

如何向开源社区提问题 #545

lifesinger opened this issue on Feb 17, 2013 · 32 comments

lifesinger commented on Feb 17, 2013

Member

如何向开源社区提问题

使用软件产品，或多或少都会遇到问题。对于商业产品，我们可以咨询客服寻求帮助。对于公司自己研发的产品，我们可以直接请教专家同事。但对于开源软件，在遇到问题时，如何才能及时有效地寻求帮助呢？

本文以开源类库 [SeaJS](#) 为例，说说我心目中的最佳实践。

提问前

遇到问题时，心里都很着急。在决定向开源社区提交问题前，最好先做做以下功课：

尝试从官方文档中找到答案

确保自己阅读过至少一次官方文档。这样在遇到问题时，如果能回忆起只言片语，就可以再去读一遍相关文档，问题往往也就解决了。

Google 是你的朋友

对于成熟的开源项目，你遇到的问题，很可能别人也遇到过。这时通过 Google、StackOverflow 等网站的搜索服务，可以帮你快速定位并解决问题。永远记住，地球上的你并不孤单，包括你遇到的问题。

挖掘 Bug 宝藏

开源软件一般都会有自己的 Bug 管理方案，比如 WebKit、V8、jQuery、SeaJS 等等。从它们的官网上找到 Bug 管理地址，然后通过搜索看看有无你遇到的问题。对于活跃社区来说，这一招经常很管用。比如 jQuery 的 [Bug Tracker](#)，通过右上角的 Search Tickets 可以找到非常多有用的信息。一个运作良好的 Bug 库，经常是一座巨大的宝藏。SeaJS 是直接通过 [GitHub Issues](#) 来管理，你可以在 Issues 中找到很多信息。

求助身边的朋友

如果你使用的开源软件，在朋友圈或同事圈里也有人使用，那么抬起你的脚、或拿起你的电话，真挚诚恳的探讨不会遭遇拒绝，而会增进友谊。不要犹豫，你的内心渴望面对面交流，你的朋友也是。

如果以上 4 步都无法解决你遇到的问题，也别犹豫，立马向开源社区提交问题就好。

提问时

提问有很多种，比如你认识作者，直接面对面请教就行。下面探讨的是如何通过互联网的方式来问问题。

平和对等的心态

很多开源软件都是免费的，作者往往是业余时间出于兴趣在维护，没有义务回答社区问题。提问时，不要把自己摆在顾客的位置，比如

项目马上要上线了，请务必帮忙解决
这是我的邮箱，请及时联系我

另外，也不要把自己摆在乞食者的位置，比如

冰天雪地跪求解答
救命啊，我的网站挂了

在开源社区，一切皆是朋友。无论对方是 Linux 内核的作者，还是某个 jQuery 插件的作者，你和作者都是对等的。你的提问是在帮助开源软件完善。平和对等的心态，可以让你的问题赢得更多人的阅读和思考。

通过正确的途径提交

如果遇到问题的开源软件有专门的 Bug 管理系统，请最好到这些指定系统中提交。比如，对于前端开发工程师来说，下面这些 Tracker 系统很重要。

- [jQuery Tickets](#)
- [WebKit Bugzilla](#)

- [Mozilla Bugzilla](#)

还有各个开源类库的 Issues 库，比如 SeaJS 的是：[seajs/issues](#)

最不好的途径是

- QQ、阿里旺旺、微信等群组。这些群组主要是用来工作或休闲的。对开源项目来说，在这些地方提问，作者一般不会关注，效率非常低。
- 微博、Facebook 等社交网络。不少人在微博上通过 at 或私信询问 SeaJS 问题，这些我经常看不到。看到了，也不情愿回复。微博是扯淡、交流情感的地方，一般是写代码写累了，才去逛逛，很少会有在社交网络上回答技术问题的心情。

通过正确的途径提交问题，一般可以让你的问题得到及时准确的回复。

使用明确、有意义的标题

抱着平和对等的心态，找到合适的途径后，就得静下心来将遇到的问题写成文字。书写文字不是一件简单的事情，我们可以从遵循一些简单的规则开始。

首先是标题要简洁清晰，要言之有物。比如

我遇到了一个 Ajax 问题
SeaJS 在我的浏览器上运行不了

上面的标题很糟糕，光看标题作者无法知道发生了什么事。当开源社区的问题很多时，上面这类标题，经常会让作者直接忽视或将优先级降到很低。更妥当的标题是

Ajax 请求未返回正确的 responseXML
SeaJS 2.0 在 IE6 上运行时抛错

明确、有意义的标题，可以帮助作者确定问题具体是什么类型、预估需要多少时间解决、是否现在马上解决等。一个好的标题，也有利于社区知识的沉淀和后期搜索。标题有如一个人的颜面衣着，虽然不是关键，但在嘈杂的信息社区中，这很重要。

遵循良好的模板

如果社区提供了问题模板，一定要仔细看下。比如 Google Code 社区，当你创建一个问题时，会自动提供以下模板：

What steps will reproduce the problem?
该问题的重现步骤是什么？

- 1.
- 2.
- 3.



