

# CREDO School

CREDO Detector and other mobile applications

Michał Niedźwiecki\*, Aleksander Ćwikła\*, Marta Marek\*\*

**\*Cracow University of Technology**

**\*\*Astroparticle Physics Enthusiast**

# Agenda

it should be done before the school :)

**Install and configure Android Studio**  
for compile app from source code

**CREDO Detector and other mobile applications**  
the presentation

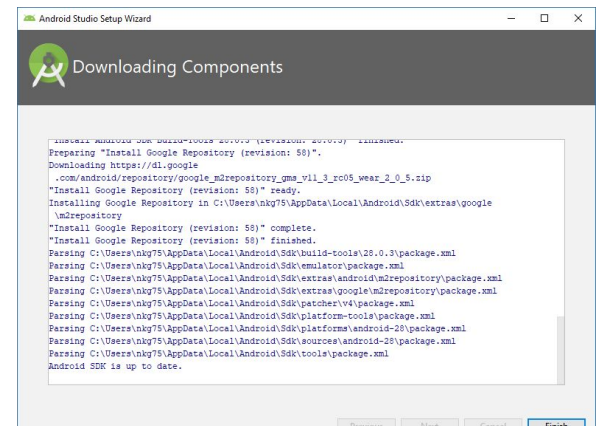
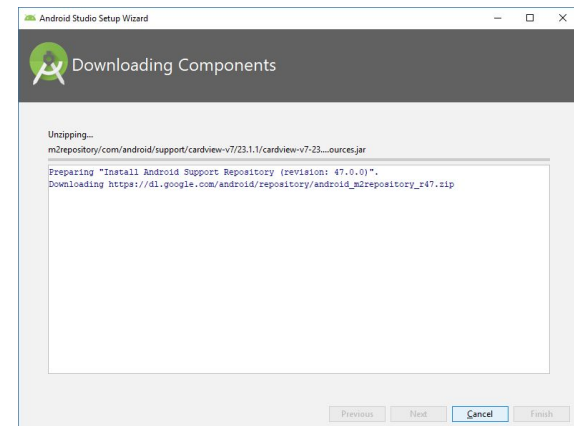
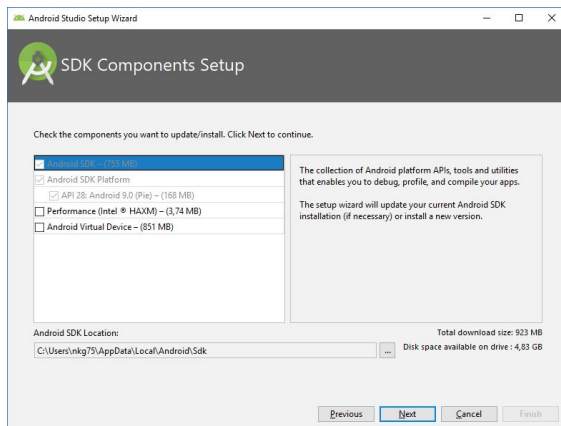
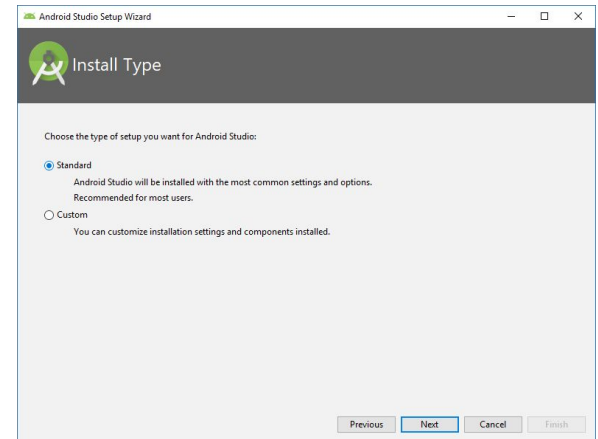
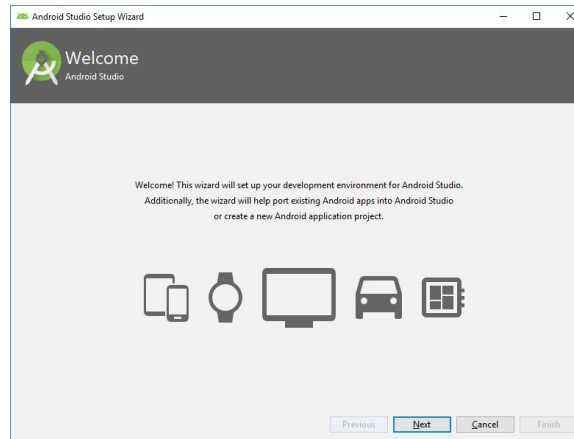
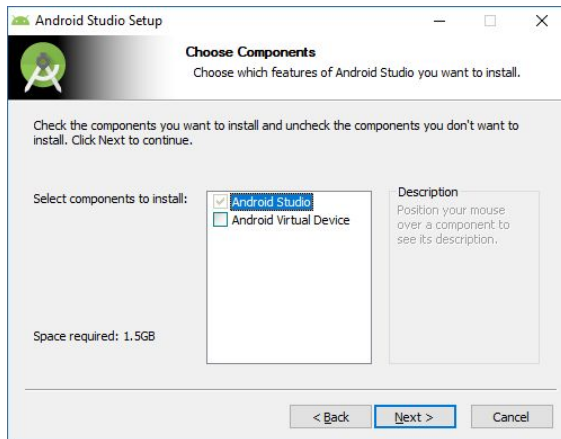
**How to use CREDO Detector mobile application**  
install, register/login, covering and run detection

**Compile CREDO Detector Android app from source code**  
and contribute on GitHub project

# Install and configure Android Studio

Please download and install Android Studio 3.2:

