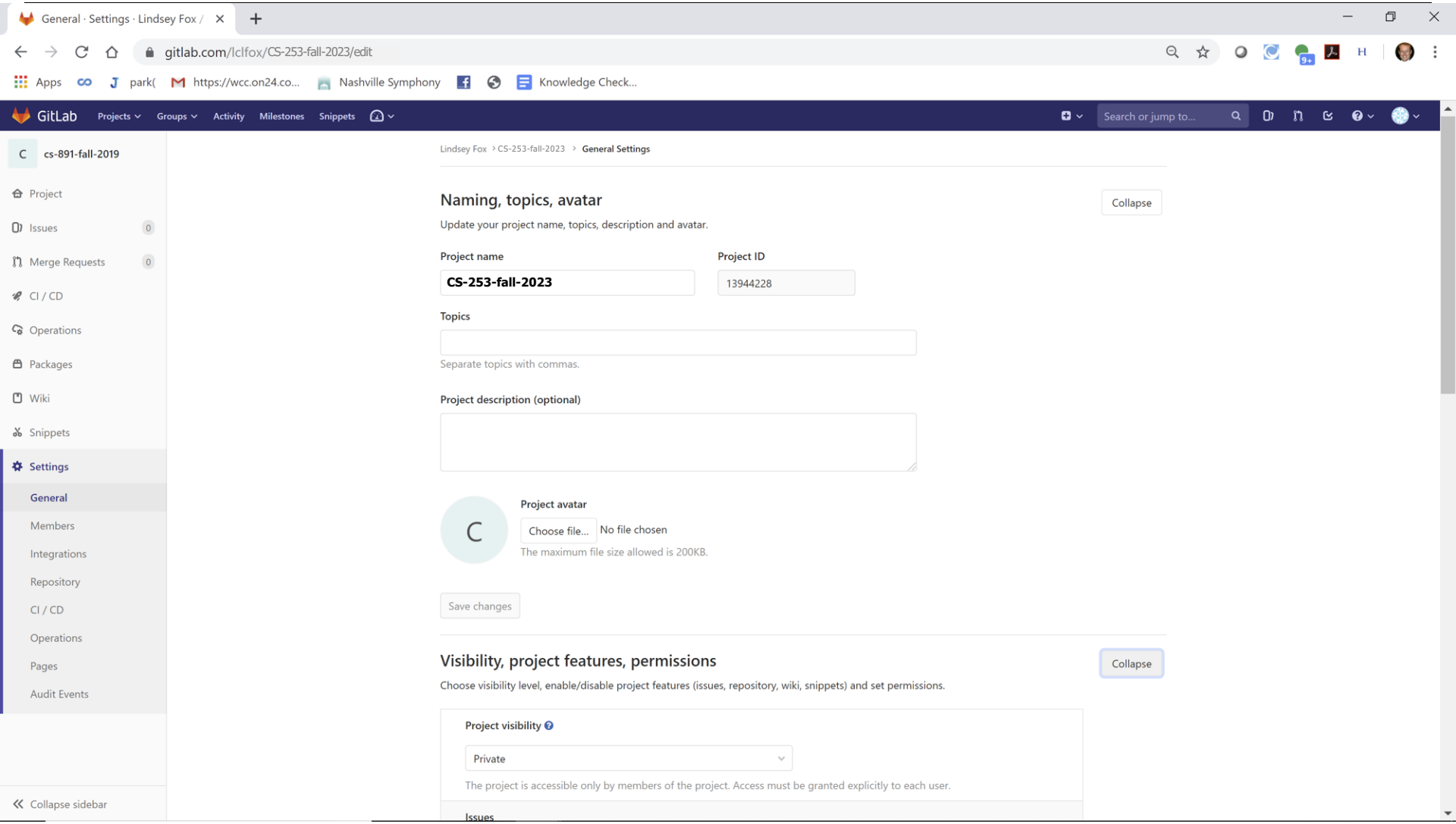
Setting Up GitLab et al.



Setting Up GitLab et al.



Setting Up GitLab et al.