What is the expected output? What do you see instead?
你期待的结果是什么？实际看到的又是什么？

What version of the product are you using? On what operating system?
你正在使用产品的哪个版本？在什么操作系统上？

Please provide any additional information below.
如果有的话，请在下面提供更多信息。

遵循这个模板去描述问题，经常能省很多事。作者一般也非常欢迎通过模板提交的问题。如果社区没有提供模板，也可以自己遵循以上模板来提交。

下面针对问题内容，具体说说一些需要注意的点。

语法正确、格式清晰

虽然我们不是作家，但正确的语法、清晰的格式，可以让读者赏心悦目，也就更有心情帮你一起思考解决问题。

对于很多需要代码来描述的问题，要尤其注意格式，比如

```
seajs.use('jquery',function($)  
{$(document).ready(function() { /* ... */ })});
```



可读性不如

```
seajs.use('jquery', function($) {  
  $(document).ready(function() {  
    // ...  
  });  
});
```



GitHub 的 Markdown 语法可以很好地支持代码排版、语法高亮等，建议书写代码时，一定要先阅读下说明：[GitHub Flavored Markdown](#)。这能让你的内容看起来很专业，社区也就更愿意会去帮助你，否则糟糕的排版，经常带来的是发帖之后的石沉大海。

描述事实、而不是猜测

事实是指，依次进行了哪些操作、产生了怎样的结果。比如

我在 Windows XP 下用 IE6 打开 [seajs.org](#) 后，点击“5 分钟上手 SeaJS”，这时浏览器弹出脚本错误提示，例子显示不正确。

上面是一段比较好的事实描述（更好的是把错误提示也截图上来），而不要像下面这样猜测：

SeaJS 在 IE6 下运行不正常，我怀疑是源码第 213 行有问题。

上面的描述，会让作者一头雾水、甚至很恼火。尽量避免猜测性描述，除非你能先描述事实，在事实描述清楚之后，再给出合理的猜测是欢迎的。

对于前端项目来说，如果能提供可重现错误的在线可访问代码，那是最好不过的。一旦你这么用心去做了，作者往往也会很用心地立马帮你解决。

描述目标、而不是过程

经常会有这种情况，提问者在脑袋里有个更高层次的目标，他们在自以为能达到目标的特定道路上卡住了，然后跑来问该怎么走。比如

SeaJS 的 `parseMap` 方法在遇到 `map` 的多个配置项同时匹配同一个路径时，应该允许用户指定是全部生效还是仅第一个匹配的配置项生效。

上面这个问题的背后，提问者实际上想解决的是如何通过 SeaJS 来做版本管理。提问者选择了通过 `map` 的方式来实现，但这过程中遇到了问题，因此跑过来继续怎么走。然而，如果只是描述过程，往往会把作者也绕进去。

实际情况却是，提问者选择的路本身就是一条崎岖之路，对于要解决的问题，实际上有更好的方式。这种情况下，描述清楚目标，讲清楚要干什么非常重要。

在描述自己是怎么做之前，一定要先描述要做什么。提问题时，What 往往比 How 更重要。

要有具体场景

无论在开源社区，还是微博、知乎等平台上，有一种非常常见的问题：

如何维护 JavaScript 代码？
如何使用 SeaJS 进行模块化开发？

这类问题还有很多，每每遇到，只能笑笑，然后悄悄地忽略掉。因此这类问题很难回答，就如下面这些问题一样：

如何才能让生命有意义？
如何打败淘宝？

这类提问者，一般比较浮躁，经常对问题本身也没有经过思考。踏实的提问者，不会让问题浮在空中无法回答，而会在具体场景中让问题落地：

我的项目有 20 多个 JS 文件，接下来还会急剧增加。目前遇到以下问题.....（省略五百字）..... 请问如何维护？

仔细检查、确保准确

是人都会犯错误，特别是在如此快节奏的互联网环境下。好不容易把问题描述清楚时，不要急着立刻提交。在提交前，至少保证从头到尾再仔细阅读一遍，比如语法错误、错别字、标点符号、排版等等。做到这些，不光是尊重别人，也是尊重自己。

提问后

提交问题后，建议通过邮件等方式订阅回复。互联网上最有效的沟通方式是异步沟通，不要期待作者马上回复，也不要心烦意乱着急地等待。出去看看天，数数云朵，你会逐步明白什么是风轻云淡。

尽可能补充信息

在接收到回复时，仔细阅读。最经常的情况是，社区回复的，经常不是你想要的。比如

根据你的描述，问题无法重现。能否提供具体使用环境和重现步骤？

这时要淡定。仔细看看自己提交的问题描述是否足够清晰，如果有可补充的信息，尽量补充，以帮助作者能尽快定位问题。比如

很抱歉，我前面有一步描述不正确，实际情况是我是在 IETester 中运行的.....

谦和淡定的交流，不光能帮助你解决问题，还有助于你结交更多朋友。

适当的总结

当问题终于解决时，建议对问题进行总结。可以编辑原帖，也可以通过博客等方式总结。你的总结，会让遇到同样问题的朋友们受益，并且对自己的技能也是一种提高。前端业界，无论国内还是国外，有很多牛人之所以成为牛人，很大程度上都是因为总结思考的好习惯。

不要忘记感谢

最后，记得感谢。很多开源软件的作者，都是利用业余时间
在创作代码。你的感谢，汇集许许多多大家的感谢，会让开源社区充满爱与力量。

延伸阅读

- [如何有效地报告 Bug](#)
- [提问的智慧](#)
- [回答的智慧](#)

最后的最后，如果你认可这篇文章，欢迎以各种形式转载。你的传播，能让整个开源社区更美好。



lifesinger closed this as completed on Feb 17, 2013

yuanjianhang commented on Feb 17, 2013

对头！这些东西还是总结下的好！

raisezhang commented on Feb 17, 2013

谢谢 玉伯 真是受益匪浅！

mutoulbj commented on Feb 17, 2013

受益匪浅

hooopo commented on Feb 17, 2013

Good Job.

fivesmallq commented on Feb 17, 2013

之前向开源项目提过一些问题,都获得了反馈,感觉很不错.尤其是国外的项目,督促自己学好英语...