# Setting up a New ZAP Development Environment

PUBLIC - ~~anyone can comment~~
Comments now disabled.

**For the latest ZAP developer docs see https://www.zaproxy.org/docs/developer/**

**We are in the process of migrating content from this document to the ZAP website, so that is the first place you should look. This doc might have additional useful information but be aware that it may also be out of date.**

Still TODO:
- Migrate to the ZAP website :)

We are in the process of migrating ZAP to use Gradle.
This document explains how to set up a ZAP development environment for both existing ZAP contributors and those new to the project.
This is not the only way to upgrade your ZAP development environment, but it is the recommended way.

## The Main ZAP Repos

This section covers the main ZAP repos that you will need if you want to work on ZAP:

- https://github.com/zaproxy/zaproxy - the ZAP core
- https://github.com/zaproxy/zap-extensions - the ZAP extensions
- https://github.com/zaproxy/zap-hud - the ZAP HUD
- https://github.com/zaproxy/zap-core-help - the ZAP core help

## Preparation

### New Contributors

1. Create a GitHub account, if you don't already have one
2. Ensure that you have git installed locally

## Existing Contributors

1. Ensure that you do not have any open PRs. If you do then it is recommended that you get these merged first
2. Delete your GitHub forks of the above repos

## Fork the Repos on GitHub

Select each of the following repos and "Fork" to your GitHub account (the button is towards the top right of the screen)



- https://github.com/zaproxy/zaproxy
- https://github.com/zaproxy/zap-extensions
- https://github.com/zaproxy/zap-hud
- https://github.com/zaproxy/zap-core-help

## Clone the Repos Locally

Create a new local directory on your machine and run the following commands from the command line in that directory to clone your forks of the repos.
Replace "`<user>`" with your github username.

```
git clone https://github.com/<user>/zaproxy.git
git clone https://github.com/<user>/zap-extensions.git
git clone https://github.com/<user>/zap-hud.git
git clone https://github.com/<user>/zap-core-help.git
```

## Add the ZAP Remote Repos

Run the following commands from your command line in the directory you have just created.

Note that some of the lines have wrapped, but if you copy and paste them they should be fine.

```
cd zaproxy
git remote add upstream https://github.com/zaproxy/zaproxy.git
cd ..

cd zap-extensions
git remote add upstream https://github.com/zaproxy/zap-extensions.git
cd ..
```

```
cd zap-hud
git remote add upstream https://github.com/zaproxy/zap-hud.git
cd ..

cd zap-core-help
git remote add upstream https://github.com/zaproxy/zap-core-help.git
cd ..
```
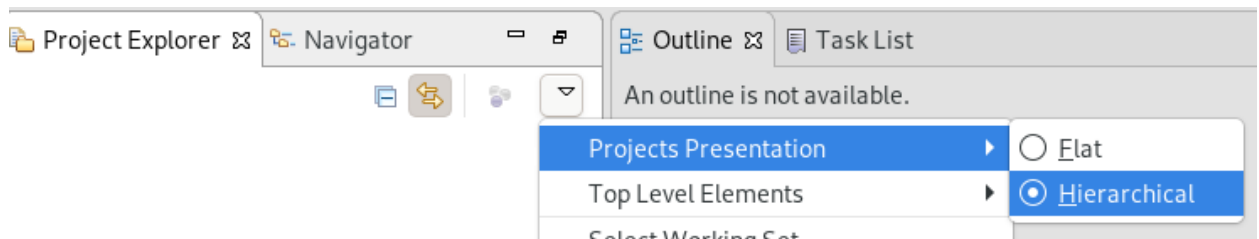
## Add the Repos to your IDE