<https://developer.android.com/studio/>

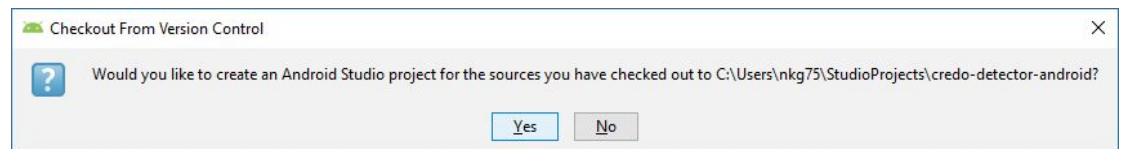
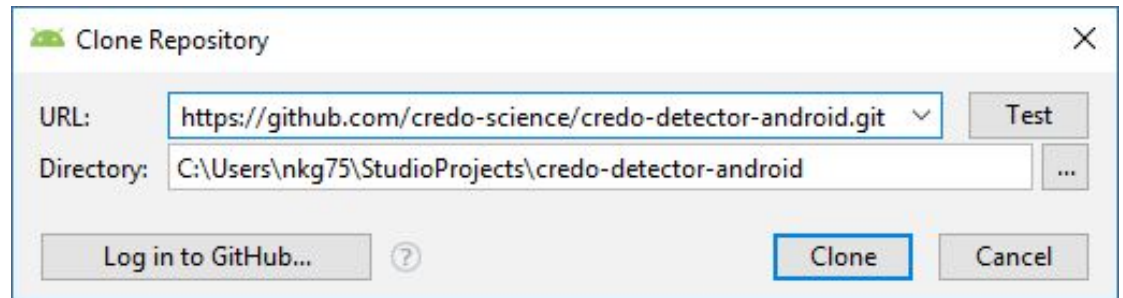
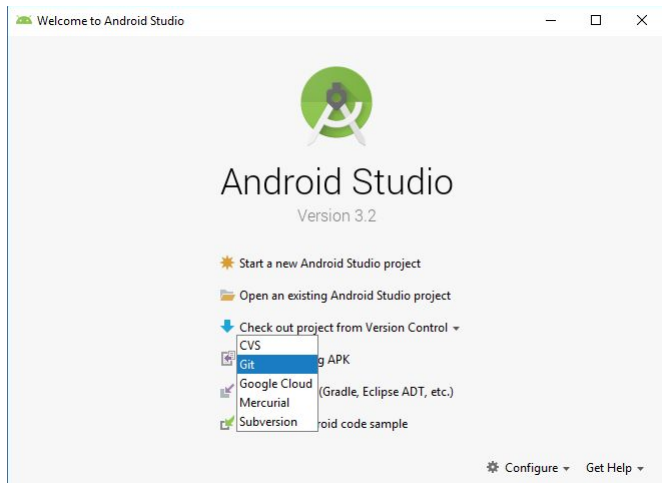


# Clone repository from GitHub

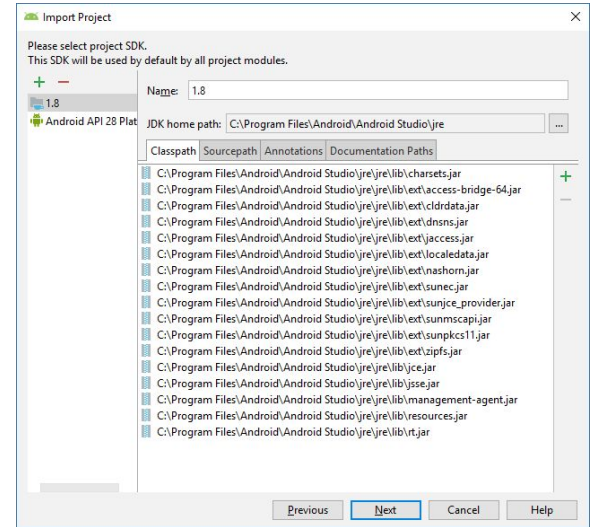
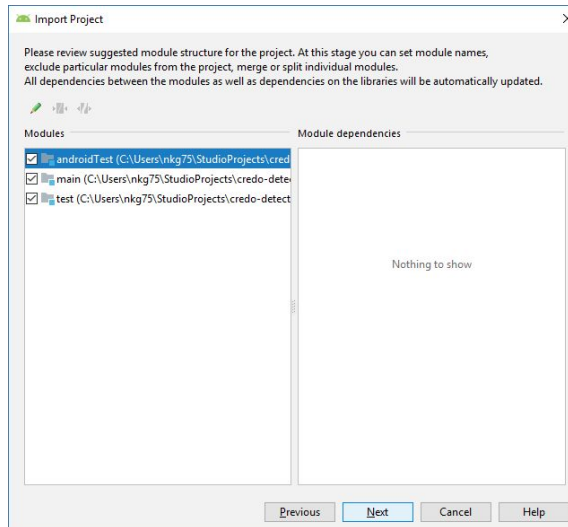
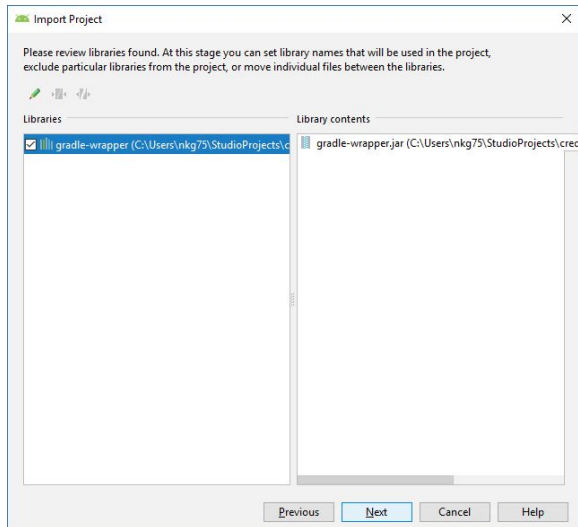
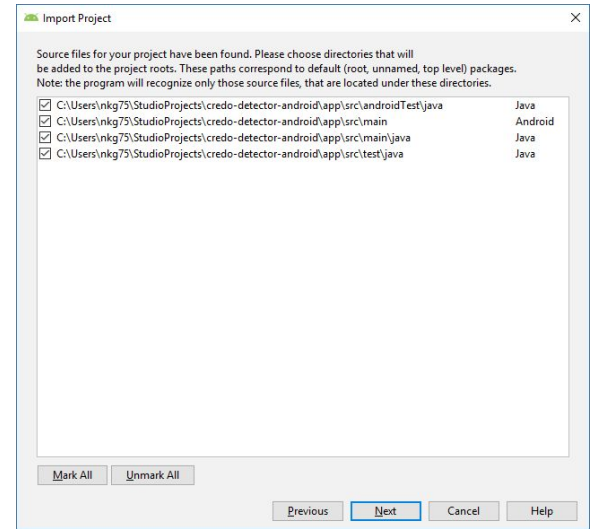
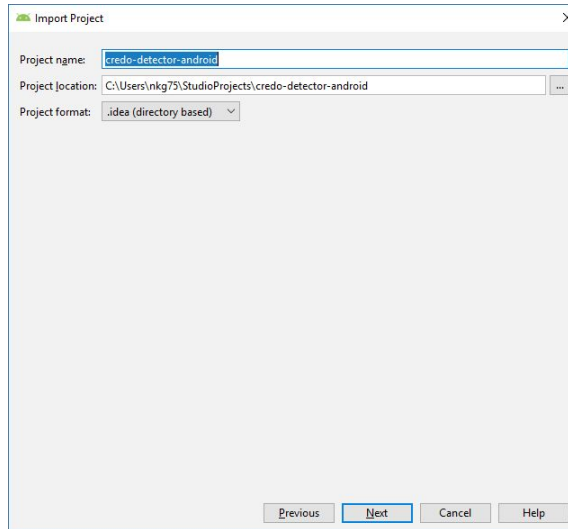
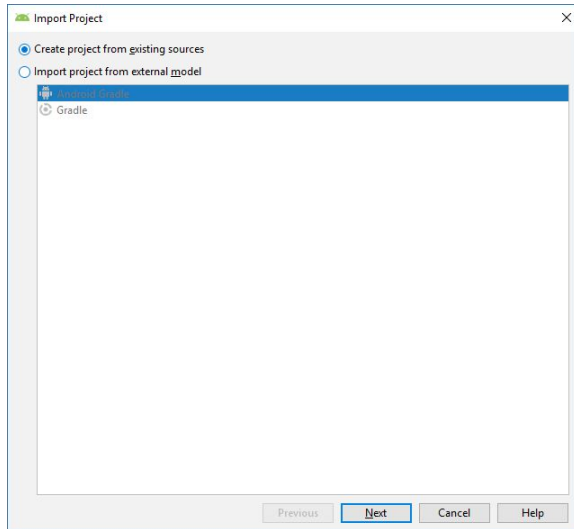
Please open Android Studio and click on menu:

