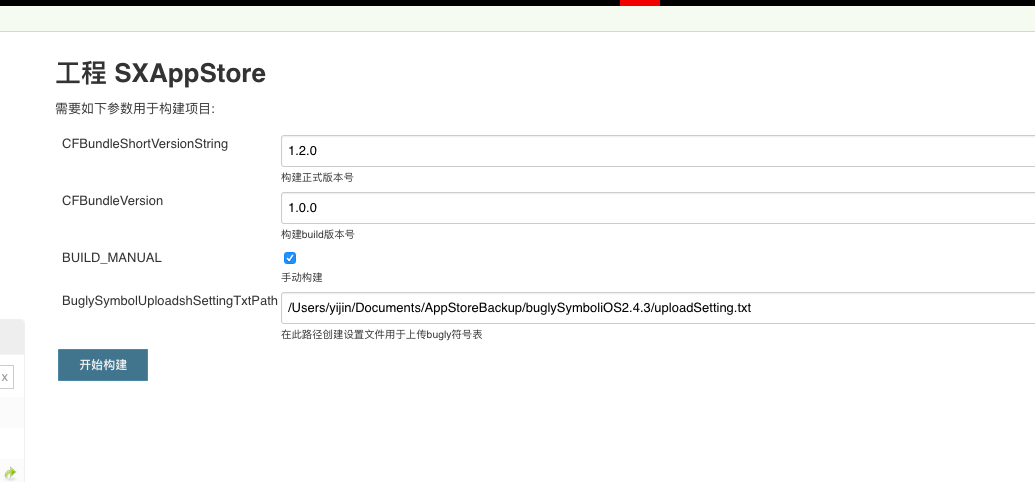
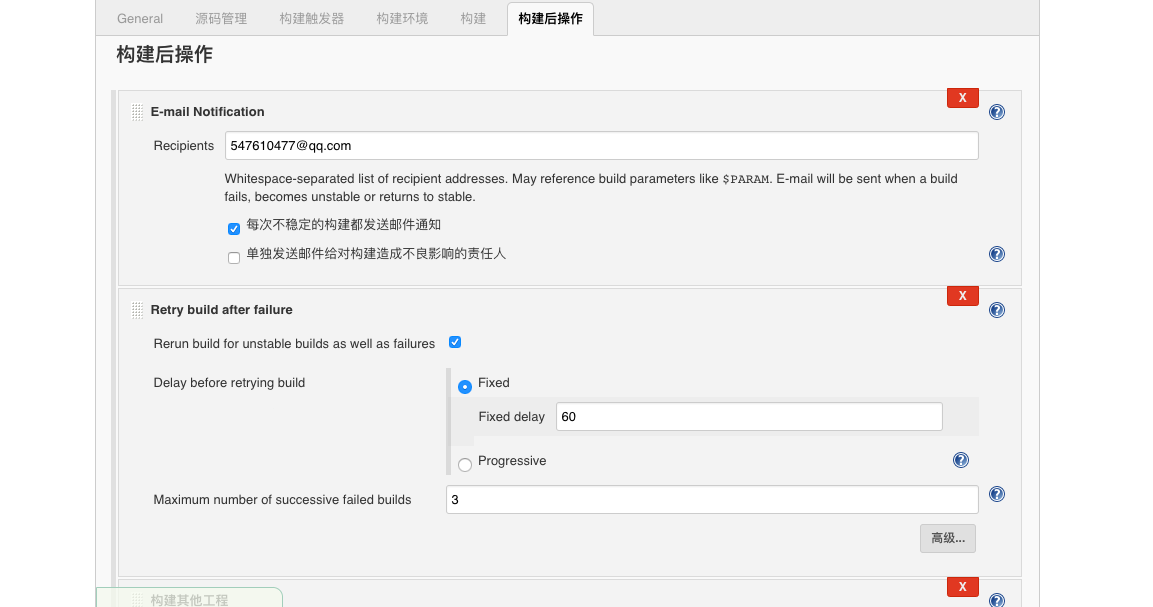
