# Debug and run UT locally

## [Set up the env and start the flask server](#DebugandrunUTlocally-Setuptheenvandstar)

## [Start a Python (>=3.8.5) virtual env for you local Control Center server](#DebugandrunUTlocally-StartaPython(>=3.8)

## [if you don't have that virtual env, you can create a new one by below command](#DebugandrunUTlocally-ifyoudon'thavethat)

## [Install dependencies in deploy/requirements.sh](#DebugandrunUTlocally-Installdependencie)

## [Set up PYTHONPATH env](#DebugandrunUTlocally-SetupPYTHONPATHenv)

## [Start the flask server](#DebugandrunUTlocally-Starttheflaskserve)

## [Setup Redis and start celery worker](#DebugandrunUTlocally-SetupRedisandstart)

## [Setup Flower](#DebugandrunUTlocally-SetupFlower)

## [Run UT locally](#DebugandrunUTlocally-RunUTlocally)

## Set up the env and start the flask server

### Start a Python (>=3.8.5) virtual env for you local Control Center server

source <virtualenv\_dir>/bin/active

### if you don't have that virtual env, you can create a new one by below command

virtualenv [-p <python\_lib>]  [<virtualenv\_dir>]

### Install dependencies in deploy/requirements.sh

pip3 install -r deploy/requirements.sh

### Set up PYTHONPATH env

# cd to the root path of the project

export PYTHONPATH=".

:$PYTHONPATH"

### Start the flask server

# Create a dir named "log" in the root path of the project

python ./server/app.py

## Setup Redis and start celery worker

Install Redis locally

Start the celery worker

# cd to the root path of the project

celery -A task.tasks worker -l INFO -f ./log/worker.log

## Setup Flower

Start flower

# cd to the root path of the project

celery -A task.tasks flower -p 5555

# URL: https://localhost:5555

## Run UT locally

Please set up the configuration in PyCharm as below

Using Script path for your test case

Set up ENV variables

Choose your own virtualenv python lib as the python interpreter