**File** → **Project from Version Control** → **Git**

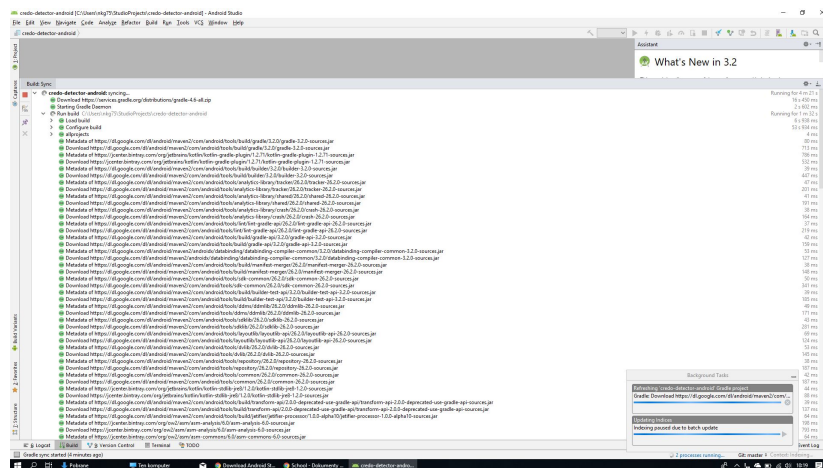
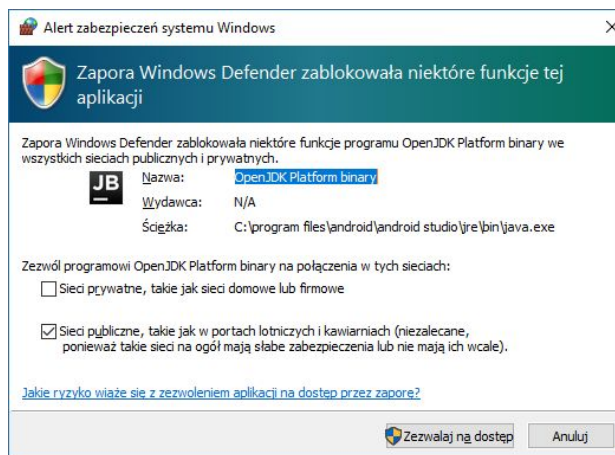
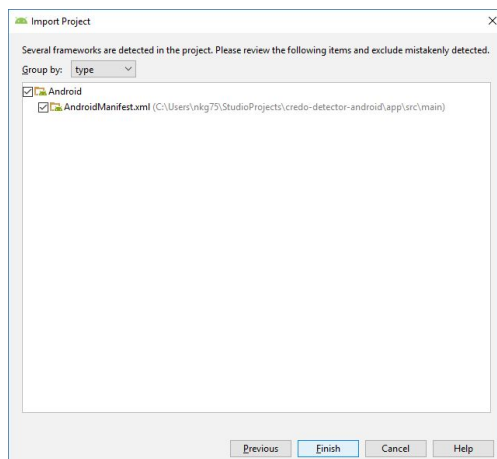
<https://github.com/credo-science/credo-detector-android.git>