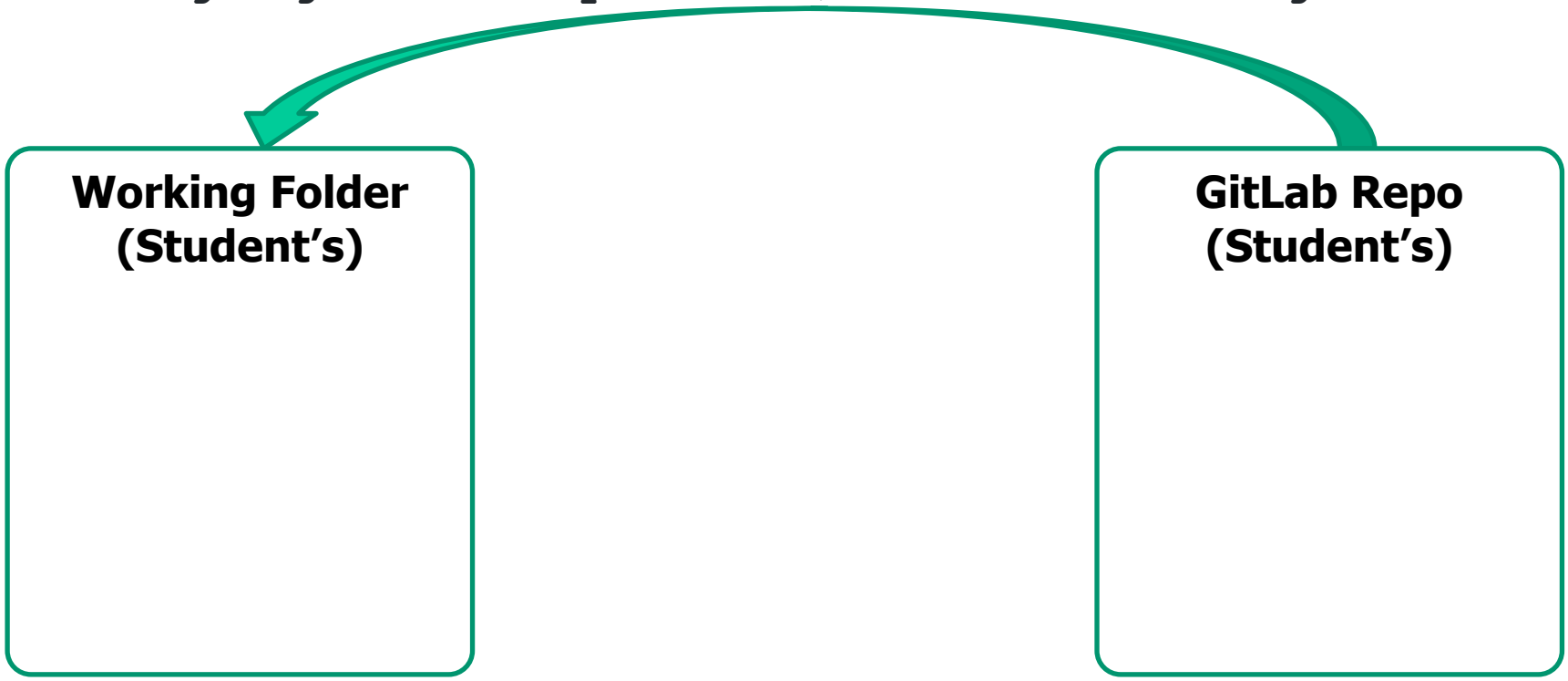
Setting Up GitLab et al.

1. Create Your GitLab Repo

`www.gitlab.com`

2. Clone your GitLab repo

```
git clone git@gitlab.com:your-name/CS-253-fall-2023.git
```




```
cd CS-253-fall-2023
```

3. Change Director into Your Working Folder

See docs.gitlab.com/ee/ssh for info on setting up an SSH key for GitLab et al.

