## Curl

Curl 加ip 可以在命令窗口请求某个ip, 在加上 -X POST可以选择使用什么方式请求.

## url和uri的区别

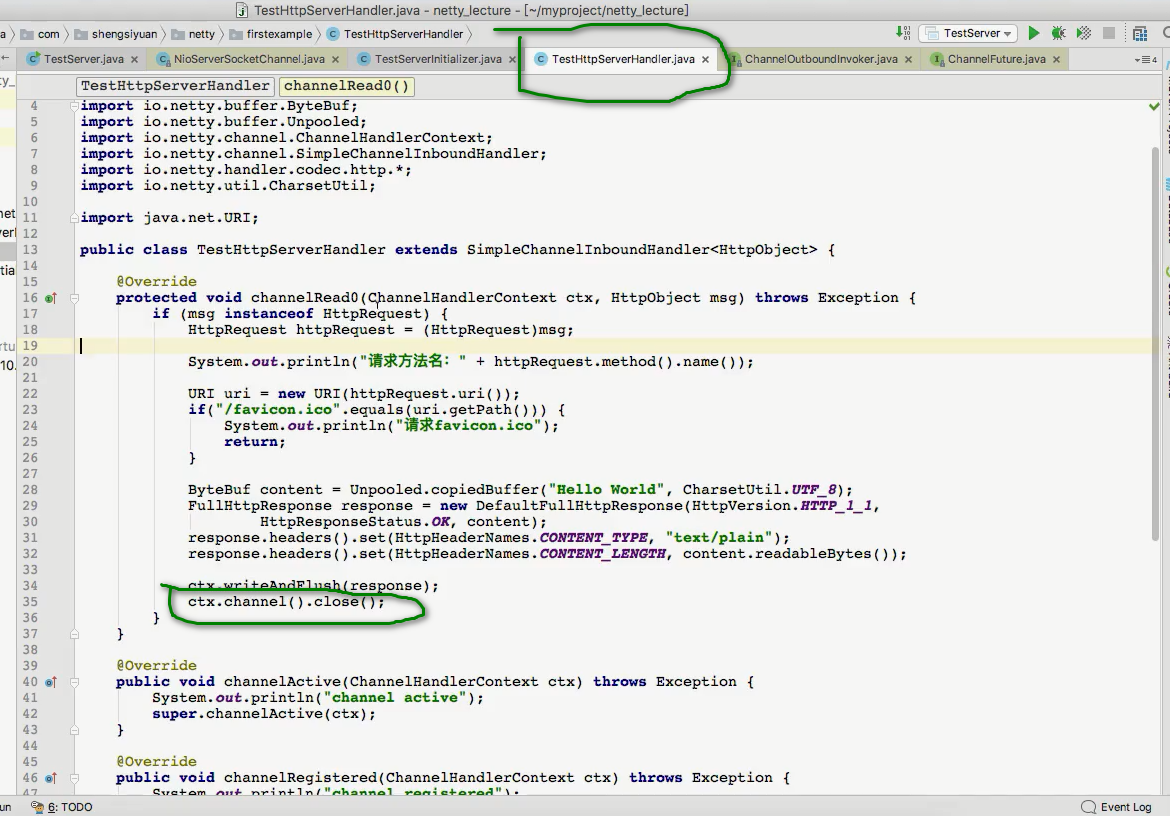
在Java的URI中，一个URI实例可以代表绝对的，也可以是相对的，只要它符合URI的语法规则。而URL类则不仅符合语义，还包含了定位该资源的信息，因此它不能是相对的，schema必须被指定。

ok，现在回答文章开头提出的问题，到底是imgUrl好呢，还是imgUri好？显然，如果说imgUri是肯定没问题的，因为即使它实际上是url，那它也是uri的一种。那么用imgUrl有没有问题呢？此时则要看它的可能取值，如果是绝对路径，能够定位的，那么用imgUrl是没问题的，而如果是相对路径，那还是不要用ImgUrl的好。总之，用imgUri是肯定没问题的，而用imgUrl则要视实际情况而定。