# Android project import



## Download dependency by Gradle builder



# Prepare smartphone to working with Android Studio

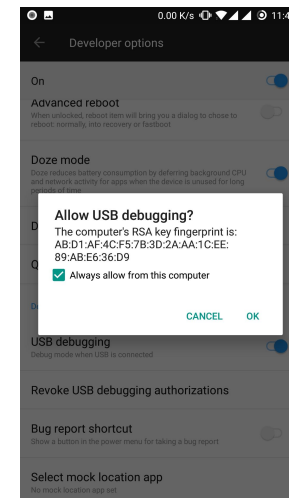
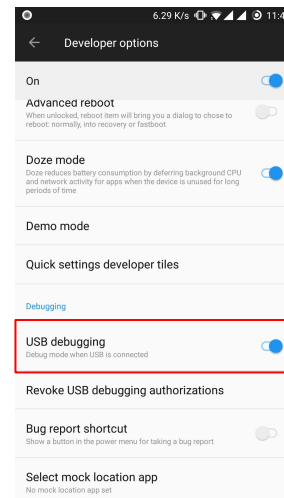
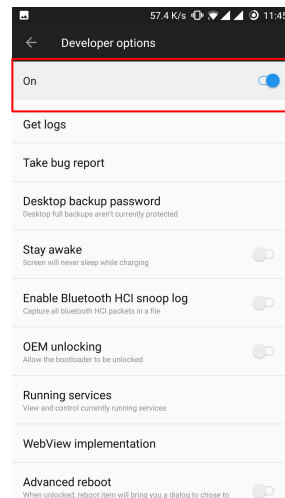
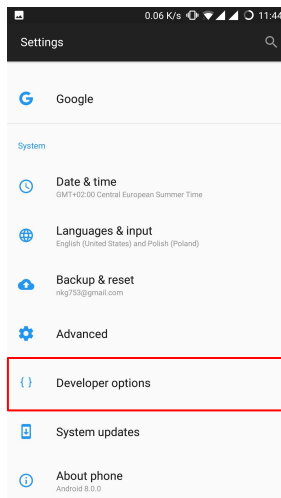
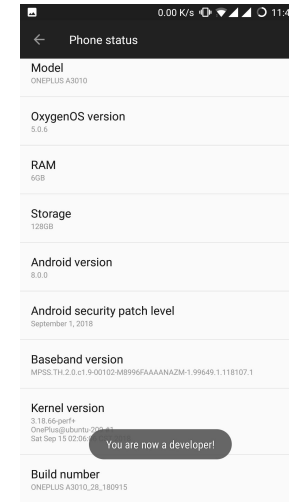
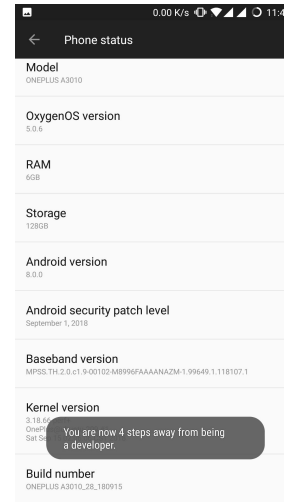
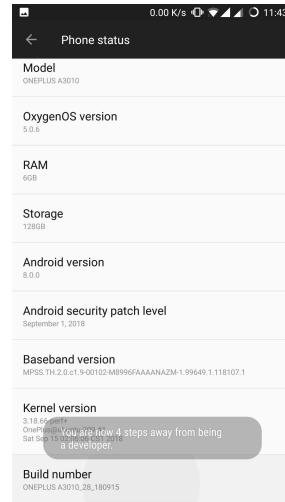
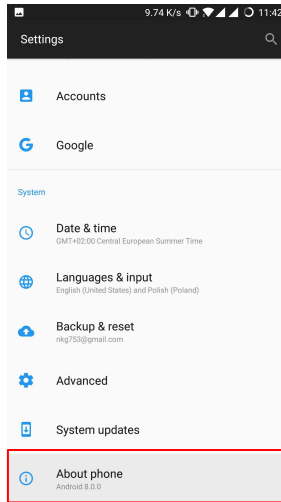
For ADB to recognize your device as a target for deploying debuggable APKs, you must first enable USB debugging in the on-device developer options.

Depending on the version of Android you're using, proceed as follows:

- On Android 8.0 and higher, go to **Settings > System > About phone** and tap **Build number** seven times.
- On Android 4.2 through 7.1.2, go to **Settings > About phone** and tap **Build number** seven times.

Return to the main **Settings** menu to find **Developer options** at the bottom. In the Developer options menu, scroll down and enable **USB debugging**.

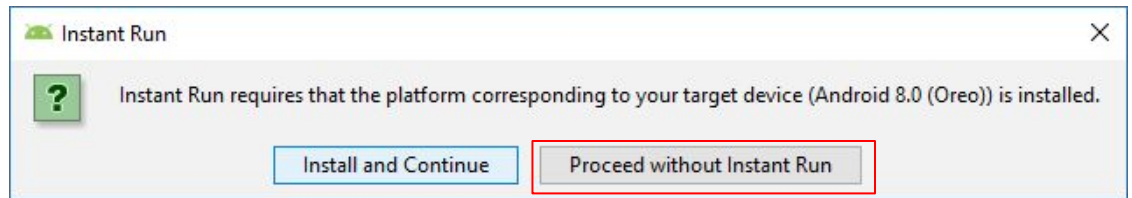
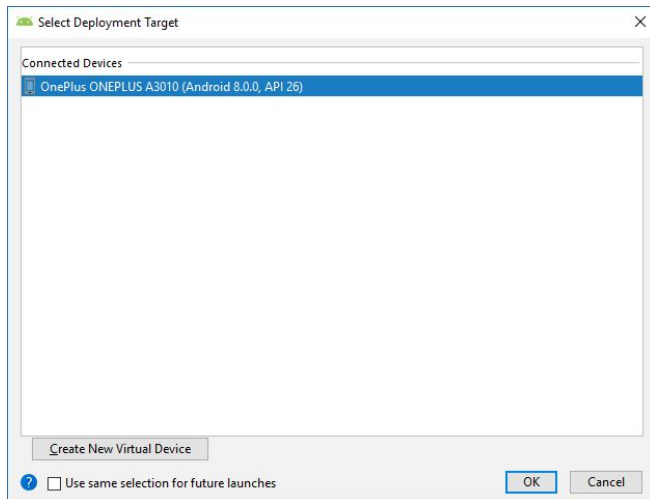
# Prepare smartphone: screen shots