In the following section "<dir>" refers to the directory you created above which contains all of your newly cloned repos.
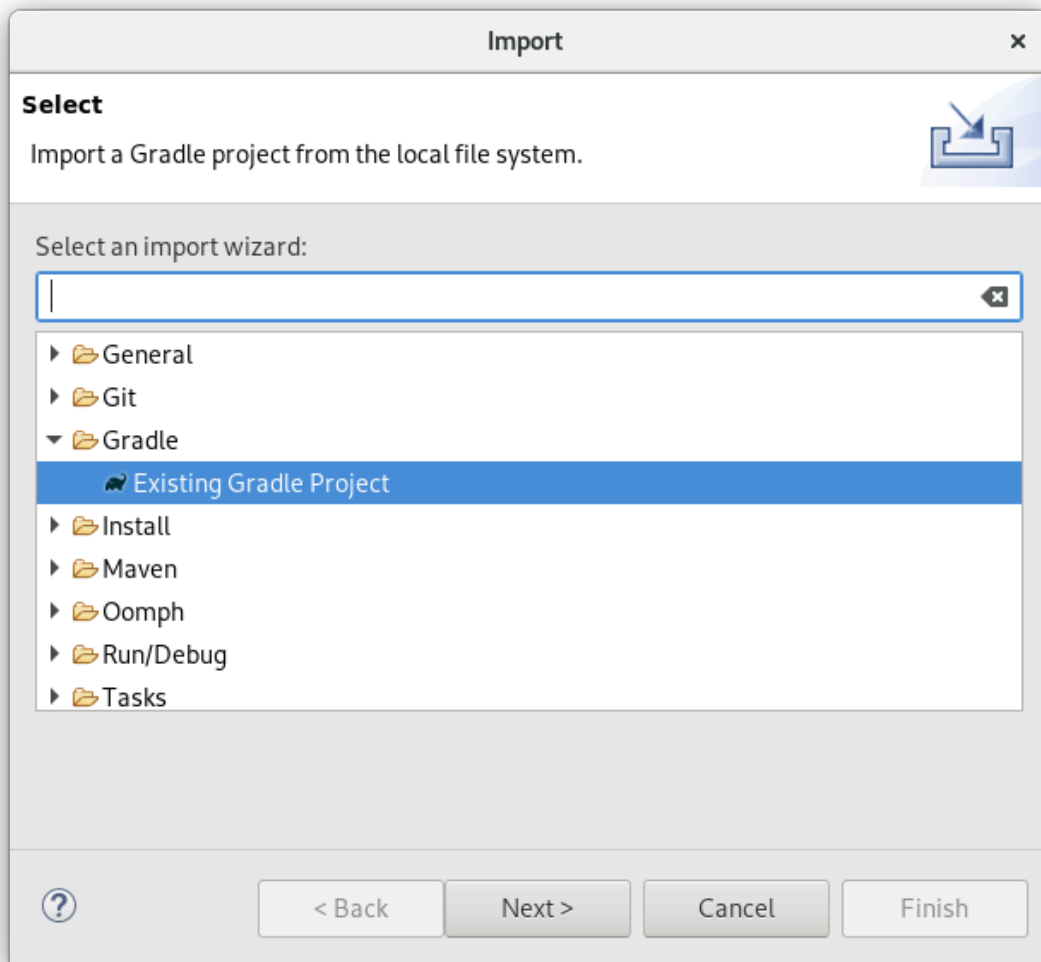
### Eclipse