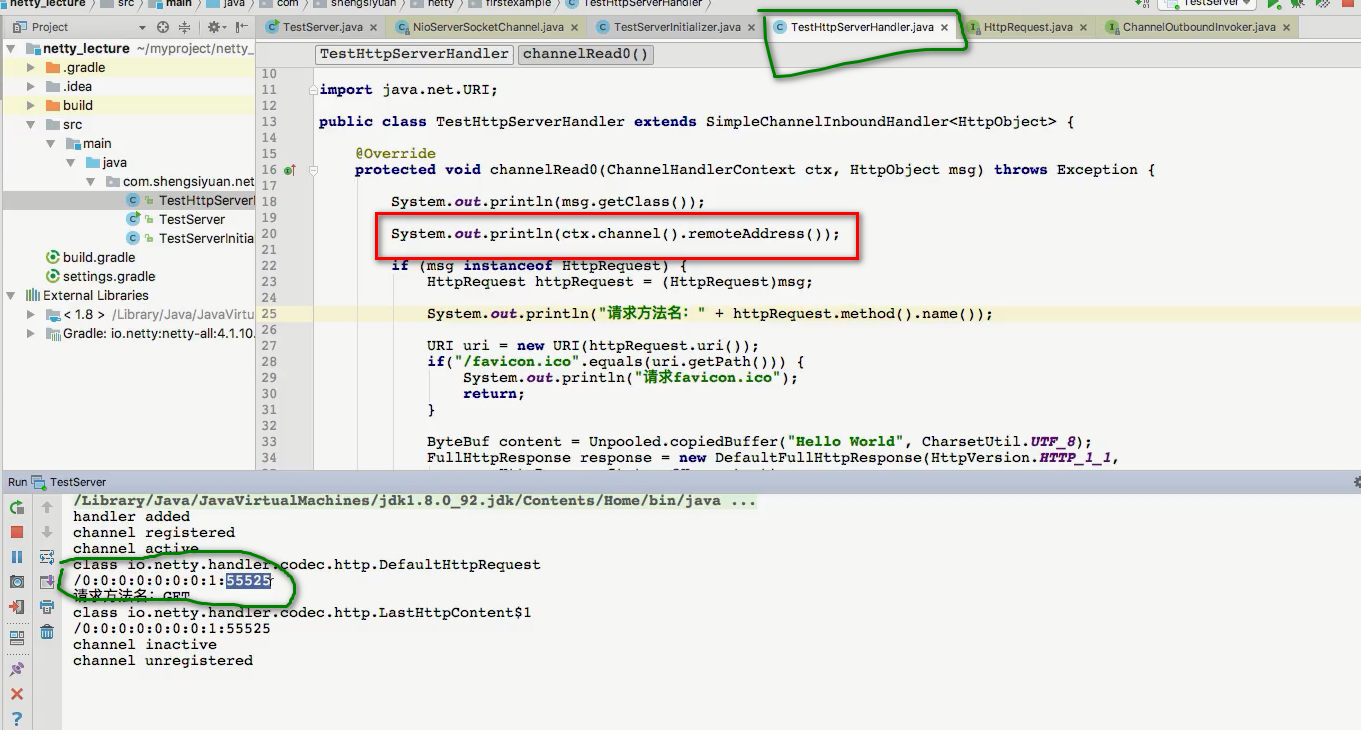
第二个，从HttpServletRequest的javadoc中可以看出，getRequestURI返回一个String，“the part of this request’s URL from the protocol name up to the query string in the first line of the HTTP request”，比如“POST /some/path.html?a=b HTTP/1.1”，则返回的值为”/some/path.html”。现在可以明白为什么是getRequestURI而不是getRequestURL了，因为此处返回的是相对的路径。而getRequestURL返回一个StringBuffer，“The returned URL contains a protocol, server name, port number, and server path, but it does not include query string parameters.”，完整的请求资源路径，不包括querystring。

总结一下：URL是一种具体的URI，它不仅唯一标识资源，而且还提供了定位该资源的信息。URI是一种语义上的抽象概念，可以是绝对的，也可以是相对的，而URL则必须提供足够的信息来定位，所以，是绝对的，而通常说的relative URL，则是针对另一个absolute URL，本质上还是绝对的。

浏览器在访问netty的时候创建的channel需要等好长时间取消,可以手动来取消.