# Run on smartphone

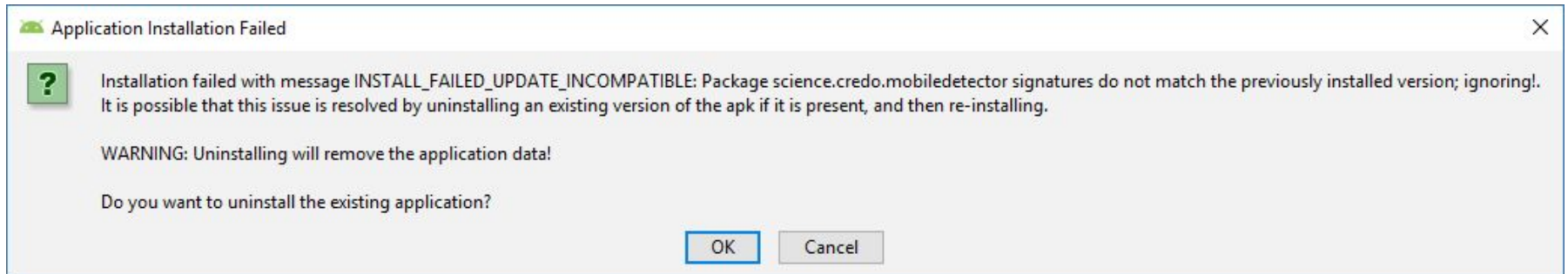
Click on menu **Run** → **Run ‘app’** and check your smartphone from list:



Instant Run is not necessary

# Troubleshooting

If you have previous installation from Google Play, you must uninstall it. Google Play apk is signed by CREDO Team key. Your apk will be signed by debug key generated during Android Studio installation.

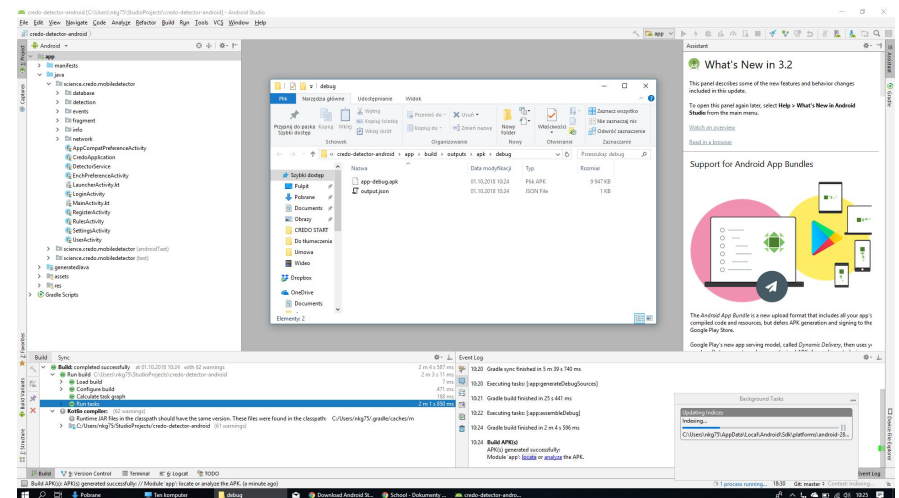
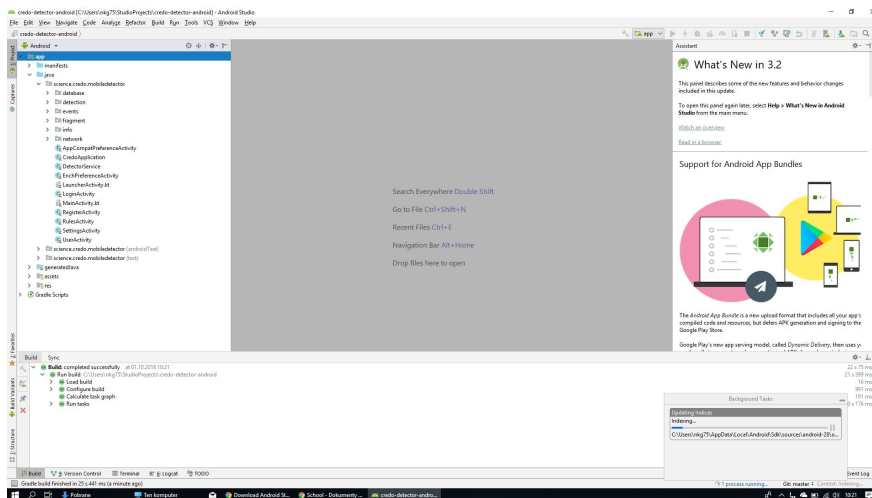


# Build APK

If you want build APK for share please clicks on menu:

**Build → Build bundles(s) / APK(s) → Build APK(s)**

The APK file will be stored in: **...\app\build\outputs\apk\debug**



# Important parts of source code

`.../science/credo/mobiledetector/detection/CameraPreviewCallbackNative.kt`

Contains `onPreviewFrame` method: analyse image from camera:

1. Save timestamp and run in async mode:
  - a. Make histogram of frame
  - b. If frame is darkness that **average factor** and black pixels is more than **black count** then go to find hits:
    - i. If brightest pixel is brighter than max factor, cut image with margin and send to server
    - ii. Fill cuted area by 0 bright pixel
    - iii. Do i. point again

`.../science/credo/mobiledetector/DetectorService.kt`

Setup camera and turn on.