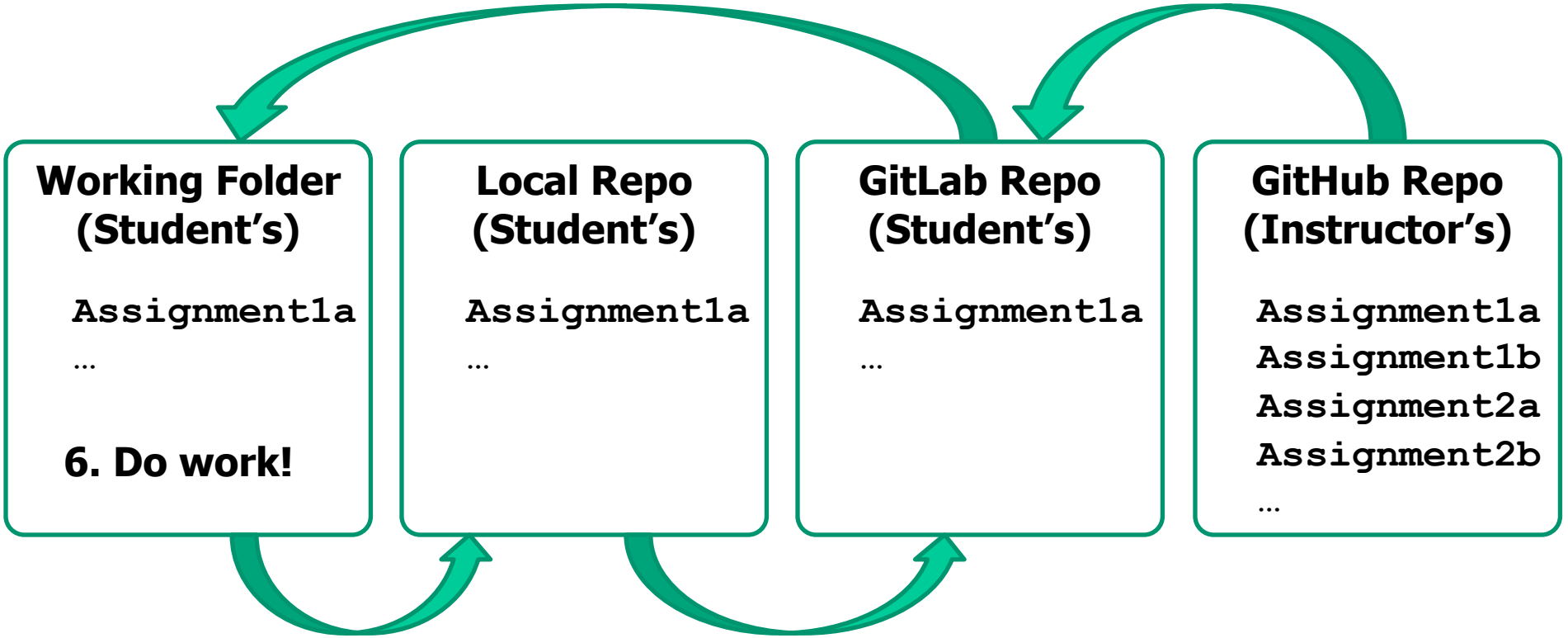
Setting Up GitLab et al.

