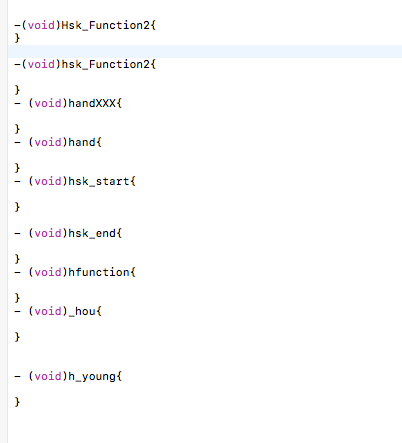
iOS方法名混淆

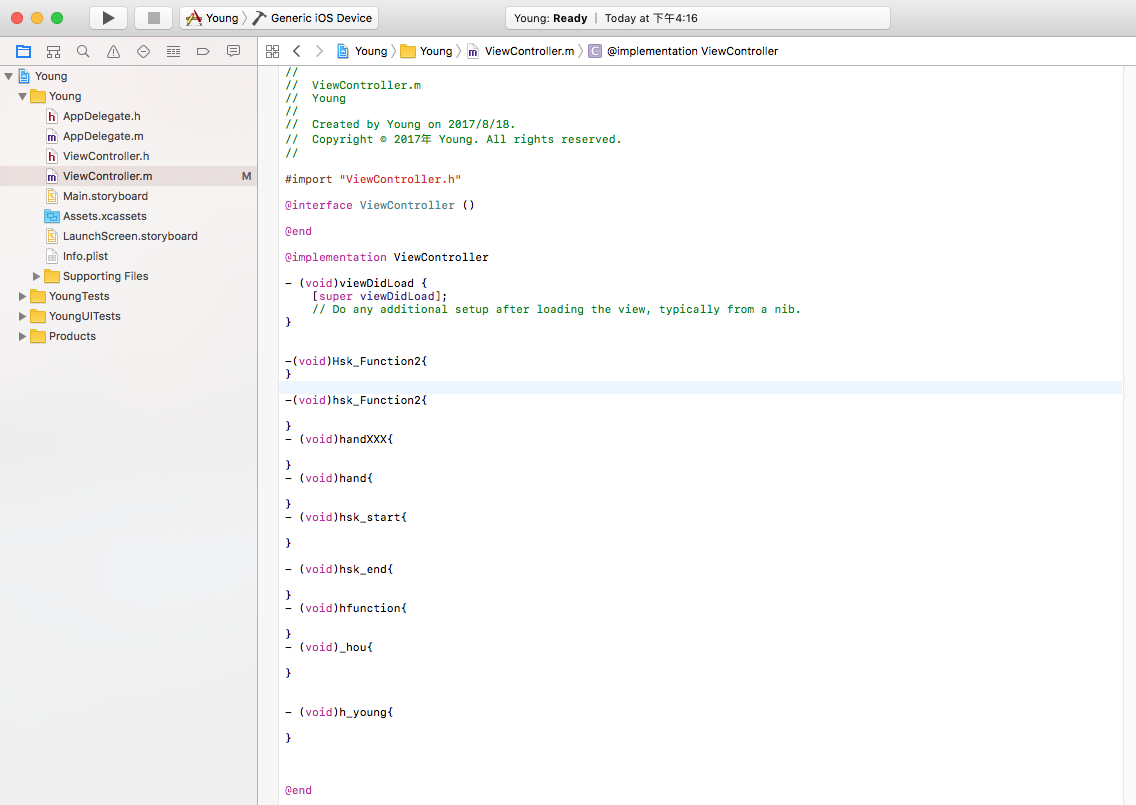
**运行项目时不要有中文路径**

找一个完整项目或新建一个项目Young

添加一些方法：

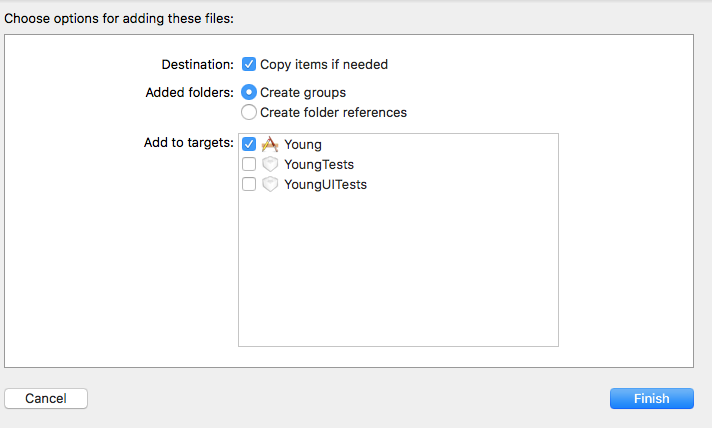


此时已为一个完整项目

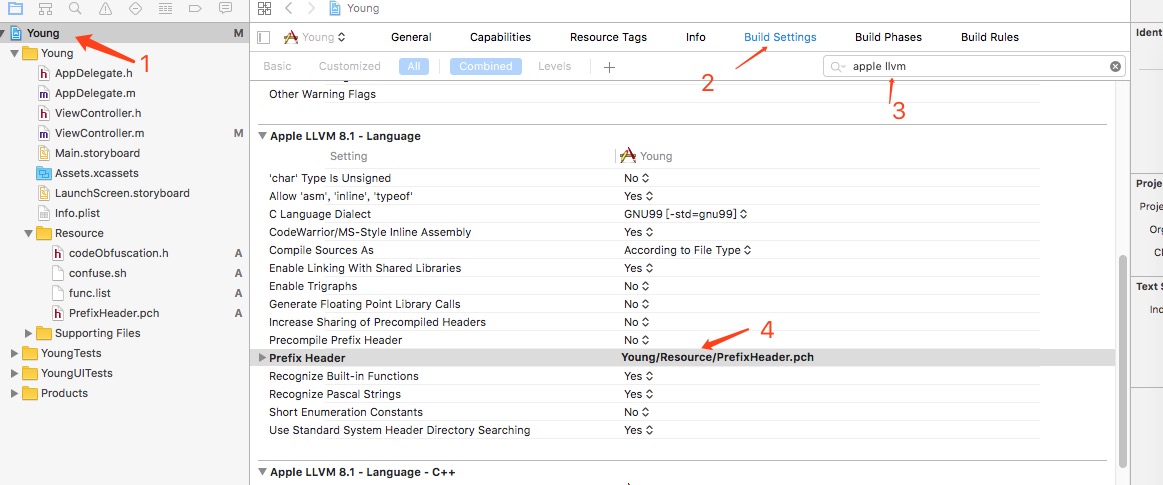


接下来进行添加项目混淆代码

1. 将混淆文件添加至项目中(可直接拖进项目



1. 查看并修改文件配置
2. 配置.pch文件



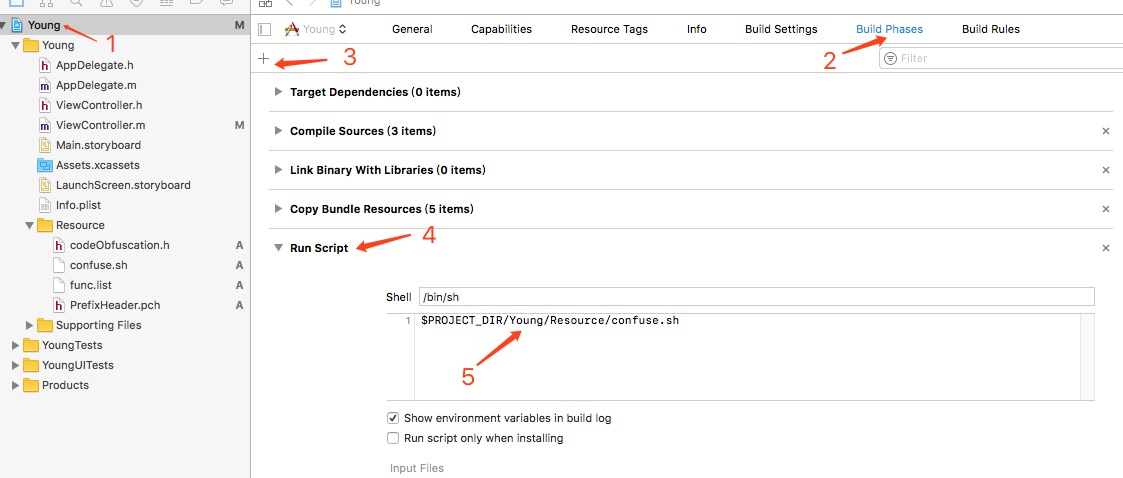
按此顺序进行配置

项目名称/Resource/PrefixHeader.pch

因为我的项目是Young

所以我的为：Young/Resource/PrefixHeader.pch

1. 配置Run Script



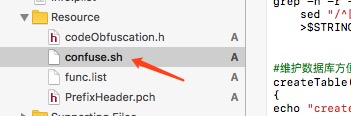
当点击到第3步时选择创建Run Script

然后输入comfuse.sh的配置路径

可完全复制，只需要将我的项目名称Young改为你的项目名称