# User interface localisation

.../res/values/strings.xml, .../res/values-pl/strings.xml,  
.../res/values-{lang}/strings.xml etc.

Interface translation. {lang} is the 2-letters language suffix. Android supports more specialized {lang} suffix i.e. “en-rGB”.

If you want make translation please create .../res/values-{lang}/ directory and copy string.xml from ...res/values/strings.xml. Now please change all messages like:

```
<string name="button_go_to_experiment">Go to experiment</string>
```

To:

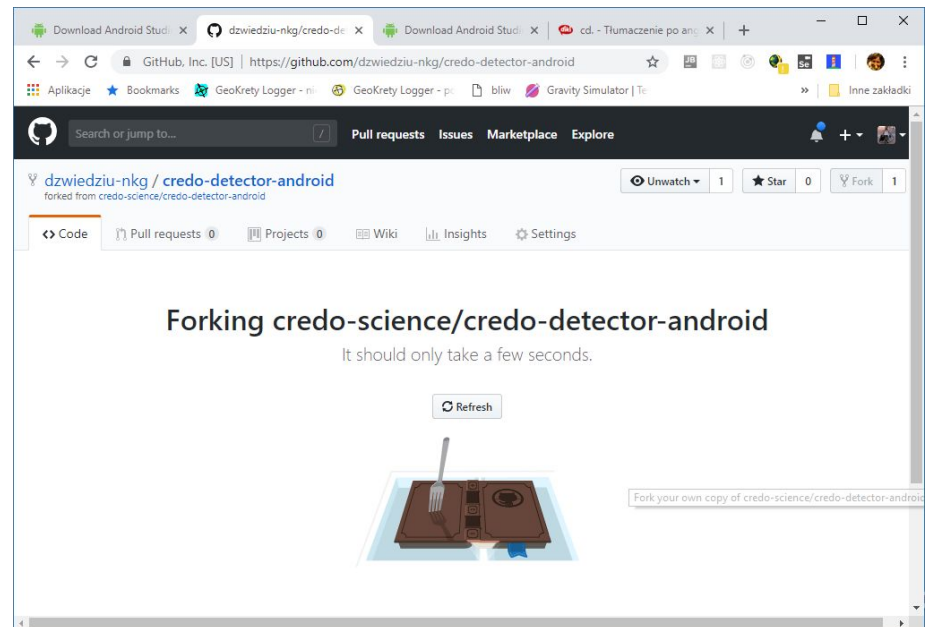
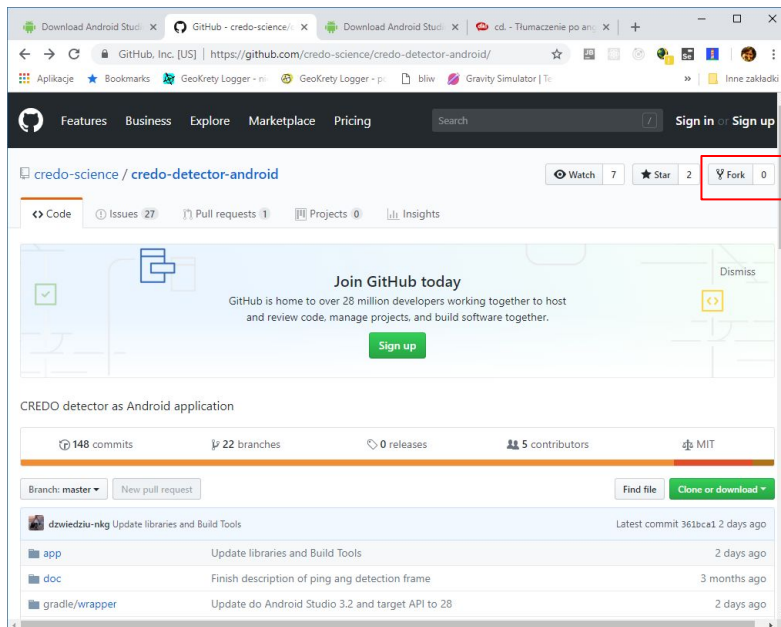
```
<string name="button_go_to_experiment">Uruchom eksperyment!</string>
```

Please test your localisation before sent to authors. Some messages may be too long and may be cut on interface. But you can fix the layout issue and send fixed whole project :)