4. Update from Read-Only GitHub Repo

```
git remote add skeletons
  git@github.com:douglascraigschmidt/CS253.git
```

5. Get Current Version

```
git pull skeletons main
```



7. Commit Changes

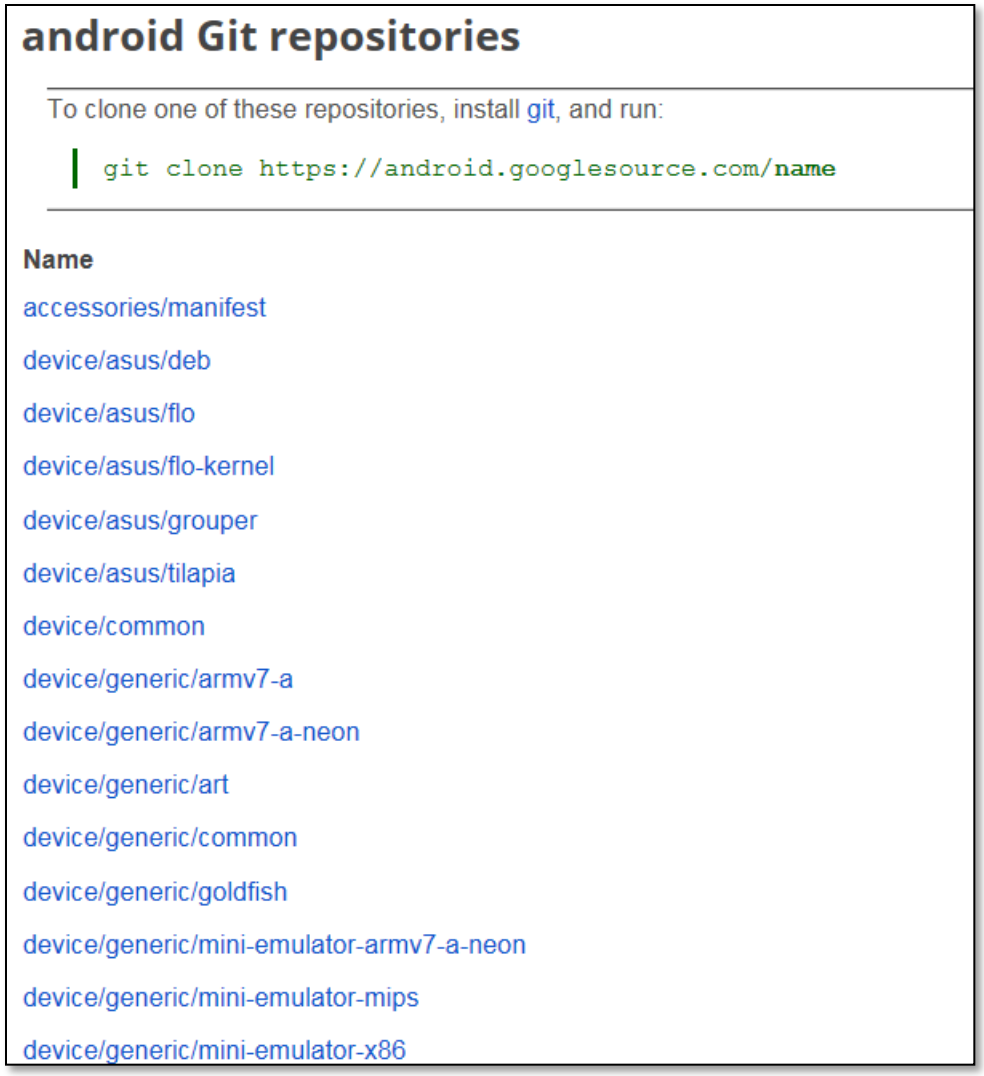
8. Send Changes to GitLab Repo

See item #13 at github.com/douglascraigschmidt/CS253/wiki/CS-253-FAQ

Accessing Java & Android Source Code

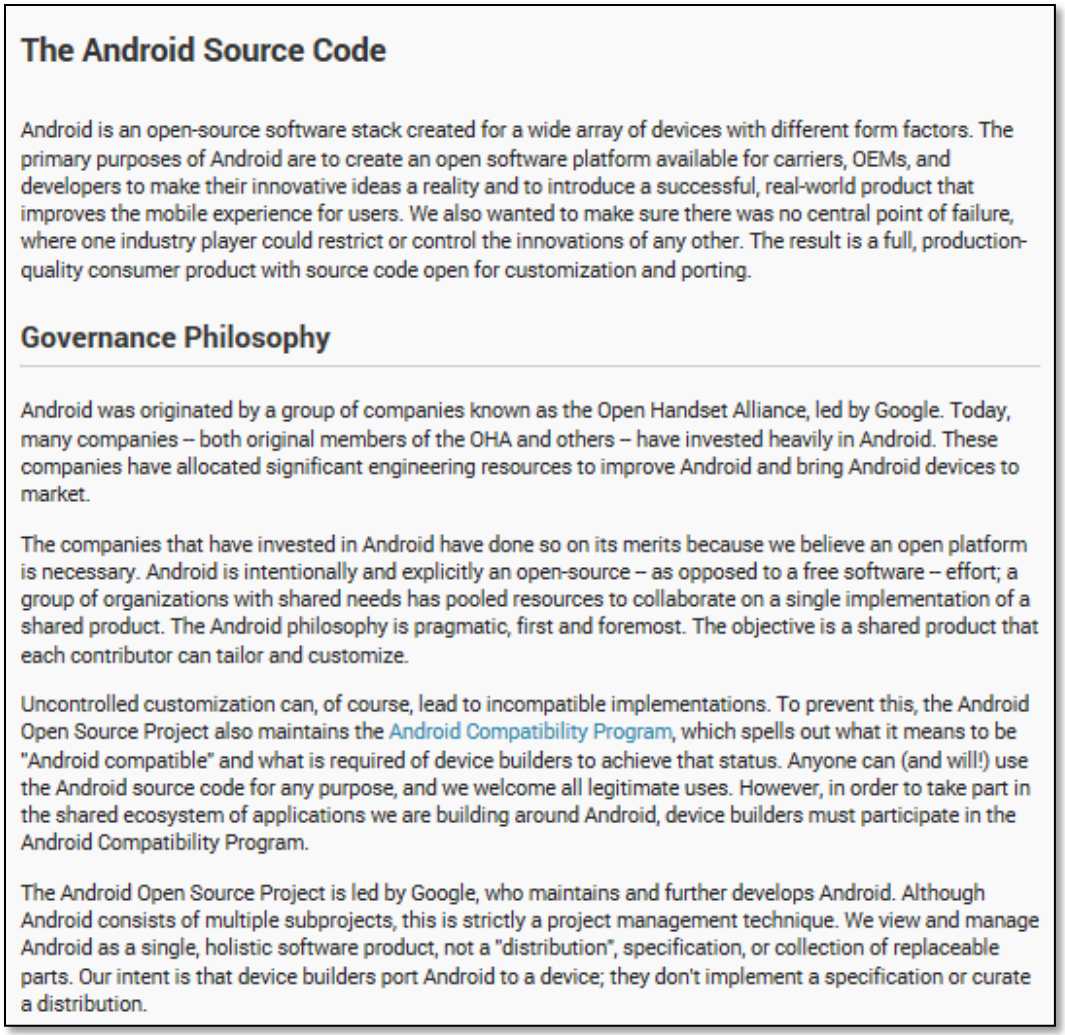
Accessing Java & Android Source Code

- Android source code is available
  - For browsing android.googlesource.com



Accessing Java & Android Source Code

- Android source code is available
  - For browsing android.googlesource.com
  - For downloading source.android.com



The Android Source Code

Android is an open-source software stack created for a wide array of devices with different form factors. The primary purposes of Android are to create an open software platform available for carriers, OEMs, and developers to make their innovative ideas a reality and to introduce a successful, real-world product that improves the mobile experience for users. We also wanted to make sure there was no central point of failure, where one industry player could restrict or control the innovations of any other. The result is a full, production-quality consumer product with source code open for customization and porting.

Governance Philosophy

Android was originated by a group of companies known as the Open Handset Alliance, led by Google. Today, many companies -- both original members of the OHA and others -- have invested heavily in Android. These companies have allocated significant engineering resources to improve Android and bring Android devices to market.

The companies that have invested in Android have done so on its merits because we believe an open platform is necessary. Android is intentionally and explicitly an open-source -- as opposed to a free software -- effort; a group of organizations with shared needs has pooled resources to collaborate on a single implementation of a shared product. The Android philosophy is pragmatic, first and foremost. The objective is a shared product that each contributor can tailor and customize.

