

\$appid_build.yml文件说明

在**项目仓库根目录**下提供\$appid_build.yml文件(\$appid使用你自己应用的appid进行替换)

发布系统的ci部分, 通过挂了webhook的仓库在发生代码push后, 获取此文件进行编译

想要某个分支走发布系统那么这个分支下的根目录必须有一个\$appid_build.yml文件, 并且这个文件里面的branch部分必须有这个分支的名字

```
language: java
version: oraclejdk7
os: centos6.5
before_script:
  - echo ' before_script'
  - sleep 1
install_script:
  - whereis java
  - echo 'install_script'
script:
  - mvn install -Dmaven.test.skip=true -B
after_success:
  - sh web/ci_success.sh
after_failure:
  - echo 'after_failure'
outfile:
  - outfile/
notifications:_
  email:
    - yunfeng.wu@ele.me
sonar_src:
  - web/
  - common/
branch:
  '(master|dev|release|master)': alpha_stable
build_image:
  imageOutBuildScript:
    - mvn install -Dmaven.test.skip=true -B
    - sh web/ci_success.sh
  addToImageFile:
    - outfile/_____
  fromImage: docker.ele.net.me/mirror/java:openjdk-8u77-jdk-alpine
  dockerFileRun:
    - mkdir /111 && mkdir /222
```

language和version定义编译的语种信息, 支持语言和版本的组合见[这里](#)

script部分你可以放置进行项目编译的脚本 (必须)

after_successscript

outfile

```
notificationsci
```

```
branchpushalpha_stablepushcinull
```

branch部分支持精准正则匹配

branch部分如果是纯数字的branch，需要在branch 名上加单引号

```
build_imagebuild docker image
```

```
imageOutBuildScript
```

```
addToImageFileADDimage,/data/$appid
```

```
fromImageimage
```

```
dockerFileRundockerfilerun
```

生成的docker image的Entrypoint和COMMAND没有特别指定的地方，需要你在基础镜像中预先指定好