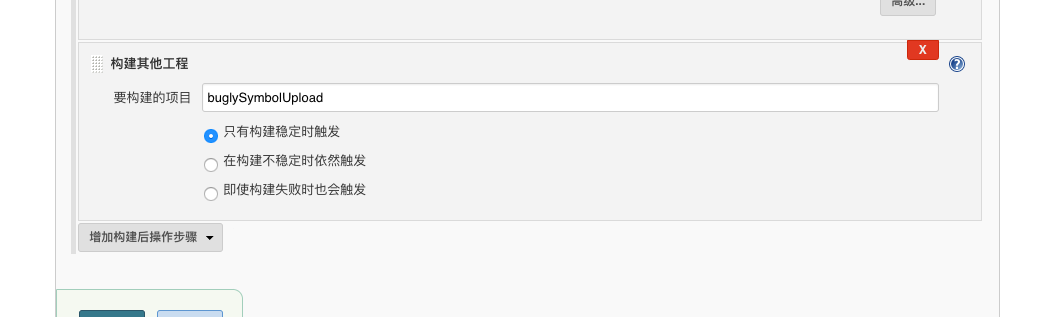
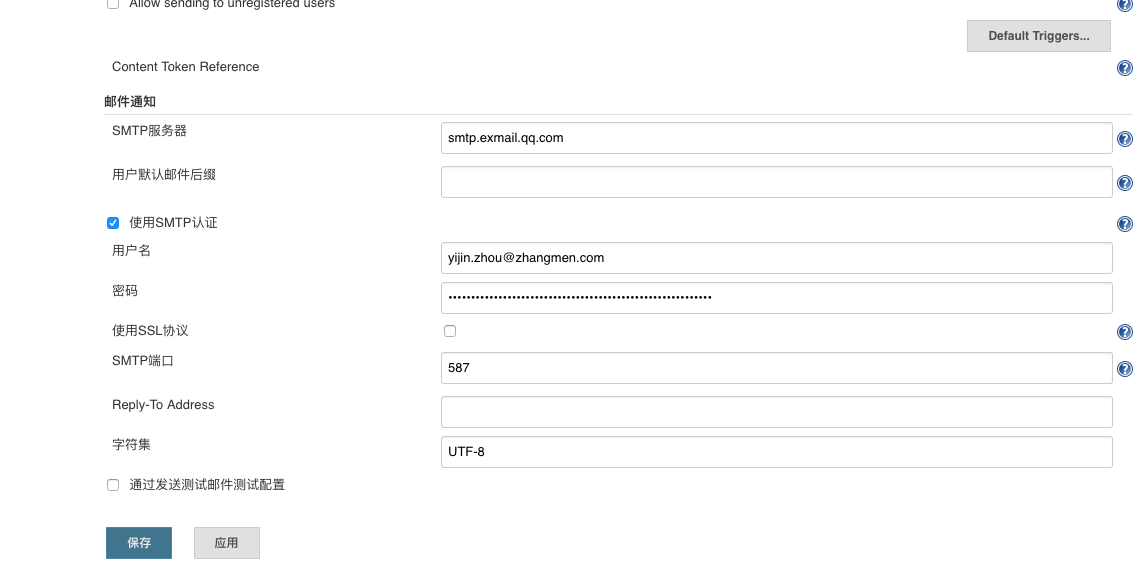
自动打包上传appstore