Uncontrolled customization can, of course, lead to incompatible implementations. To prevent this, the Android Open Source Project also maintains the Android Compatibility Program, which spells out what it means to be "Android compatible" and what is required of device builders to achieve that status. Anyone can (and will!) use the Android source code for any purpose, and we welcome all legitimate uses. However, in order to take part in the shared ecosystem of applications we are building around Android, device builders must participate in the Android Compatibility Program.

The Android Open Source Project is led by Google, who maintains and further develops Android. Although Android consists of multiple subprojects, this is strictly a project management technique. We view and manage Android as a single, holistic software product, not a "distribution", specification, or collection of replaceable parts. Our intent is that device builders port Android to a device; they don't implement a specification or curate a distribution.

Accessing Java & Android Source Code

- Java source code is available
  - For browsing zgrepcode.com

JDK Project

The goal of this long-running Project is to produce a series of open-source reference implementations of the Java SE Platform, as specified by JSRs in the Java Community Process. The Project ships a feature release every six months according to a strict, time-based model, as proposed.

Releases

- 20 (in development)
- 19 (in development)
- 18 (GA 2022/03/22)
- 17 (GA 2021/09/14)
- 16 (GA 2021/03/16)
- 15 (GA 2020/09/15)
- 14 (GA 2020/03/17)
- 13 (GA 2019/09/17)
- 12 (GA 2019/03/19)
- 11 (GA 2018/09/25)
- 10 (GA 2018/03/20)

Resources

- Development list: *jdk-dev*
- Main-line code repository: https://github.com/openjdk/jdk/

Accessing Java & Android Source Code

- Java source code is available
  - For browsing zgrepcode.com
  - For downloading openjdk.org/projects/jdk

JDK Project

The goal of this long-running Project is to produce a series of open-source reference implementations of the Java SE Platform, as specified by JSRs in the Java Community Process. The Project ships a feature release every six months according to a strict, time-based model, as proposed.

Releases

- 20 (in development)
- 19 (in development)
- 18 (GA 2022/03/22)
- 17 (GA 2021/09/14)
- 16 (GA 2021/03/16)
- 15 (GA 2020/09/15)
- 14 (GA 2020/03/17)
- 13 (GA 2019/09/17)
- 12 (GA 2019/03/19)
- 11 (GA 2018/09/25)
- 10 (GA 2018/03/20)

Resources

- Development list: *jdk-dev*
- Main-line code repository: https://github.com/openjdk/jdk/

CS 253: Parallel Functional Programming with Java & Android: Overview (Part 3)