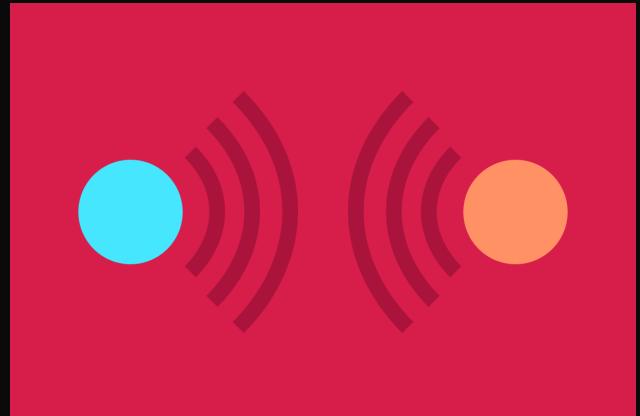


A Developer Friendly Framework for Asking for Help

01-12-22 | [Sheridan McKisick](#)

As a developer, asking for help is the best way to overcome tough challenges. Learn when to ask for help, how to ask for help, and how to make the most out of the help you've received.



Needing to ask for help is inevitable. I regularly face problems that are unfamiliar to me. Even the most experienced developers can still find themselves in situations where a bit of collaborative support goes a long way. In this article, I'll detail a few solutions for knowing when to ask for help, how to ask for help, and how to make the most out of the help you've received.

Knowing When You Need Help

Use Your Resources

It can be difficult to know when it's the right time to reach out for help, but feeling confident that you have exhausted all of the resources at your disposal can make that process easier. Keeping a mental (or written) list of different avenues of support, such as project [wikis](#), [documentation](#), Slack channels, and [pull requests](#), can make searching for answers more systematic and make you less likely to forget about a potentially valuable source of information. It's easier to put your best foot forward once you're armed with more information.

Try Timeboxing

However, if you begin to feel directionless, overwhelmed, or are struggling to gain ground, try giving yourself a time constraint, or [timeboxing](#). Timeboxing is creating a defined period of time to work on a problem and neither exceeding nor falling short of that allotted time. Don't think of it as a time crunch or a race against the clock; instead, think of it as a great technique to help you avoid spinning your wheels on a problem for too long. Asking for help immediately can stifle problem-solving skills and self-reliance. Conversely, avoiding asking for help can create blockers for your team moving forward. Timeboxing nurtures independence without costing your team velocity.

Reaching Out For Help

Prepare Yourself

If you've consulted your resources, tried timeboxing, and still have an unsolved problem, it's time to ask for help. The first step is doing a bit of preparation before contacting anyone because you should be able to articulate the problem that you are experiencing. It also helps to have examples or supporting evidence to explain the problem, like screenshots, copies of log messages, or the ability to replicate the problem.

You should also be able to articulate how you have attempted to solve the problem. Being able to recount the actions you've taken (and the effects of those actions) and acknowledge the resources you've used so far makes it easier for whoever is helping you. Do keep in mind that being prepared may not prevent your colleagues from asking you to repeat steps you've already taken. However, don't be offended—this is sometimes part of the process of receiving help.

Understand Scope

It can also be helpful to have a clear scope for the question you're asking. For example, is it just a clarifying question or is it a question that will require a lengthier response or a pairing session? Understanding the scope of the question you're asking can also help you to determine where you ask your question. You can pose broader questions towards developers in general, but you should ask your project teammates any questions that are highly specific to projects.

It's typically best to take advantage of channels or group conversations rather than DMs. Favoring group messages avoids constantly putting the responsibility of answering on any single developer and provides an opportunity for input from multiple developers. Asking for help in more public settings can be intimidating, but remember, if you've done your research and still have questions, there's a good chance that others may have the same questions as well.

Be Organized

When you do reach out for help, come to the table with your notes and resources already organized. Taking the time to organize and prepare in advance signals to your teammates that you value their time. You have already armed yourself with answers that someone dispensing help would likely ask, and that cuts back on the amount of time required from whoever is helping you. It also signals to your teammates that you have attempted to solve the problem independently. Although you have hit a snag, you are still trying your best to be resourceful.

Funnily enough, sometimes when you are preparing to ask for help, you may find yourself solving your problems. Recounting the steps you've taken may cause you to realize that you've overlooked something or have not checked a particular resource. Additionally, the process of switching gears from problem-solving an issue to documenting an issue can generate new insight. Seeing a problem from a slightly different angle can help to lead you to a solution.

Making the Most Out of Help

After receiving (and applying) help, there is still one final step to take: recording it. Jotting down the locations of resources that helped you to solve the problem can make for a great future reference. You can also make use of an engineering daybook to record how you solved the problem. David Thomas and Andrew Hunt introduce the idea of an engineering daybook, or a journal for recording ideas and things learned, in the [Pragmatic Programmer](#). Physically writing something down can help commit what you learned to memory, and organizing what you've written by date can make it a viable reference in the future.

Additionally, your commits or pull requests can be another opportunity to make notes about your problem-solving process. If the solution was not obvious to you, it may not be obvious to your teammates either. Taking the time to craft valuable commits or to update documentation translates to being kind to future—you and any [future teammates](#). You can also pay it forward and provide a learning opportunity for your entire team by sharing a quick summary of the issue and its documentation in a group channel.

Don't Be Afraid to Ask For Help

Asking for help might feel intimidating, but it doesn't have to be. Learning how to strike a balance between independent problem-solving and collaboration is a valuable skill. With a bit of forethought (and some afterthought), asking for help can transform into something highly beneficial to both you and your teammates. Ultimately, asking for help is a necessary part of collaboration, teamwork, and being a lifelong learner.