shell 1 :拉取更新依赖

#更新依赖的项目

#bin/bash - 1

export LANG=en\_US.UTF-8

export LANGUAGE=en.US.UTF-8

export LC\_ALL=en\_US.UTF-8

PROJECTBASEPATH=$WORKSPACE/SXMainProject

cd $PROJECTBASEPATH

RN\_ShengXue=../RN\_ShengXue

ZMTeacherCommunity=../ZMTeacherCommunity

ZMPKit=../ZMPKit

if [ -d ${RN\_ShengXue} ];then

if [ "`ls -A ${RN\_ShengXue}`" = "" ]; then

echo "${RN\_ShengXue} is empty"

git clone -b sx\_release git@zmgitlab1.zmlearn.com:xiaoyong.chen/RN\_ShengXue.git ${RN\_ShengXue}

else

echo "${RN\_ShengXue} is not empty"

cd ${RN\_ShengXue}

git pull origin sx\_release

fi

else

git clone -b sx\_release git@zmgitlab1.zmlearn.com:xiaoyong.chen/RN\_ShengXue.git ${RN\_ShengXue}

fi

cd $PROJECTBASEPATH

if [ -d ${ZMTeacherCommunity} ];then

if [ "`ls -A ${ZMTeacherCommunity}`" = "" ]; then

echo "${ZMTeacherCommunity} is empty"

git clone -b sx\_release git@zmgitlab1.zmlearn.com:ios/ZMTeacherCommunity.git ${ZMTeacherCommunity}

else

echo "${ZMTeacherCommunity} is not empty"

cd ${ZMTeacherCommunity}

git pull origin sx\_release

fi

else

git clone -b sx\_release git@zmgitlab1.zmlearn.com:ios/ZMTeacherCommunity.git ${ZMTeacherCommunity}

fi

cd $PROJECTBASEPATH

if [ -d ${ZMPKit} ];then

if [ "`ls -A ${ZMPKit}`" = "" ]; then

echo "${ZMPKit} is empty"

git clone -b sx\_release git@zmgitlab1.zmlearn.com:yaozong.lin/ZMPKit.git ${ZMPKit}

else

echo "${ZMPKit} is not empty"

cd ${ZMPKit}

git pull origin sx\_release

fi

else

git clone -b sx\_release git@zmgitlab1.zmlearn.com:yaozong.lin/ZMPKit.git ${ZMPKit}

fi

shell2:

#bin/bash - 1

export LANG=en\_US.UTF-8

export LANGUAGE=en.US.UTF-8

export LC\_ALL=en\_US.UTF-8

PROJECT\_DIR=$WORKSPACE/SXMainProject

cd ${PROJECT\_DIR}

echo "当前路径:"

pwd

PROJECT\_NAME=ShengXue

PROJECT\_SCHEME=ShengXue

BUILD\_CONFIGURATION=Release

EXPORTOPTION\_PATH=${PROJECT\_DIR}/AutoPackage/SXAPPSTOREExportOptions.plist

WORKSPACE\_FILE=${PROJECT\_NAME}.xcworkspace

INFOPLIST\_PATH=${PROJECT\_DIR}/${PROJECT\_NAME}/info.plist

NOTISERVICE\_INFOPLIST\_PATH=${PROJECT\_DIR}/NotificationService/info.plist

if [ ${BUILD\_MANUAL} == "true" ]; then

#设置App构建版本号

/usr/libexec/PlistBuddy -c "Set CFBundleShortVersionString ${CFBundleShortVersionString}" ${INFOPLIST\_PATH}

#设置推送服务构建版本号

/usr/libexec/PlistBuddy -c "Set CFBundleShortVersionString ${CFBundleShortVersionString}" ${NOTISERVICE\_INFOPLIST\_PATH}

#设置build构建版本号

/usr/libexec/PlistBuddy -c "Set CFBundleVersion ${CFBundleVersion}" ${INFOPLIST\_PATH}

fi

echo "移除并创建新的打包输出相关文件夹"

BUILDTIME=`date +%Y-%m-%d\_%H:%M:%S`

EXPORTBASEDIR=${PROJECT\_NAME}\_Release\_${BUILDTIME}\_v${CFBundleShortVersionString}

ARCHIVE\_DIR=~/Desktop/${EXPORTBASEDIR}/build\_archive

EXPORT\_DIR=~/Desktop/${EXPORTBASEDIR}/build\_ipa

DSYM\_DIR=~/Desktop/${EXPORTBASEDIR}/build\_dsym

rm -rf ~/Desktop/${EXPORTBASEDIR}

mkdir -p ${ARCHIVE\_DIR}

mkdir -p ${EXPORT\_DIR}

mkdir -p ${DSYM\_DIR}

LOGPATH=~/Desktop/${EXPORTBASEDIR}/log.txt

touch ${LOGPATH}

echo "打包之前先清理"

xcodebuild clean -configuration ${BUILD\_CONFIGURATION} -scheme ${PROJECT\_SCHEME} -quiet >> ${LOGPATH}

echo "清理完成"

echo "更新cocoapods库"

