# IOS使用jenkins进行持续集成

1. 安装jenkins

../share/tools/Jenkins/jenkins-1.605.pkg

1. 安装xcode插件

进入系统管理→管理插件→高级→上传插件

按照下面的顺序依次上传

../share/tools/Jenkins/token-macro.hpi

../share/tools/Jenkins/xcode-plugin.hpi

1. 新建job

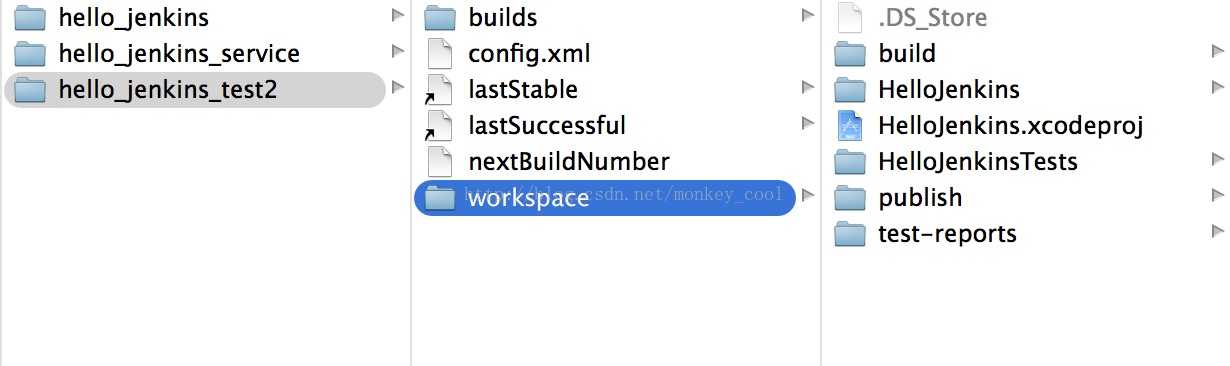
新建一个free的job.



1. 源码管理

如果使用svn，选择subversion，直接指向地址即可；如果没有使用svn，选择none，则需要先运行下job，这样jenkins的工作目录就创建了一个结果failed的job，目录为.jenkins/jobs/你的job名字/。这时候，job下的目录还有builds，workspace等目录文件，我们需要的就是往workspace中放入我们的代码工程（这是官方的解决方案）。

