

# Putting the app on the phone\*

---

\*Only works for android. Guides for iOS (not tested provided at the end of the file).

## Prerequisites:

Your app is running as expected on the laptop. In the same folder as the `main.py` and `DigiCert.crt.pem` start a terminal and activate your python virtual environment. Make sure you have the `buildozer` library installed (hint try `pip list`) with pip (if not do `pip install buildozer`).

You have a java jdk installed, you can do this command in the terminal: `java -version`. If not please install from: <https://www.oracle.com/java/technologies/downloads/#java17> (tested for java jdk 17)

## First time setup

---

### Steps:

1. In the terminal run: `buildozer init`
2. Now you have a `buildozer.spec` file in your folder. It contains a list of configuration tokens. All lines starting with `#` are ignored. You are interested in having the following lines uncommeted and filled out correctly, find them in the `.spec` file:

```
title = CloudBased

package.name = cloudbased

package.domain = dk.diku

source.dir = .

source.include_exts = py,png,jpg,kv,atlas,pem,crt.pem

version = 1.0

requirements = python3,kivy,xmldict,mysql-connector-python

orientation = portrait

android.api = 31

android.accept_sdk_license = True

android.enable_androidx = False

p4a.branch = develop
```

The other configuration tokens stay the same. More details at:

<https://buildozer.readthedocs.io/en/latest/specifications.html#specifications>

3. The `httpx` library we used so far for the DCR Repository REST API is not compatible with `python-for-android` (the library that manages the apps in the android app) so we need to switch from `httpx` to `requests` (they do both handle http requests). Check again with `pip list` in the terminal if you have it, otherwise do `pip install requests`.
4. Now open the `main.py` file. We have to do some code housekeeping. Change the line `import httpx` to `import requests`. Now in each place where you did `httpx.[get,post,delete]` you need to do the following:

```
# replace in the method get_enabled_events
next_activities_response =
httpx.get("https://repository.dcrgraphs.net/api/graphs/" + graph_id +
"/sims/" + sim_id + "/events?filter=only-enabled", auth=auth)
# with
req = requests.Session()
req.auth = auth
next_activities_response =
req.get("https://repository.dcrgraphs.net/api/graphs/"
        + graph_id + "/sims/" + sim_id + "/events?
filter=only-enabled")

# replace in the method execute_event
url = (f"https://repository.dcrgraphs.net/api/graphs/{self.graph_id}/sims/"
        f"{self.simulation_id}/events/{self.event_id}")
response = httpx.post(url, auth=(self.username, self.password))
# with
url = (f"https://repository.dcrgraphs.net/api/graphs/{self.graph_id}/sims/"
        f"{self.simulation_id}/events/{self.event_id}")
auth=(self.username, self.password)
req = requests.Session()
req.auth = auth
req.post(url)

# replace in the method start_sim
self.current_auth = (self.username_textinput.text,
self.password_textinput.text)
newsim_response = httpx.post(

url=f"https://repository.dcrgraphs.net/api/graphs/{self.graph_id_input.text
}/sims",
    auth=self.current_auth)

self.simulation_id = newsim_response.headers['simulationID']
# with
self.current_auth = (self.username_textinput.text,
self.password_textinput.text)

url=f"https://repository.dcrgraphs.net/api/graphs/{self.graph_id_input.text
}/sims"
```

```

        auth=self.current_auth
        req = requests.Session()
        req.auth = auth
        resp = req.post(url)
        resp_headers = resp.headers
        if resp_headers:
            self.simulation_id = resp_headers['simulationID']

# replace in the method stop_sim
response = httpx.delete(

url=f"https://repository.dcrgraphs.net/api/graphs/{self.graph_id_input.text
}/sims/{self.simulation_id}",
        auth=self.current_auth)

# with
url=f"https://repository.dcrgraphs.net/api/graphs/{self.graph_id_input.text
}/sims/{self.simulation_id}"#,
auth=self.current_auth
req = requests.Session()
req.auth = auth
req.delete(url)

```

If you have any other places where you are calling the DCR repository with `httpx` make sure you also replace it with the `requests` library in a similar way.

More details of how requests works at: <https://requests.readthedocs.io/en/latest/>

5. Check that the app works on the desktop and all functionality is there.
6. Connect your phone as file transfer via a USB cable to your laptop. Approve any requests on the phone (optionally also enable debugging mode in android settings).
7. In the terminal run: `buildozer -v android debug`. The first time you run this command it creates the pipeline for publishing the app (it will take some time so grab a coffee, lunch or a nap, depending on your laptop).
8. In the terminal run: `buildozer -v android deploy run logcat`

Tested on linux. You might need some extra steps and libraries for OS X and Windows.

If you have any errors with the buildozer build or deploy commands check that you have the right Java version and gradle libraries installed on your machine and visible in your `PATH` environment variable. Hint check the: `.buildozer/android/platform/build-arm64-v8a_armeabi-v7a/dists/cloudbased/build.gradle` file to see which Java major version and gradle version buildozer expects.

## After the first run

---

After the first run if you wish to make changes to the app you only need to run these steps again to update the app on your phone. In the same folder as the `main.py` and `DigiCert.crt.pem` start a terminal and activate your python virtual environment.

1. In the terminal run: `buildozer -v android debug`. This updates the code.
2. In the terminal run: `buildozer -v android deploy run logcat`. This publishes the app to your phone.

More details at: <https://kivy.org/doc/stable/guide/packaging-android.html>

## For iOS (not tested)

---

After doing only the code changes from the android guide you can try packaging your app on iOS. This is only possible on a Macbook, plus you must have Xcode and homebrew installed. Your `main.py` and `DigiCert.crt.pem` project files stay the same.

Prerequisites: <https://kivy.org/doc/stable/guide/packaging-ios-prerequisites.html>

Guide: <https://kivy.org/doc/stable/guide/packaging-ios.html>