1. Run Eclipse
2. Select <dir> as a new workspace
3. Change Projects Presentation to Hierarchical
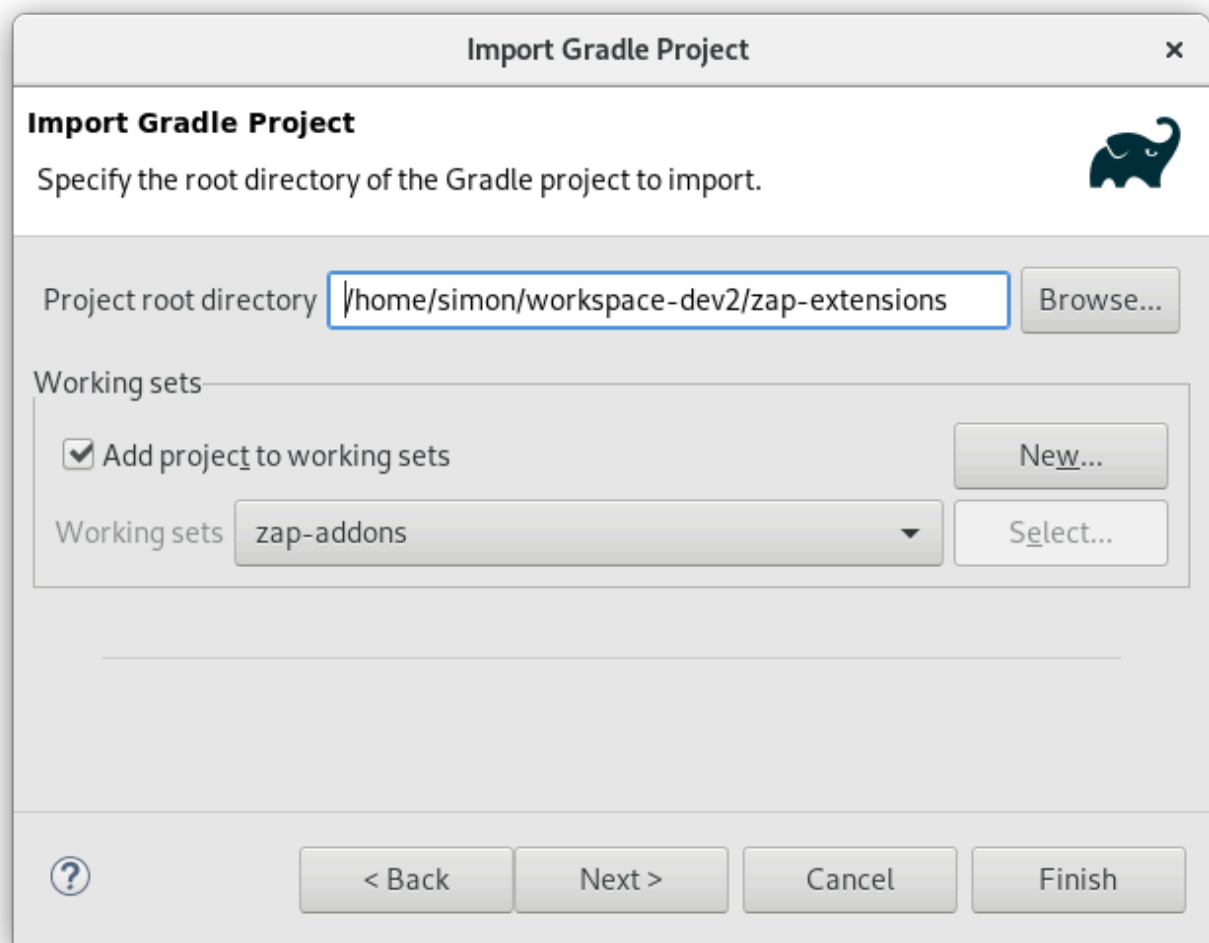


Import the zaproxy project:
1. Select menu: "File / Import…"
2. Select the option "Gradle / Existing Gradle Project"

3. In the Import Gradle Project dialog enter:
    ○ Project root directory: "<dir>/zaproxy"
    ○ Working sets
        ■ Add project to working sets
        ■ Working sets:
            1. Click the "Select" button
            2. Click "New"
            3. Select "Java"
            4. Click "Next >"
            5. Working set name: "zaproxy"
            6. Click "Finish"
            7. Click "OK"
            8. Click "Finish"  (This step may take quite a while depending on your connection speed, disk, etc.)

Import the zap-extensions project:

1. Select menu: "File / Import…"
2. Select the option "Gradle / Existing Gradle Project"
3. In the Import Gradle Project dialog enter:
   - Project root directory: "<dir>/zap-extensions"
   - Working sets
     - Add project to working sets
     - Working sets:
       1. Click "New..."
       2. Select "Java"
       3. Click "Next >"
       4. Working set name: "zap-addons"
       5. Click "Finish"
       6. Click "OK" (leaving "zap-addons" checked)
       7. Click "Finish" (This step may take quite a while depending on your connection speed, disk, etc.)
     - **Note**: Eclipse doesn't seem to always apply working sets correctly, if you import or switch/checkout branches and don't see something you expect to don't forget to check the "Other Projects" category in your Package Explorer, you might just need to assign the proper working set or drag & drop things.

Note that the following add-on may be flagged as having errors, this will not affect other addons:
● browserView: this will happen if the JRE you are using does not include JavaFX

Import the zap-hud project:

1. Select menu: "File / Import…"
2. Select the option "Gradle / Existing Gradle Project"
3. In the Import Gradle Project dialog enter:
    ○ Project root directory: "<dir>/zap-hud
    ○ Working sets
        ■ Add project to working sets
        ■ Working sets: "zaproxy"
        ■ Click "Finish"

Note that you can of course use a new working set if you want, just add it as before.

