

你写的 return null 正确吗? - 文章 - 伯乐在线



原文出处: [richiezhong](http://richiezhong.com)

上次《[你写的try...catch真的有必要吗](#)》引起了很多朋友的讨论。本次我在code review又发现了一个问题,那就是有人有意无意的写出了return null这样的代码,例如:

C#

```
public User GetUser(Guid userId)
{
    if ( /*for some reason*/ )
        return null;
    return DB.GetByUserId(userId);
}
```

这样的写法有木有问题?

在我看来没有充分的理由不应该返回null,因为方法的使用者并不知道在何种条件下会得到null,从而导致臭名昭著的NullReferenceException异常。

如果由于其他原因导致无法得到一个User,我们应该抛出一个明确的异常供上层服务决定如何处理:

C#

```
public User GetUser(string userName, string password)
{
    if ( /*for some reason*/ )
        return new SpecificException("can't get this user because
....");
    return DB.GetByUserId(userId);
}
```

在我读过的开源项目中我几乎没有见到过return null的写法。能让我一时想到的两个linq方法FirstOrDefault()和LastOrDefault(),这两个方法通过命名的方式提醒使用者该方法会返回null。

说到FirstOrDefault()方法让我想起了很多人容易犯的另一个错误:

C#

```
public User GetUserById(Guid userId)
{
    return list.FirstOrDefault(x=>x.UserId==userId);
}
```

在确认数据库中该userId必须对应一个User的情况下使用FirstOrDefault()方法,此种场景我会建议你

果断使用Single()方法。因为此时使用FirstOrDefault()会隐藏bug。你期望该userId必须得到一个User，如果Single()方法抛出异常则说明有bug出现，并且让你在第一时间发现该bug。

F#为了减少null类型的使用引入了option类型，在将option用作函数的返回类型时，如果没有对未定义的类型做处理，编译器会报错。

C#

```
let invalidInt = None
match invalidInt with
| Some x -> printfn "the valid value is %A" x
| None -> printfn "the value is None"
```

如果此处的模式匹配忘记编写None->分支，编译器将会报错，从而提醒你必须处理invalidInt值为None时的逻辑。但是在C#中使用null类型，编译器给予不了我们帮助，所以我们应该避免return null这样的代码，你觉得呢？

刚才搜了一下stackoverflow，发现一篇很有意思的讨论 [Should a retrieval method return 'null' or throw an exception when it can't produce the return value?](#) 我觉得里面的回答比较准确。

拿高薪，还能扩大业界知名度！优秀的开发工程师看过来 -> 《[高薪招募讲师](#)》

1 赞 收藏 [评论](#)



合作联系

Email: bd@jobbole.com

QQ: 2302462408 (加好友请注明来意)

更多频道

- [小组](#) - 好的话题、有启发的回复、值得信赖的圈子
- [头条](#) - 分享和发现有价值的内容与观点
- [相亲](#) - 为IT单身男女服务的征婚传播平台
- [资源](#) - 优秀的工具资源导航
- [翻译](#) - 翻译传播优秀的外文文章
- [文章](#) - 国内外的精选文章
- [设计](#) - UI, 网页, 交互和用户体验
- [iOS](#) - 专注iOS技术分享
- [安卓](#) - 专注Android技术分享
- [前端](#) - JavaScript, HTML5, CSS
- [Java](#) - 专注Java技术分享
- [Python](#) - 专注Python技术分享