$PROJECT\_DIR/Young/Resource/confuse.sh

回到混淆文件，选中confuse.sh.进行修改



点开confuse.sh后如图

￼

将1、2、3处的项目名称Young改为自己的项目名称

箭头4处表示为混淆以hsk\_开头的方法名 ，可以修改为自己需要的方法名

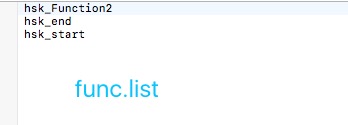
另外3个文件不用做修改



所有配置均已完成，开始运行项目

第一遍运行过后(conmand + B 或 conmand + R),

func.list文件里生成以hsk\_开头的所有的方法名



codeObfyscation.h文件生成了对应方法名的随机字符串



第一遍运行过后，生成的APP里还未产生代码混淆

再运行一次，第二次运行过后，通过class-dump反编译，查看到方法名已发生混淆替换



更为有趣的是app包里的混淆后的随机数，是上次编译的对应随机值，也就是说：每编译一次，app会把上次的<方法名：随机值>存入APP, 这次会产生新的随机值，已供下次编译使用，并且没有保存。

当我们对APP进行解压或查看包内容，如图：



发现有混淆代码的源文件在里面，但没有方法名的对应关系，所以这种方法还是很安全的。

混淆后的代码用Hopper打开的结果，如图

