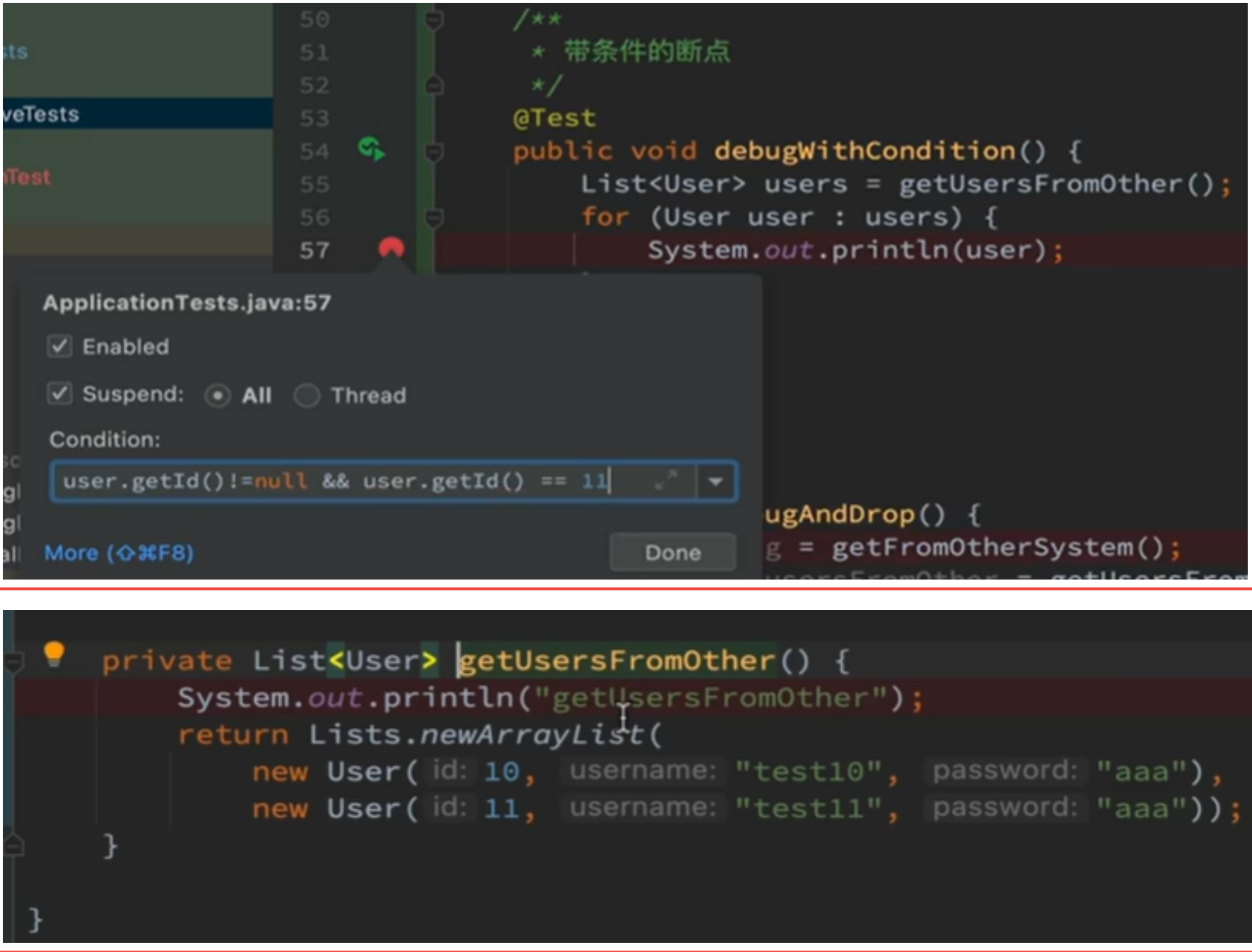


IDEA&Debug四技巧

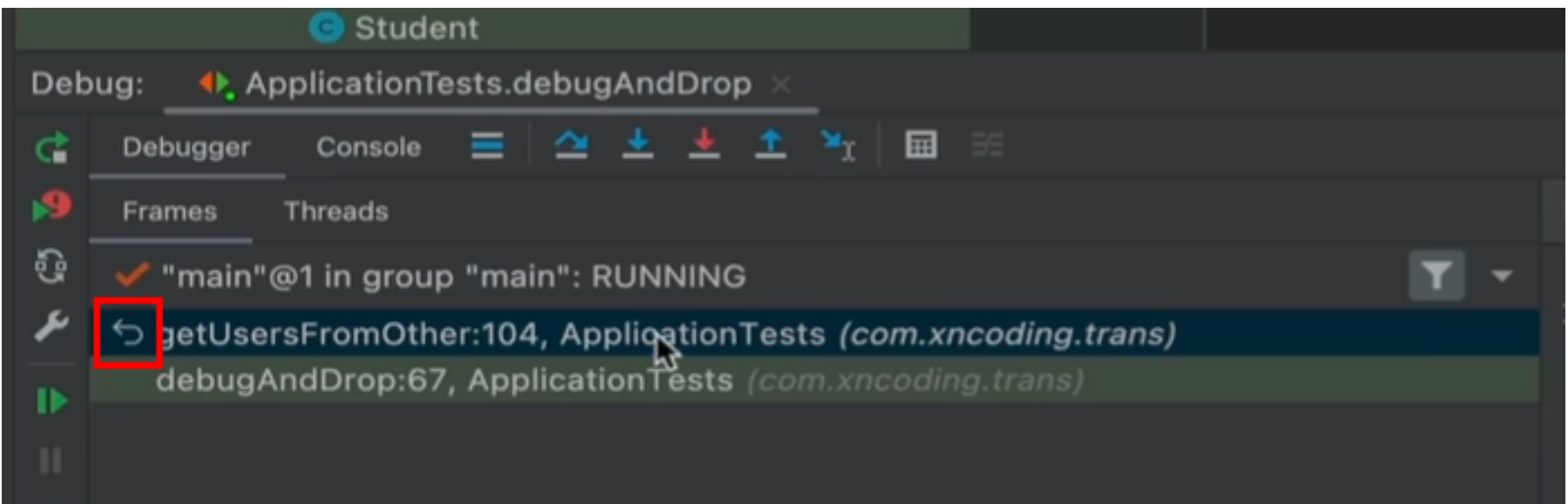
带条件的断点

首先打断点，然后右键鼠标右键，然后有个condition，它会读取到当前行的变量，比如eg中有一个user的变量，就可以这样做，"user.getId()!=null && user.getId() == 11"，首先user的值不等于空然后user的值等于11时，我们就让它停住。而在eg中的getUsersFromOther中会有两条测试数据返回，10和11。