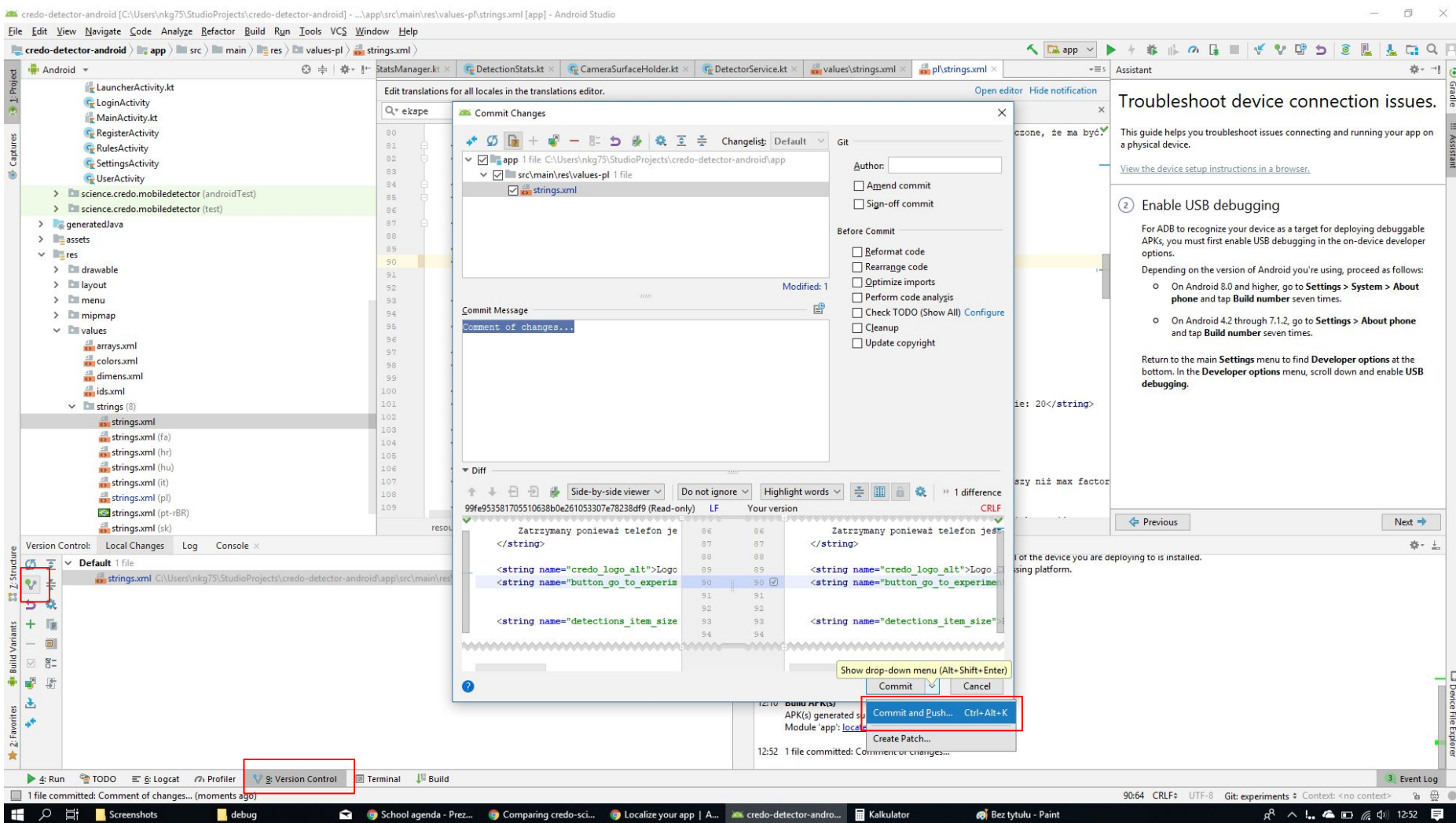
# Contribute in GitHub

Sending fixed source code to authors via e-mail is confusing. Best way is:

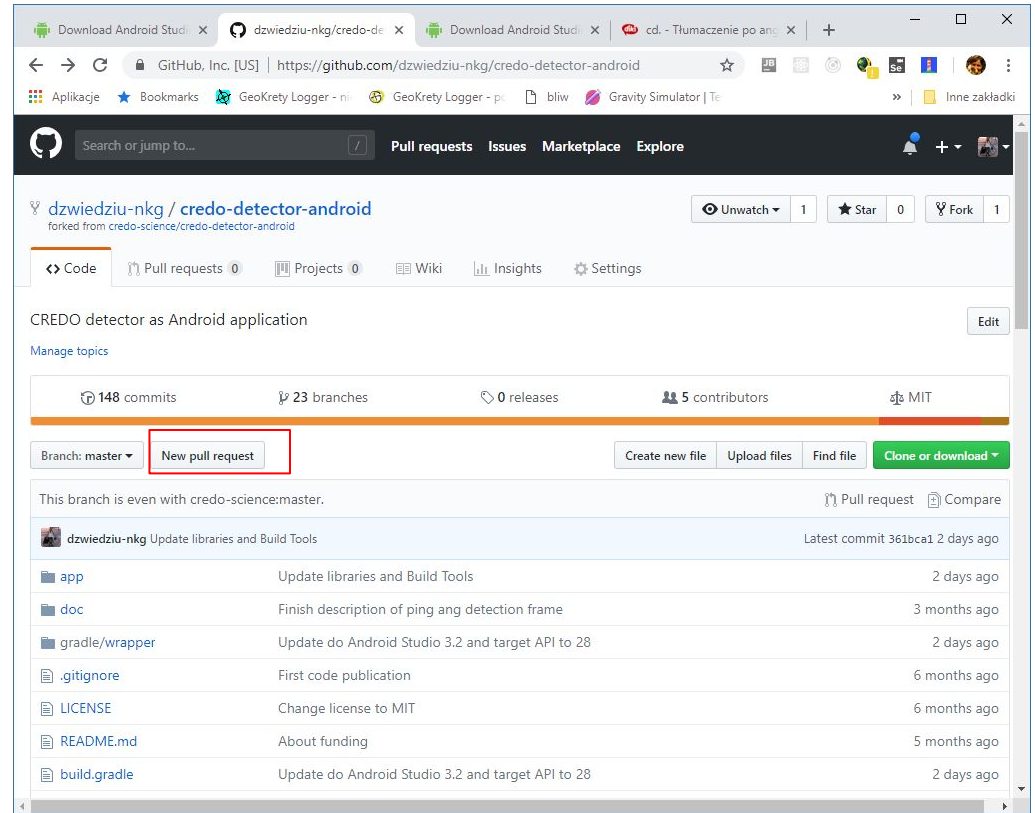
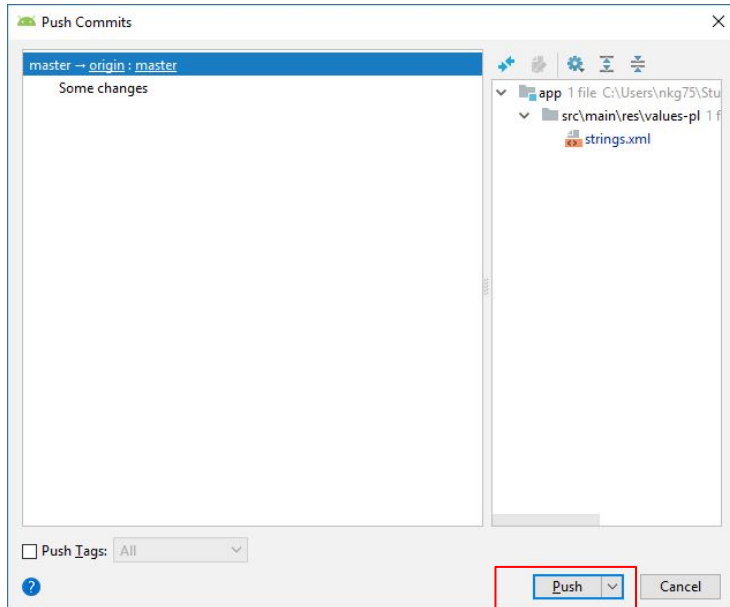
1. **Create** your GitHub account
2. Go to **Fork** project on your account
3. **Clone** project from your fork to Android Studio, make changes and **push** to GitHub
4. Make **pull request** on GitHub website to authors