Netty可以获取请求的ip和端口



Lsof -i:端口号 : lsof -i :8091 知道8091端口被哪个进程占用



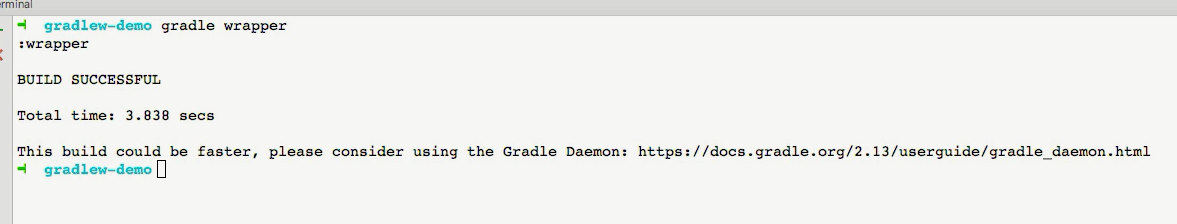
Grpc

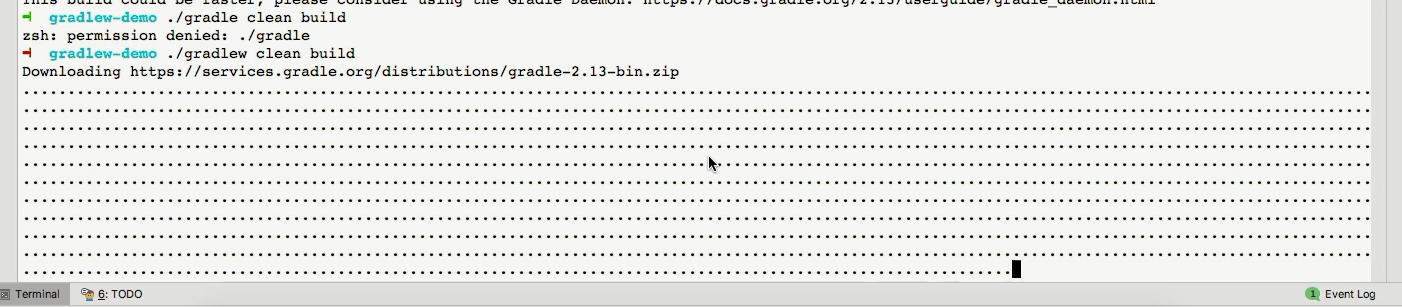
进入下载的文件目录下的example

打开dos

执行 ./gradlew installDist win环境直接点gradlew.bat执行

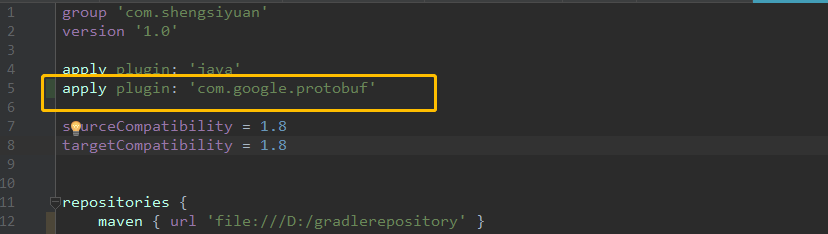
使用gradlewrapper可以将项目打包,并且声称可执行的文件

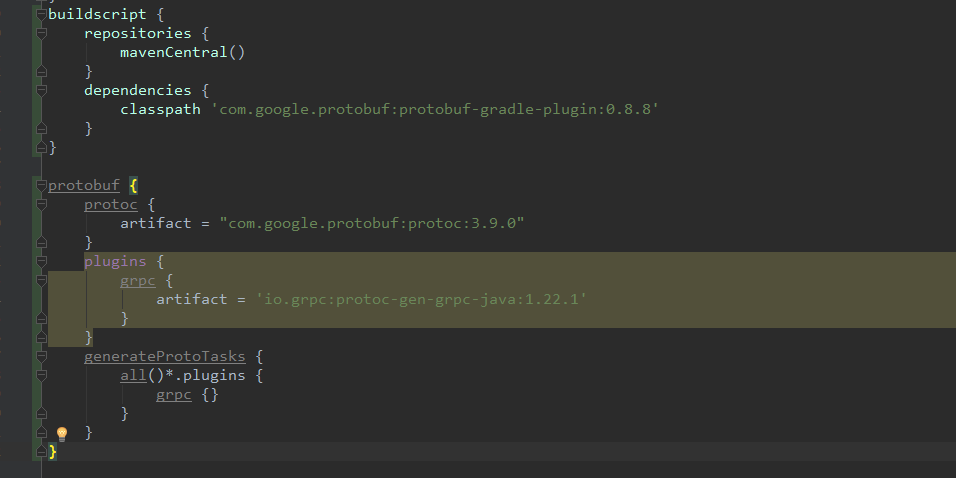




Grpc基于protobuf

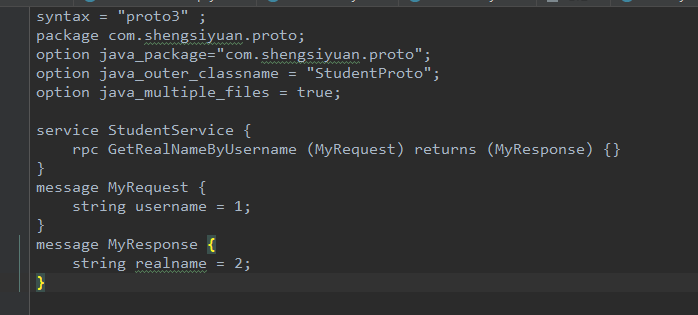
在gradle的配置文件中添加配置信息





在main目录下创建proto文件夹创建Studet.proto文件

输入