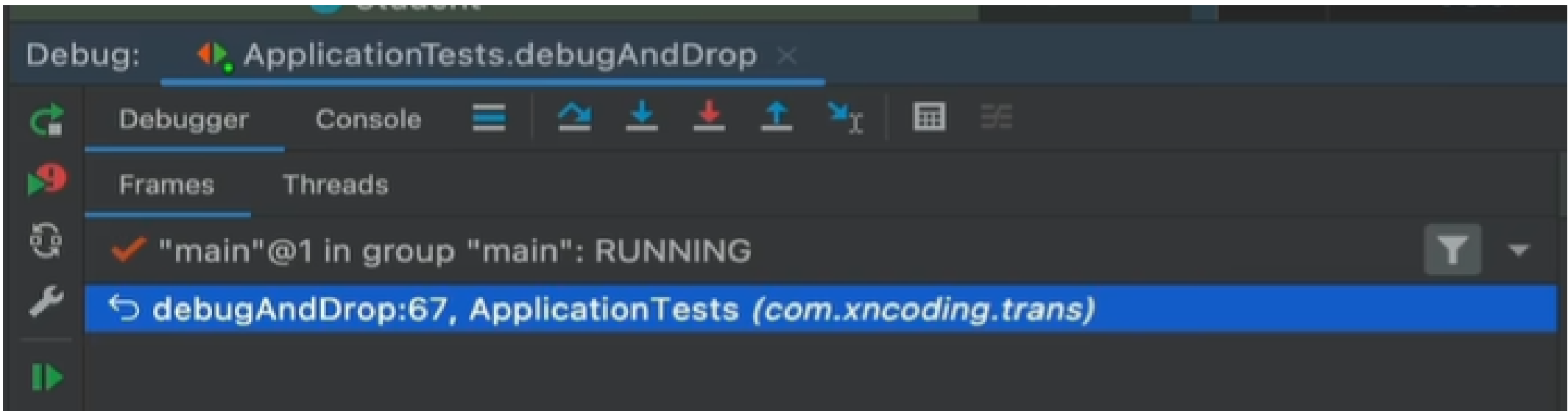


断点回退

即断点执行到了一个具体的行号，行号上面的一些内容没看清楚，想让它退回去重新执行一遍再来做一些调试。再来做一些细节的查看，此时就可用到断点的回退。

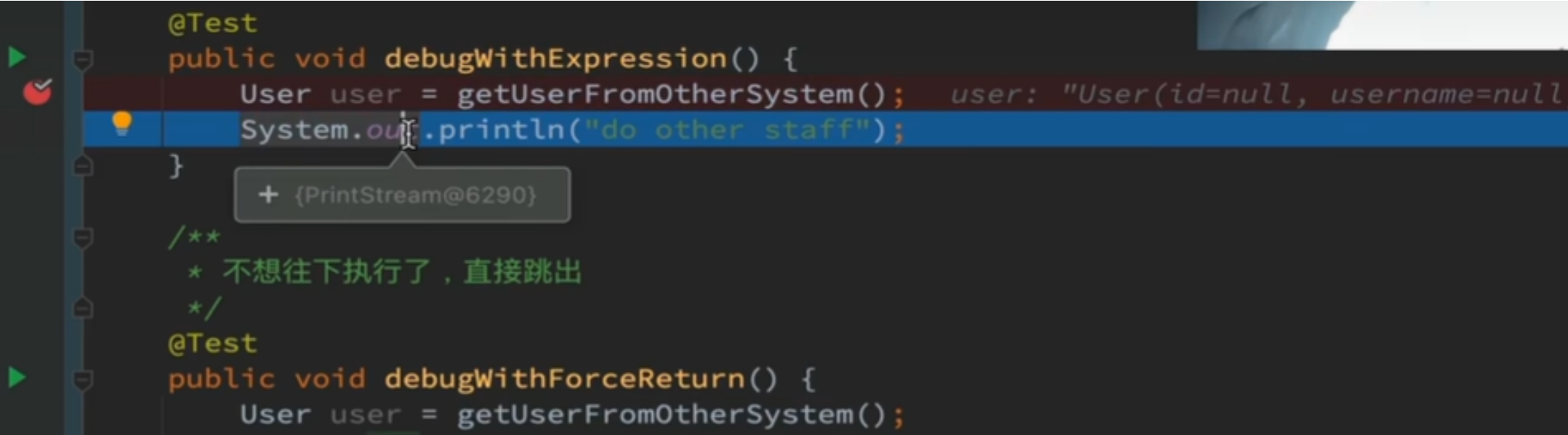


新的IDEA在我们执行的这一行有一个Debugger面板，下面有一个Frames，是执行的“栈帧”，在每一帧的前面，会有一个撤回的这样一个“reset frame”的快捷操作，老一点的IDEA可能是在上面符号的区域，就按照自己的方式找到“reset frame”或者“drop frame”这样的操作就可以了，此时选择“reset frame”