Import the zap-core-help project:

1. Select menu: "File / Import…"
2. Select the option "Gradle / Existing Gradle Project"
3. In the Import Gradle Project dialog enter:
   - Project root directory: "<dir>/zap-core-help
   - Working sets
     - Add project to working sets
     - Working sets: "zaproxy"
     - Click "Finish"

## Set Line Endings and Character Set

1. Select menu: "Window / Preferences"
2. Expand "General" and select "Workspace"
3. Set "Text file encoding" to "Other:" "UTF-8"
4. Set "New text file line delimiter" to "Other:" "Unix"
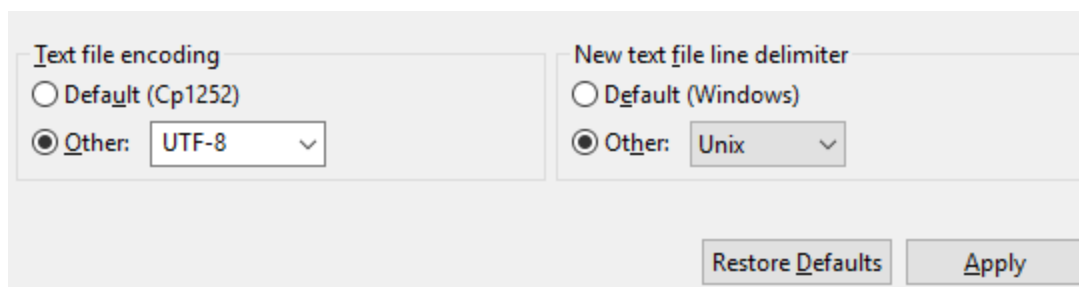


Note: If you open a PR (or push and do a compare) and all your content looks new, it's probably because of EndOfLine (EOL) conversion (maybe git, maybe you missed this step before doing work). You can use "File > Convert Line Delimiters To" in Eclipse to fix things up, commit, and push again.

## IntelliJ

TODO

# Deploy the main ZAP add-ons

Note that the following task may fail if the webdrivers fail to download. If that happens try running the task again.

## Command Line

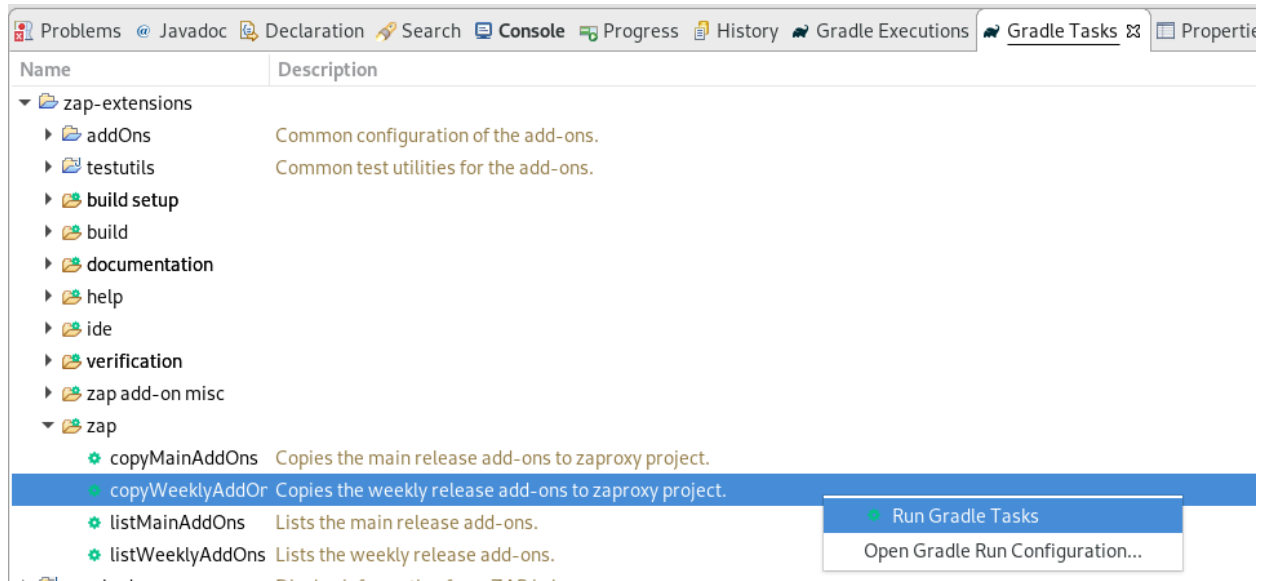Run the following command in the zaproxy directory:

```
./gradlew copyWeeklyAddOns
```