/usr/local/bin/pod install

/usr/local/bin/pod update ZMYAnalyse --no-repo-update

echo "开始编译打包成Archive"

BEGIIN\_TIME=`date +%s`

CONFIGURATION\_BUILD\_PATH=${CONFIGURATION\_BUILD\_DIR}/${BUILD\_CONFIGURATION}-iphoneos

ARCHIVE\_PATH=${ARCHIVE\_DIR}/${PROJECT\_SCHEME}.xcarchive

xcodebuild archive -workspace ${WORKSPACE\_FILE} -scheme ${PROJECT\_SCHEME} -configuration ${BUILD\_CONFIGURATION} -archivePath ${ARCHIVE\_PATH} -quiet >> ${LOGPATH}

echo "~~~~~~~~~~~~~~~~~~~~~~~~ 查看是否版本构建成功 ~~~~~~~~~~~~~~~~~~~~~~~~"

#xcrachive 是一个文件夹不是个文件 所以使用 -d 来判断

if [ -d $ARCHIVE\_PATH ]

then

echo "版本构建成功!"

APPDSYM=${ARCHIVE\_PATH}/dSYMs/${PROJECT\_NAME}.app.DSYM

if [ -d ${APPDSYM} ]; then

cp -rf ${APPDSYM} ${DSYM\_DIR}/

echo "DSYM文件导出成功!"

fi

else

echo "版本构建失败......"

exit 1

fi

END\_TIME=`date +%s`

ARCHIVE\_TIME="构建版本所耗时间$[ END\_TIME - BEGIIN\_TIME ]秒"

BEGIIN\_TIME=`date +%s`

echo "开始导出ipa包"

xcodebuild -exportArchive -archivePath ${ARCHIVE\_PATH} -exportOptionsPlist ${EXPORTOPTION\_PATH} -exportPath ${EXPORT\_DIR} -allowProvisioningUpdates -quiet >> ${LOGPATH}

EXPORT\_IPA=${EXPORT\_DIR}/${PROJECT\_SCHEME}.ipa

#如果是文件则说明成功

if [ -f ${EXPORT\_IPA} ]

then

echo "导出ipa成功!"

open ${EXPORT\_DIR}

else

echo "ipa导出失败......"

exit 1

fi

END\_TIME=`date +%s`

EXPORT\_IPA\_TIME="导出ipa所耗时间$[ END\_TIME - BEGIIN\_TIME ]秒"

echo "时间消耗"

echo "$ARCHIVE\_TIME"

echo "$EXPORT\_IPA\_TIME"

echo "开始上传ipa到AppStore"

#验证并上传到App Store

# 将-u 后面的XXX替换成自己的AppleID的账号，-p后面的XXX替换成自己的密码

altoolPath="/Applications/Xcode.app/Contents/Applications/Application Loader.app/Contents/Frameworks/ITunesSoftwareService.framework/Versions/A/Support/altool"

echo "开始ipa验证"

"$altoolPath" --validate-app -f ${EXPORT\_IPA} -u studyonline@china-start.cn -p Study2018

echo "开始上传"

"$altoolPath" --upload-app -f ${EXPORT\_IPA} -u studyonline@china-start.cn -p Study2018

echo "准备上传符号表"

echo "创建setting.txt"

if [ -f ${BuglySymbolUploadshSettingTxtPath} ];then

rm ${BuglySymbolUploadshSettingTxtPath}

fi

#获取bundleid

CFBundleIdentifier=$(/usr/libexec/PlistBuddy -c "Print CFBundleIdentifier" ${INFOPLIST\_PATH})

echo ${APPDSYM} > ${BuglySymbolUploadshSettingTxtPath}

echo ${CFBundleIdentifier} >> ${BuglySymbolUploadshSettingTxtPath}

echo ${CFBundleShortVersionString} >> ${BuglySymbolUploadshSettingTxtPath}

echo "你好牛逼啊！！！！！！"