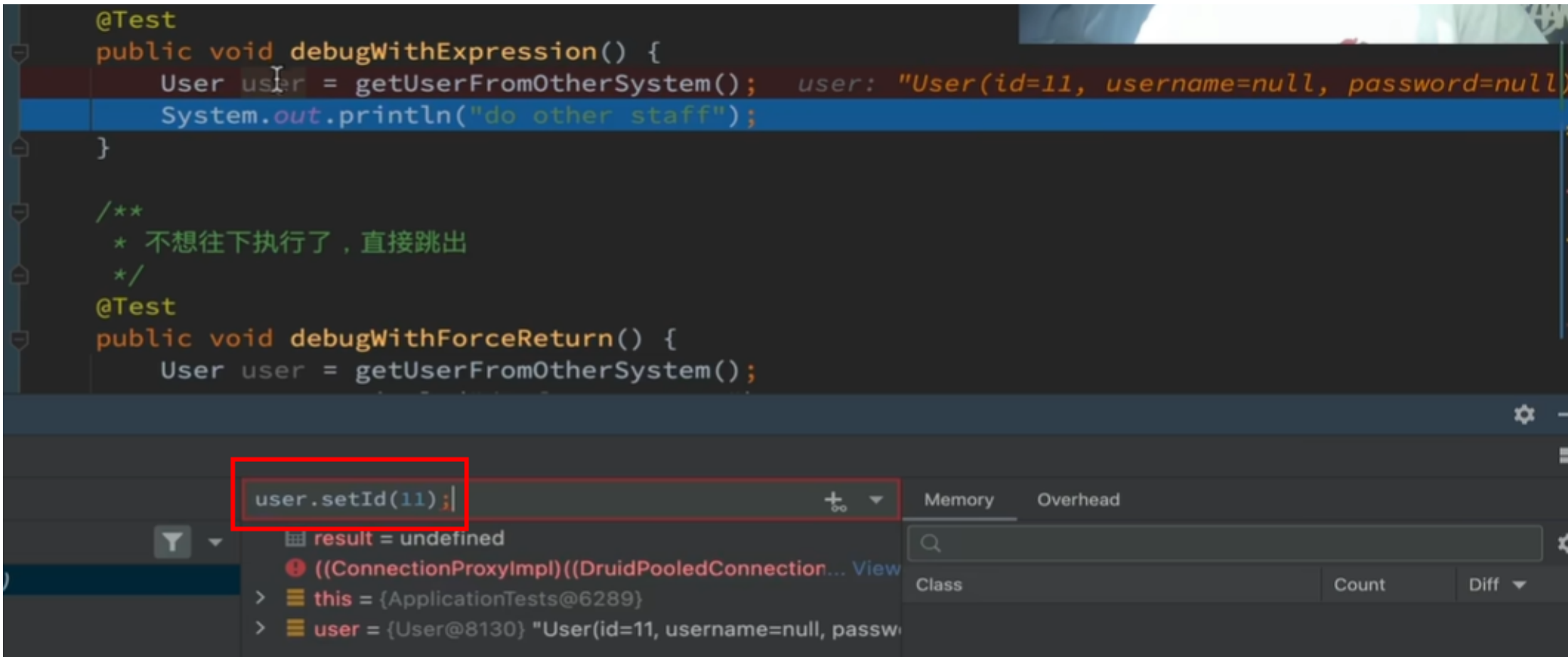
可以看到它退出了方法，调到了进入方法之前的时候。

执行表达式

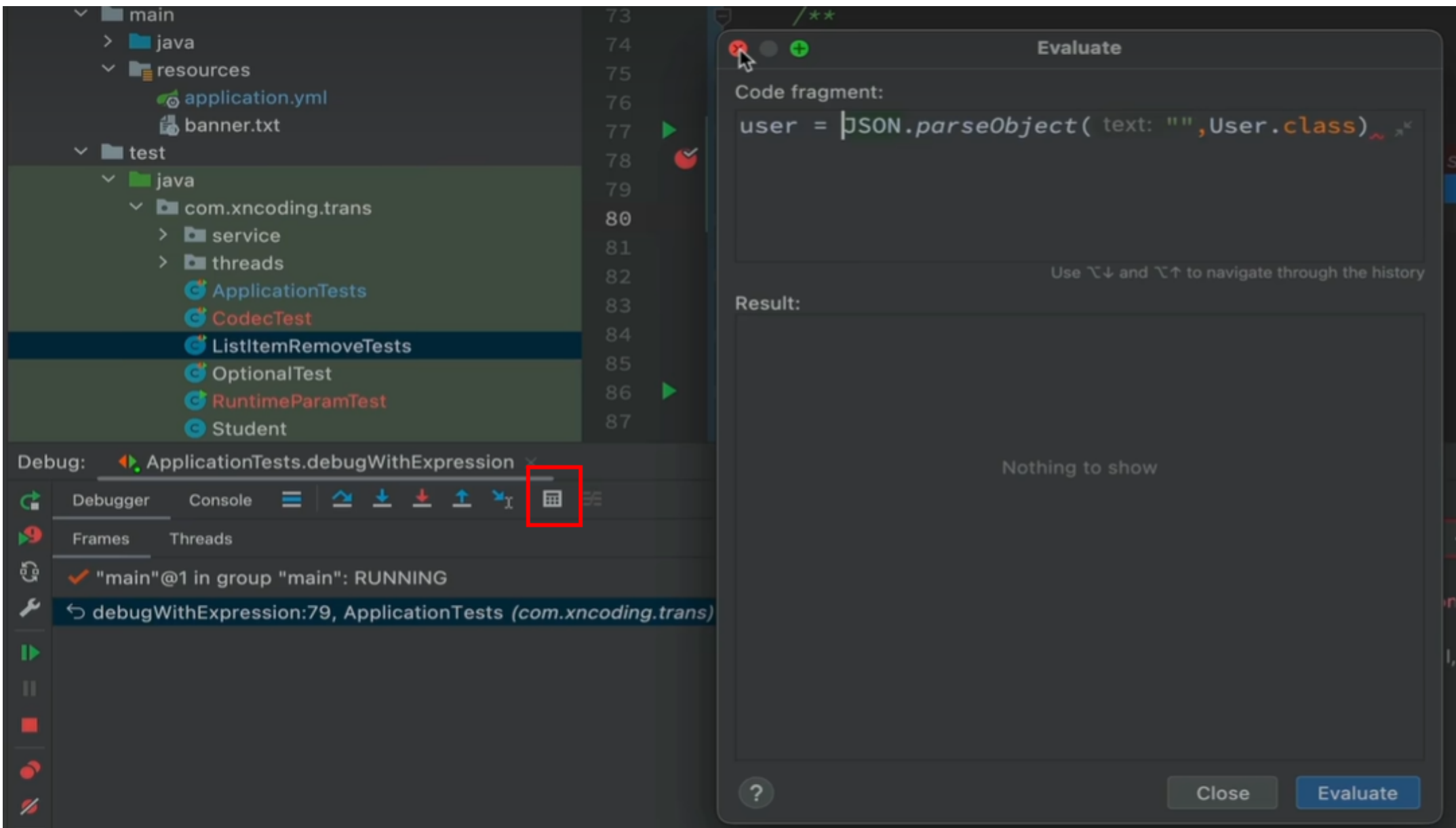


如图所示啊，这里通过“getUserFromOtherSystem”拿到了一个“user”的值，然后假如拿到的这个user的里面比如ID、比如一些属性不是我们想要的，即目前user的值不能满足下面的一些逻辑时，比如“user某属性等于某值，才会执行xxx”，这时呢就可以去执行一些表达式。