# Eclipse

In the Gradle Tasks tab select:
- zaproxy
  - zap
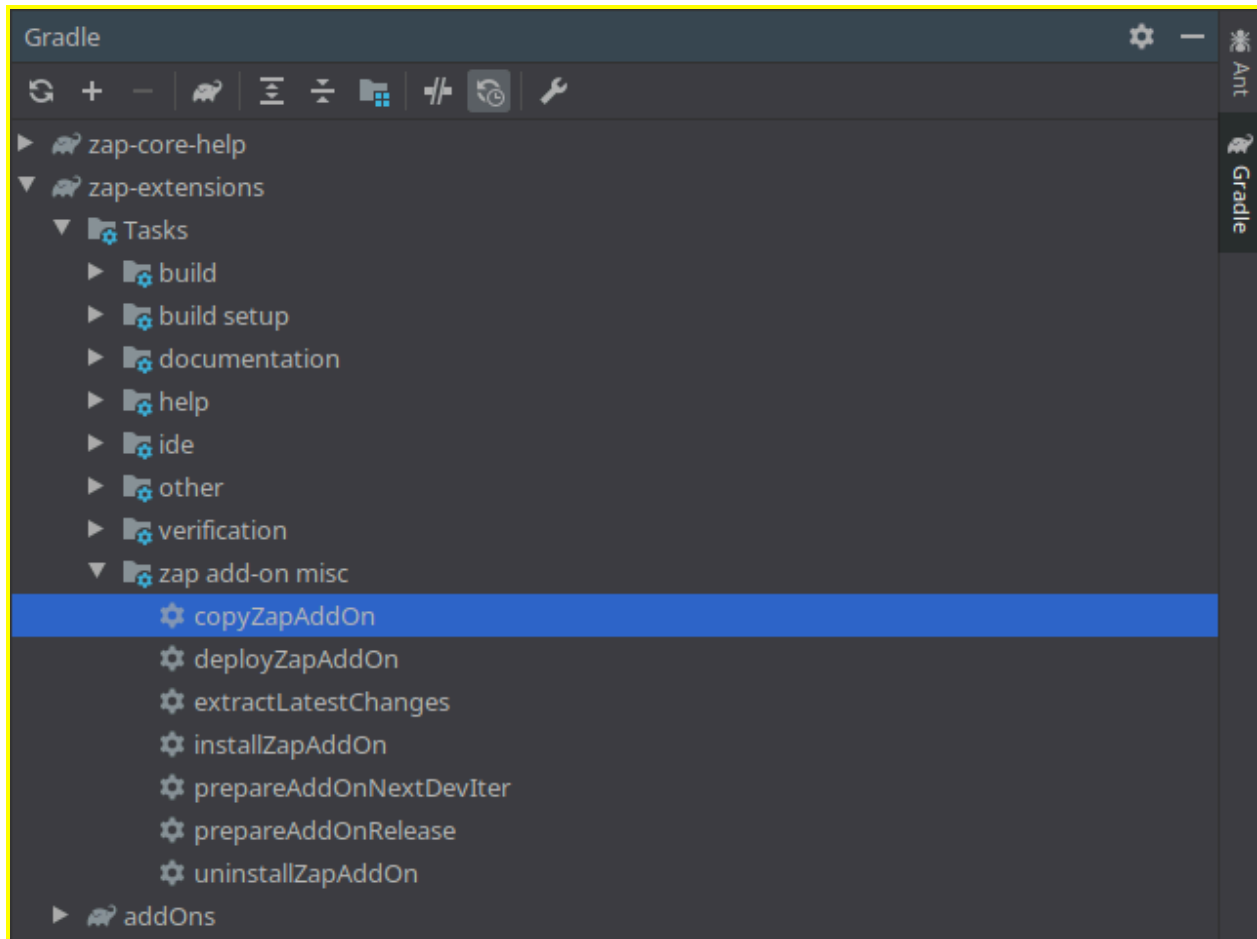    - copyWeeklyAddOns

Right click it and select "Run Gradle Tasks"



# IntelliJ

In the Gradle tab:
- Zap-extensions
  - Tasks
    - zap add-on misc
      - copyZapAddOn

Right click on it and Run it.