## Commit and push changes to GitHub repository

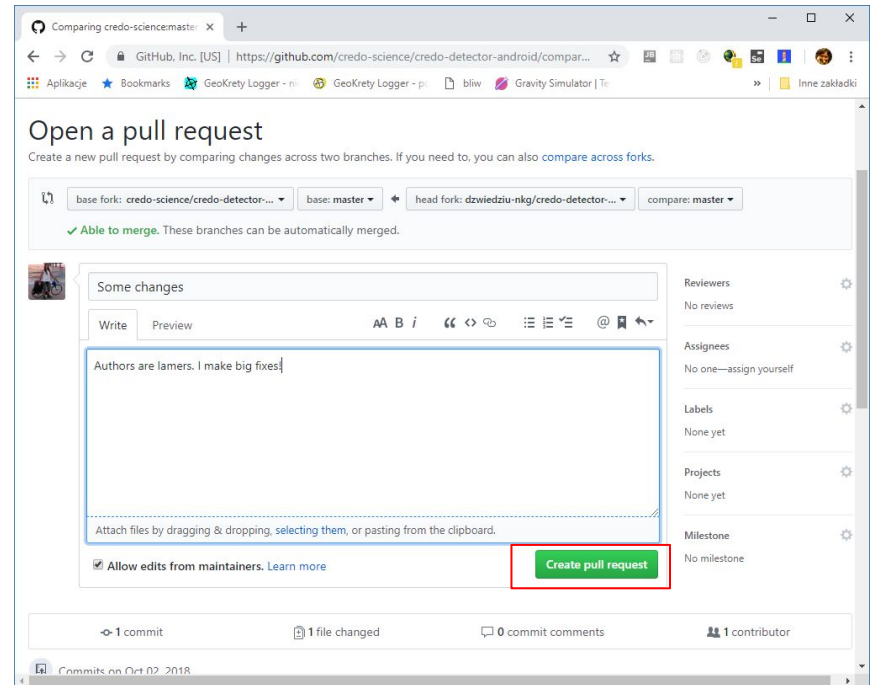
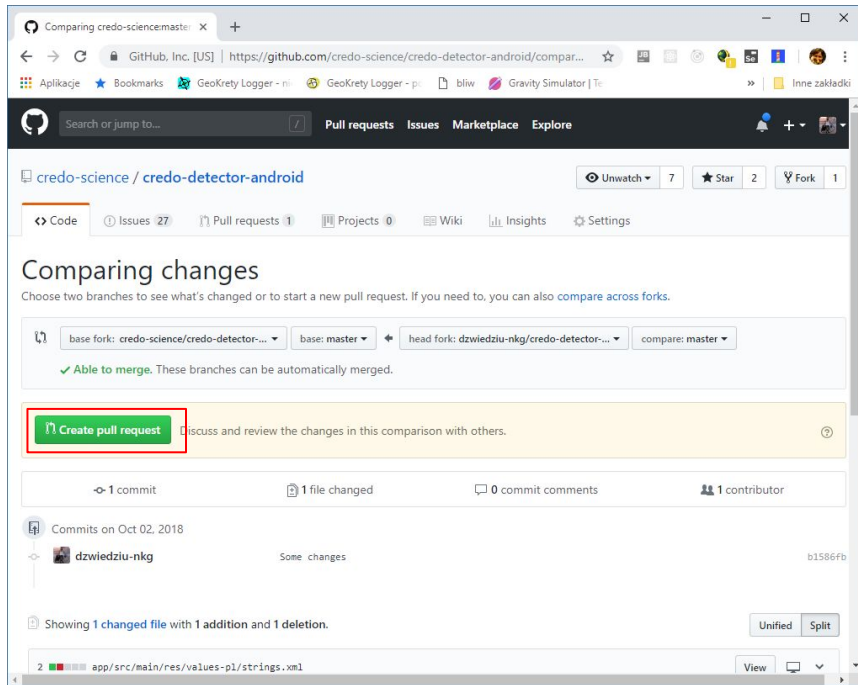


# Making push request to authors





# Sending push request to authors



# How we see your push request?

The screenshot shows a GitHub pull request interface. At the top, the browser address bar displays the URL `https://github.com/credo-science/credo-detector-android/pull/43`. The repository name `credo-science / credo-detector-android` is shown with 7 watchers, 2 stars, and 1 fork. The pull request title is `Some changes #43`, and the description states `dzwiedziu-nkg wants to merge 1 commit into credo-science:master from dzwiedziu-nkg:master`. A comment from `dzwiedziu-nkg` says, "Authors are lamers. I make big fixes!". The status bar indicates `+1 -1` changes. On the right, there are sections for `Reviewers` (No reviews), `Assignees` (No one—assign yourself), `Labels` (None yet), `Projects` (None yet), `Milestone` (No milestone), and `Notifications`. At the bottom, a green box contains the message `Continuous integration has not been set up` and `This branch has no conflicts with the base branch`. A red rectangle highlights the `Merge pull request` button.

Thanks for attention  
**and welcome to contribution!**