1. Setup phase

First, we clone everything and install dependencies

```
# Clone required repositories

git clone https://github.com/ktdumper/ktdumper
git clone https://github.com/ktdumper/fs-tools
git clone https://github.com/usernameak/keitai_fs_tools.git

# Run dependency setup

sudo apt install python3 dub python3-usb python3-tqdm gcc-arm-none-eabi gcc-arm-linux-gnueabi qemu qemu-user-static qemu-system-arm

pip3 install qiling
```

Then, we compile programs and get environments setup

```
cd ./fs-tools/ssr200
python3 -m venv ./
source bin/activate
pip3 install git+https://github.com/nathanhi/pyfatfs.git
patch -p1 ./lib/python3.10/site-packages/pyfatfs/PyFat.py pyfatfs.diff

cd ../xsr2/
arm-linux-gnueabi-gcc xsr_stl.ko main.c -o main

cd ../fsr_ll/
arm-linux-gnueabi-gcc -static -omain fsr.ko fsr_stl.ko main.c

cd ../../keitai_fs_tools/xsr1/
dub
```

2. Operation phase
    1. ktdumper

       1. Normal behaviour

       - Parse phones and have selectable dropdown of models and methods from `${ktdumper_root}/ktdumper/devices.py`. Format:
       `` `Device("MODEL_NAME", VID, PID, {        "METHOD_NAME": ... ,     },` `... , - Wait for that vid/pid to connect from lsusb (Have a refresh button`

`to find that using`lsusb | awk '{print $6}'`) 1. Possible errors -` RuntimeError("Could not locate {} in supported devices list".format(args.phone))`. Happens when a device isn't found in the list. Can be mitigated by having a dropdown of models. -`usb.core.USBError: [Errno 110] Operation timed out`. Happens whenever ?(not enough current on USB port).

2. fs-tools
   1. xsr1
      1. Normal behaviour
      - Usage: `rfs_dumper_xsr1app input.bin output.bin`
      - Do a check on a bin to find `91 5A 49 2C` header and strip everything before the first occurrence. Else, reject.
   2. ssr200