# Deploy the HUD

## Command Line

Run the following command in the zap-hud directory:
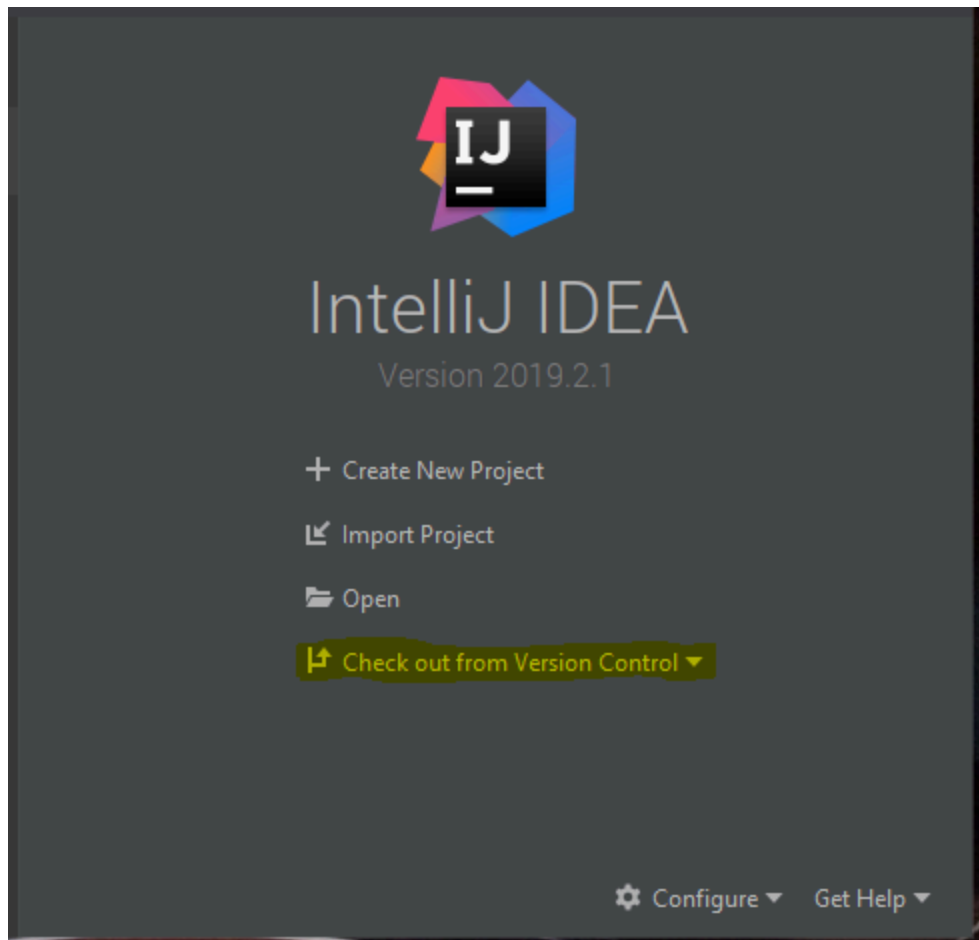
```
./gradlew deployZapAddOn
```

## Eclipse

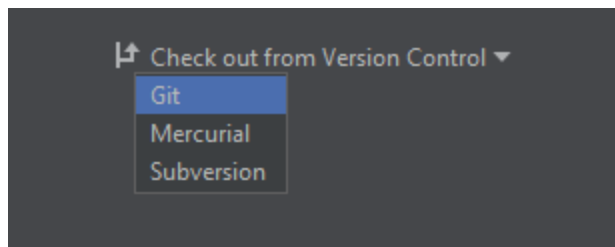In the Gradle Tasks tab select:
- zap-hud
  - deployZapAddOn

## IntelliJ

IntelliJ has the ability to clone from GitHub and open and configure the gradle project in one step.