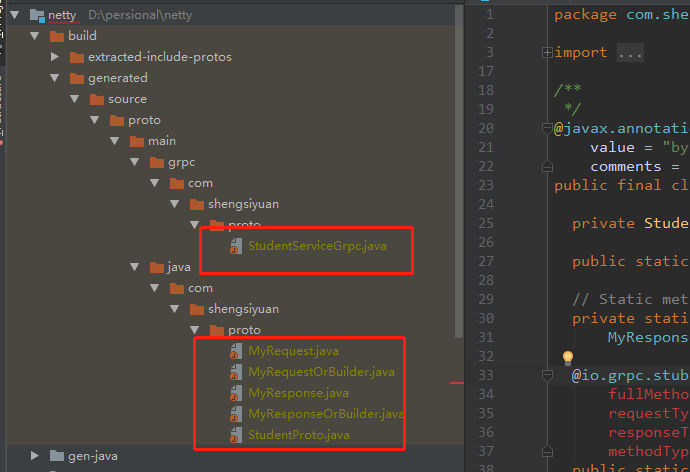


执行命令

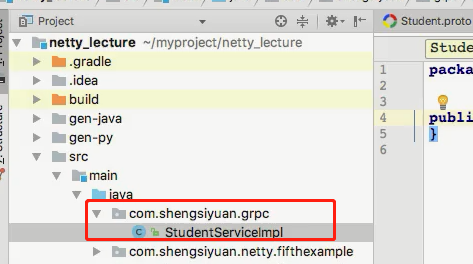
gradle generateProto 声称rpc文件

在builde文件中的generate文件下找到对应的文件

将文件拷贝到项目的proto文件下(创建)



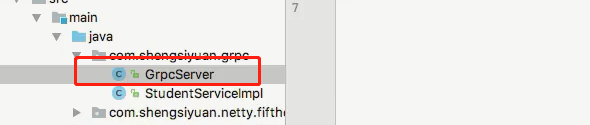
视频25



创建对应的文件



类中的代码如下

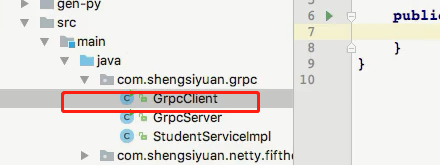


创建服务端代码

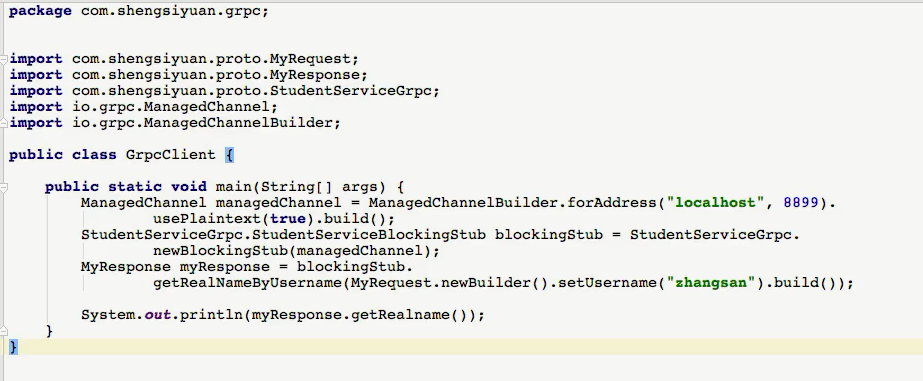


服务器端代码

定义了服务启动方法,服务停止方法,其中有个等待终止方法,该方法必须要在main中调用,否则服务启动后就会立即停止.



创建客户端文件



客户端代码



服务端的关闭钩子,防止服务停止后,socket不能关闭.关闭钩子是一个初始化了但是还没有开始执行的线程.

视频26