Continuous Integration using *CircleCI* for Android

Create a circle.yml in your project's root directory. If this file exists, CircleCI will read it each time it runs a build.

➢ Enable *CircleCI*

1. Login on CircleCI https://circleci.com/ -> Log In with bitbucket or Log In with GitHub
2. Follow the project on which you want Circle CI to run.
   Note : To follow the project, you have to be admin.

➢ Configuring *CircleCI*

1. File Structure and content

The circle.yml file has seven primary sections. Each section represents a *phase* of the Build-Test-Deploy process:

- **machine:** adjust the behavior of the virtual machine (VM)
- **checkout:** checkout and clone code from a repository
- **dependencies:** install your project's language-specific dependencies
- **database:** prepare a database for tests
- **compile:** compile your project
- **test:** run your tests
- **deployment:** deploy your code to your web servers

A circle.yml can also contain 2 optional sections that are not linked to specific phases:

- **general:** use for broader build-related configuration
- **experimental:** test out features that are currently in development

Each section consists of settings and/or Bash commands, which are run in the order they appear in the file.

On completion, each command will return an exit code: 0 is a success, and any other number is a failure.

Note : If a command fails during any of the setup sections (machine, checkout, dependencies, database, compile), the entire build will fail early. If a command fails in the test section, the build will continue to run.

You can specify when to run custom commands relative to CircleCI's inferred commands using three special keys:

- pre: run *before* inferred commands
- override: run *instead of* inferred commands
- post: run *after* inferred commands

2. Machine configuration

```
machine:
  environment:
    ANDROID_HOME: /usr/local/android-sdk-linux
```

Note : **pre** and **post** are supported in the machine section, **override** is not.

3. Checkout configuration

```
checkout:
  post:
    - git submodule sync
    - git submodule update --init
```

Note : only **post** is supported in the this section because we don't read circle.yml until after the checkout phase.

4. Dependencies configuration

CircleCI automatically runs commands to fetch dependencies.

```
dependencies:
    pre:
      - echo y | android update sdk --no-ui --all --filter "platform-tools,build-tools-25.0.2,android-25,sys-img-armeabi-v7a-android-25,extra-android-support,extra-google-google_play_services,extra-google-m2repository,extra-android-m2repository"
```

5. Compile configuration

You can customize your project’s build in the compile section of circle.yml.

```
compile:
   override:
        # Generate a debug-apk
      - ./gradlew assembleDebug
```

6. Test configuration

All test commands will run, even if one fails.

```
test:
   post:
      - mkdir $CIRCLE_ARTIFACTS/apks
      - curl -L -o gdrive "https://docs.google.com/uc?id=0B3X9GlR6EmbnQ0FtZmJJUXEyRTA&export=download"; chmod +x gdrive
      - ./gdrive --refresh-token "${GDRIVE_REFRESH_TOKEN}" upload --parent "${GDRIVE_DIR_ID}" app/build/outputs/apk/app-debug.apk
```

7. Deployment configuration

The deployment section is optional. You can run commands to deploy to staging or production. These commands are triggered only after a successful (green) build.

```
deployment:
  beta:
     branch: Production
     commands:
        - fastlane android beta
```

In the following example, if a developer pushes to the *Production* branch, the command *fastlane android beta* runs.

Note : The deployment section doesn’t support sub-sections such as *pre* that you might find elsewhere in circle.yml.

8. Tags configuration in Deployment

In addition to deploying based on branch, you can deploy based on tags.

```
deployment:
  beta:
    tag: /release-.*/
    owner: circleci
    commands:
      - fastlane android beta
```

In the following example, if a developer pushes a tag named *release-v1.05*, the command *fastlane android beta* runs.

Note : Change owner from circleci to the username or organization that the repo belongs to or your build won’t run.

9. General configuration

CircleCI by default tests every push to *any* branch in the repository.

You may, however, blacklist branches from being built in CircleCI. This example excludes gh-pages from being built in circle:

```
general:
   branches:
     ignore:
       - gh-pages        # list of branches to ignore
       - /release\/.*/   # or ignore regexes
```

You may also whitelist branches, so only whitelisted branches will trigger a build.

This example limits builds in CircleCI to Development and feature-.* branches:

```
general:
  branches:
    only:
      - Development     # list of branches to build
      - /feature-.*/        # or regexes
```

➢ Example file of *CircleCI*

```
general:
  branches:
    only:
      - Development # or regexes
      - Production

 artifacts:
   - app/build/outputs/apk/

dependencies:
    pre:
       # Install the android packages
     - echo y | android update sdk --no-ui --all --filter "platform-tools,build-tools-25.0.2,android-25,sys-img-armeabi-v7a-android-25,extra-android-support,extra-google-google_play_services,extra-google-m2repository,extra-android-m2repository"
     - gem install fastlane
       # Test lanes
    - fastlane lanes
    - sudo apt-get install sendmail
   cache_directories:
    - "/opt/circleci/.rvm/gems"

compile:
  override:
       # Generate a debug-apk
      - ./gradlew assembleDevelopmentDebug

test:
  post:
     - mkdir $CIRCLE_ARTIFACTS/apks
          # Download gdrive and give user the execute permissions
     - curl -L -o gdrive "https://docs.google.com/uc?id=0B3X9GlR6EmbnQ0FtZmJJUXEyRTA&export=download"; chmod +x gdrive
         # Upload file to gdrive
     - ./gdrive --refresh-token "${GDRIVE_REFRESH_TOKEN}" upload --parent "${GDRIVE_DIR_ID}"  app/build/outputs/apk/app-development-debug.apk

deployment:
     beta:
         branch: Production
         commands:
             - fastlane android beta
```

In the above example, the when the code is pushed on *Development* or *Production* branch, circleci will start triggering the build. Also the *build(apk)* is created and uploaded on *google drive*. If the code is merged on *Production* branch, *fastlane* will upload the build to beta in *google play store*.

For uploading to google drive -

Install gdrive from terminal :

```
brew install gdrive

gdrive info "${GDRIVE_DIR_ID}"

Enter the verification code from the url given

After entering the verification code, it will generate the .gdrive folder in home directory.
```

GDRIVE_REFRESH_TOKEN is taken from the json file (hidden file) present in ./gdrive folder of home directory

GDRIVE_DIR_ID is the parent id of the folder from google drive.

➢ References

- https://circleci.com/docs/1.0/configuration/
- https://github.com/prasmussen/gdrive