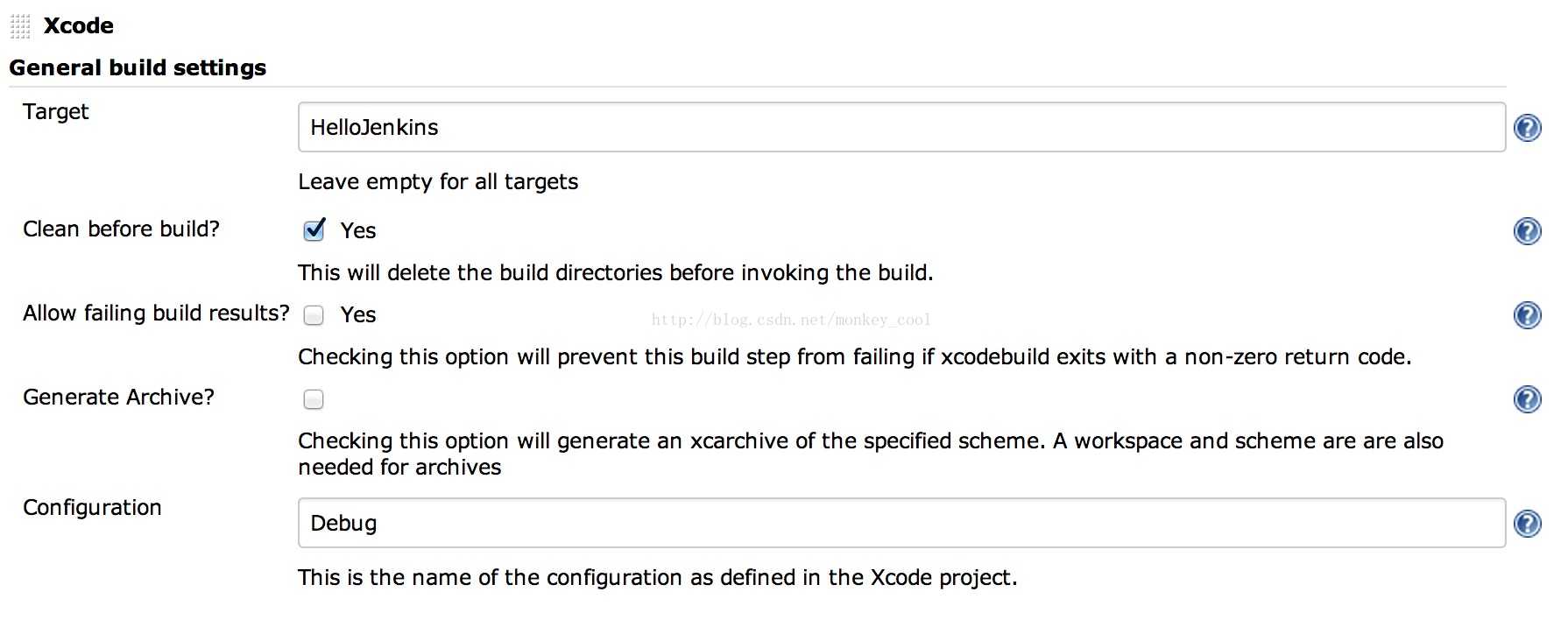
目录结构见下图。



这样workspace中就有了代码。

1. 构建xcode项目

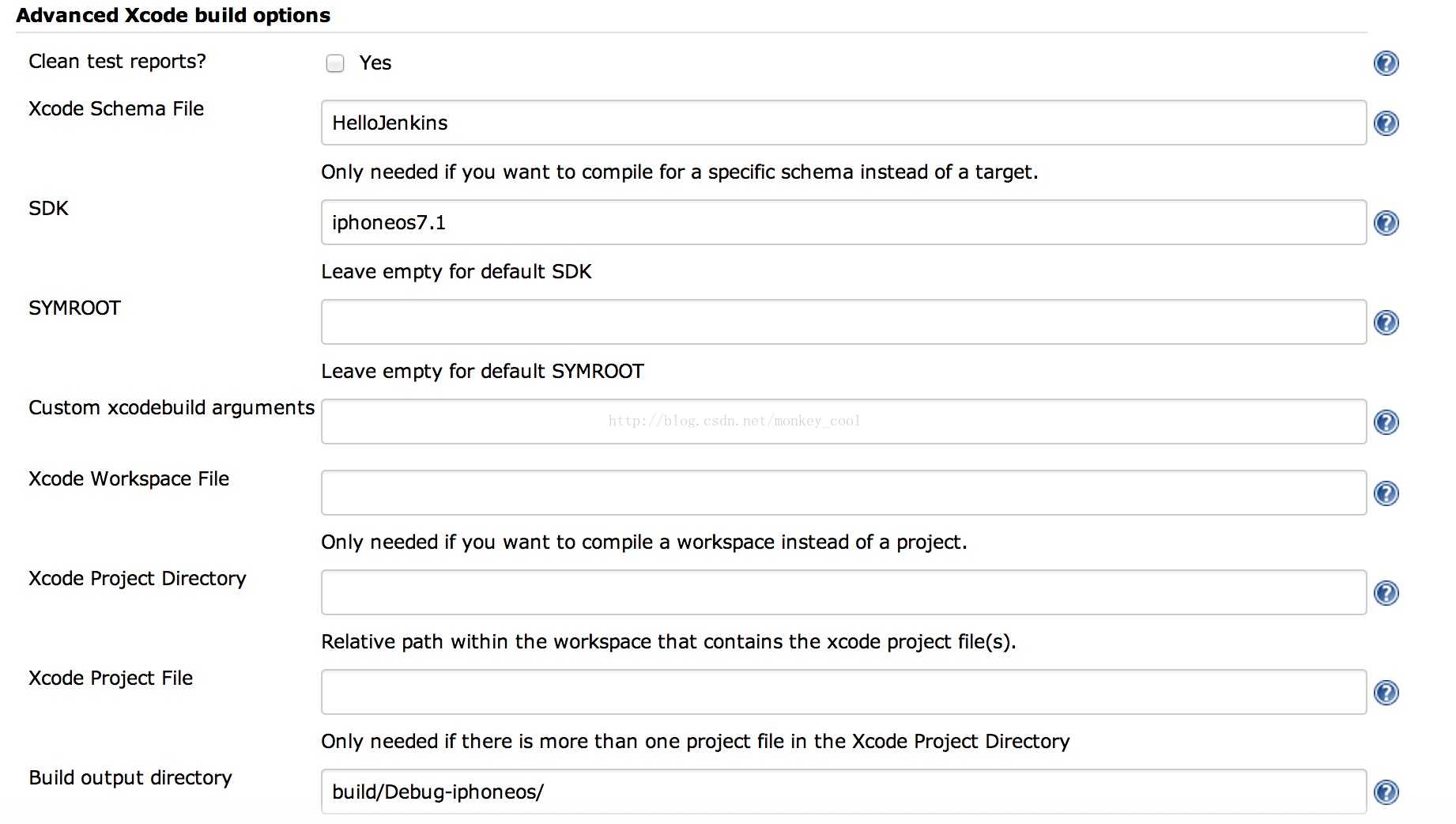
在增加构建步骤下拉中选择xcode。



如上图，简单配置参数，执行项目的target，Configuration为Debug(看情况选择Release)，下面还有个Pack application and build .ipa选项，这个我没选。

下面还有个Code signing和钥匙链的选项，可以不填，会自动找默认的（项目中配置的吧，可以在运行时的console中看到执行的日志，输出中会有每一步做什么了）。

在下面还有个xcode的高级参数配置，如图：



scheme配置，以及需要build运行的SDK，这里选择iphoneos（这是要准备在真机上运行），还有构建的输出目录。

在这里说下模拟器的事项，这里不能配置成iphonesimulator，是因为没用，模拟器不能运行起来的，首先苹果unix命令能够启动模拟器，但是没有提供api来命令行方式往模拟器中安装ipa程序并运行（xcodebuild命令没有这个支持），上网找了个解决方案，发现是很久之前的了，4年前的，有个github的提供的插件，但是这个插件工程需要依赖iPhoneSimulatorRemoteClient.framework这个库，而现在的mac都搜不到这个库，所以不能运行，so命令行方式启动模拟器，并安装运行ipa的方式不行了，只能真机方式来了。

运行job，此时应该能出现当前的目录文件，那个ipa还出不来呢。



现在到这，说明jenkins的这个IOS的job已经能够基本ok了。