Once your projects are forked, as shown above, you can click the "Check-out from Version Control" option on the welcome screen.
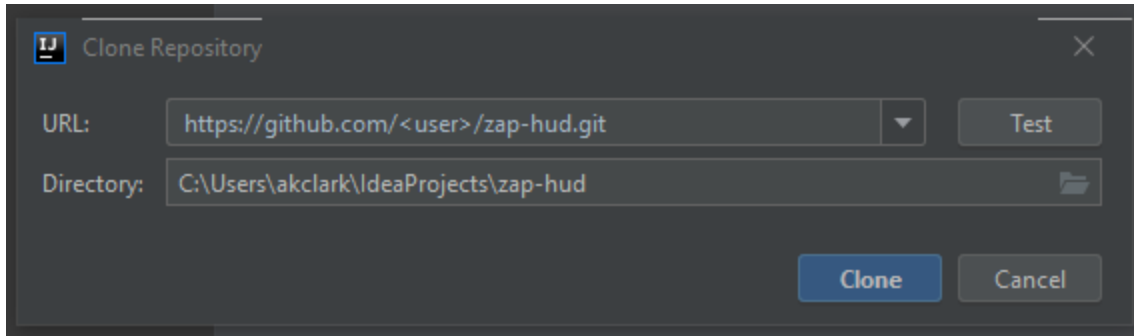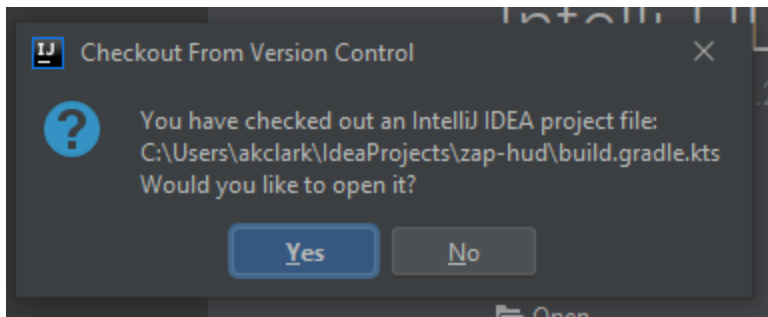


You will select the "Git" option.



And put in the link to your repo.(Change <user> to your username!)
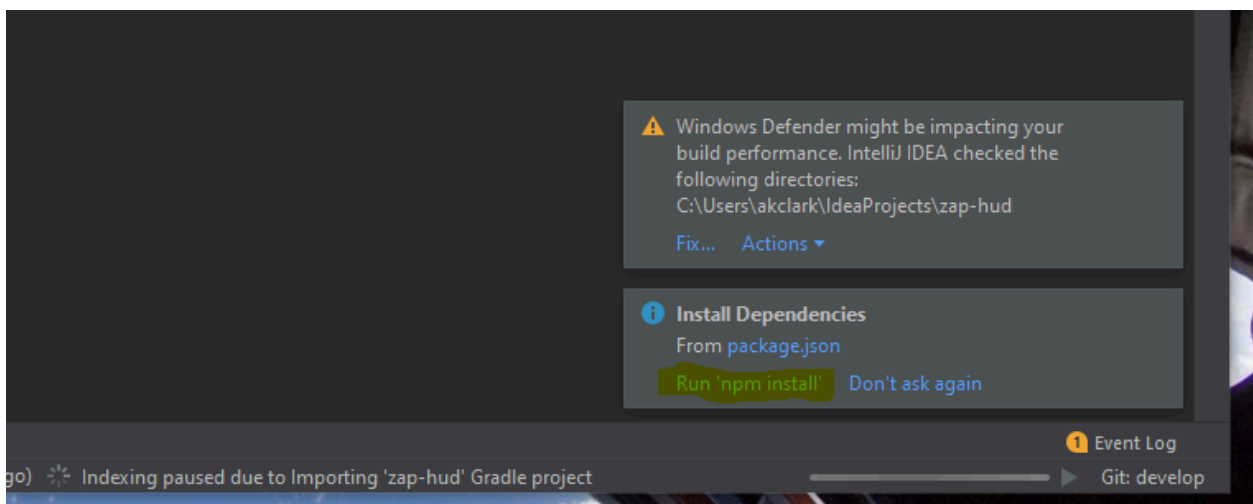** You will need to do this for all four of your forked repos.

And then click the "Clone" button.

Once checked out, IntelliJ will auto-detect the gradle file and show this dialog box.



Click "Yes"

If not previously installed, you may see some "Install Dependencies" dialogs in the bottom corner of the screen. Click the "Run 'npm install'" link.



You should now have a buildable project.

From the "Build" menu, if you click "Build Project", once completed you should see the following (or something similar.)

```
BUILD SUCCESSFUL in 54s
6 actionable tasks: 6 executed
3:23:09 AM: Tasks execution finished ':classes :testClasses'.
```

## Running ZAP

TODO

# The Supplementary ZAP Repos

The following ZAP repos contain additional functionality which is less frequently changed.
TODO

# Making Changes

## Changing the ZAP core

TODO

## Changing an Add-on

TODO

## Creating a New Add-on

TODO

## Changing the HUD

TODO