当学会了这个执行表达式以后，后期再断点调试的话，会有非常多的使用场景，比如去修改它的值，或实现某些条件的运行等，

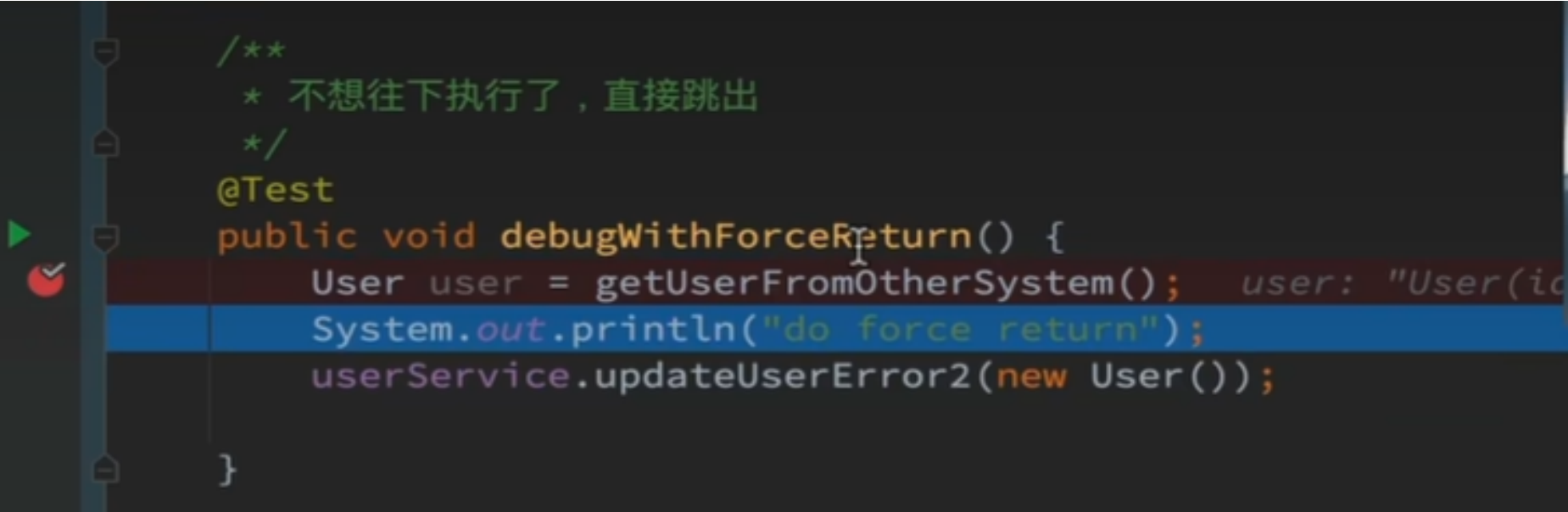


可以直接在这里执行，新版的IDEA会在这里有一个快速的expressions的输入框，比如“user.setId(11)”，然后按“回车”，就会发现user这个变量就变成了Id等于11

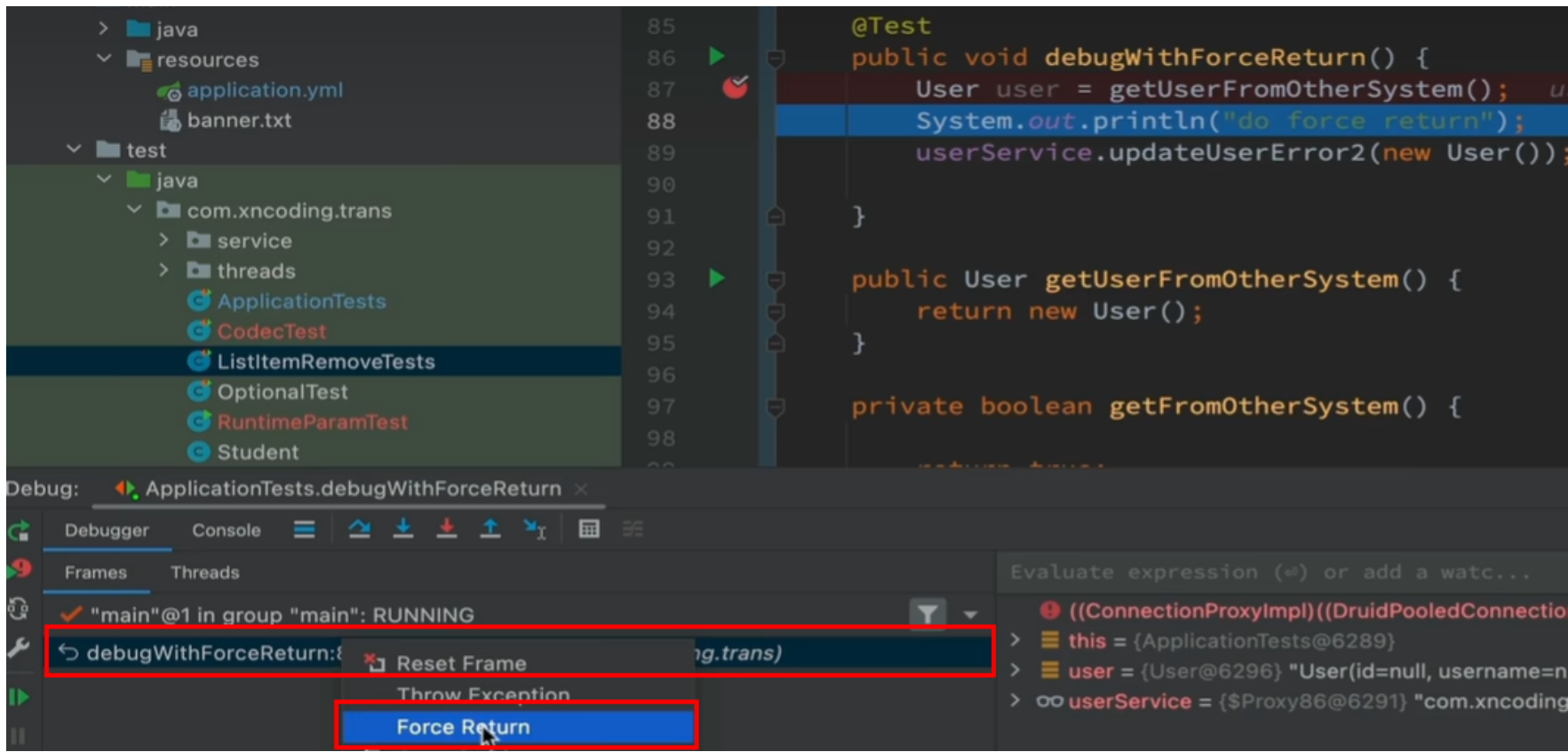


还有另外一种方式，就是可以去执行这种长段的，这里有一个“Evaluate expression”，在Mac是“Control+F8”，点完之后这里就可以执行大段的“user expression”。

不想往下执行了，直接退出



断点打在了红色那行，现在运行到了蓝色那行，首先是查出来的那个user，然后执行了一堆逻辑，然后我们发现这个逻辑有问题，不能再继续往下执行了，因为继续往下执行，这个地方就把数据库的一个值给更改了，如果说测试的数据很难造，然后还要去删数据，或者说要去改一些数据库的值，那就很麻烦，所以说当前已经找到了问题所在，就不想让它继续往下运行了。



很简单，把光标定位在执行的这一行，然后右键，这里有一个“force return”，当我们点击“force return”以后，去Console控制台，那句update语句就不会执行了，看Console没有进行任何数据库操作，即那个Mybatis的Mapper的一些打印没有打印出来，就证明我们的程序已经跳出了，后续是没有再进行操作的。

链接：[https://www.bilibili.com/video/BV1Rm4y1P7j8/?spm\\_id\\_from=333.337.search-card.all.click&vd\\_source=e30efa494d35a4ad38c40fad72f758d](https://www.bilibili.com/video/BV1Rm4y1P7j8/?spm_id_from=333.337.search-card.all.click&vd_source=e30efa494d35a4ad38c40fad72f758d)