

Forum:[Hack Forums](#)

Thread: [Open-source projects for beginners + profit of them?](#)

dreamerboy - December 1, 2019, 3:10 pm

Any suggestions where to start with open-source development as a first timer? I have experience working with Git and experience in JavaScript, Android, React and Structured Text (PLC programming)?? How can you monetize (make profit) and live off contributing towards open-source projects, because I have seen lots of experienced developer switching to open-source development?

Charger - December 1, 2019, 4:34 pm

You don't.

People see you contributing and reach out to hire you.

By you I mean the dude in India writing code for \$.15 an hour.

sdk - December 1, 2019, 8:01 pm

You can make a very popular open source product or suite of products, Github now has sponsor like system for highly used softwares / components.

dreamerboy - December 2, 2019, 7:30 am

(Dec 1, 2019 05:34 PM)Charger Wrote: [\[CITE\]](#) You don't.

People see you contributing and reach out to hire you.

By you I mean the dude in India writing code for \$.15 an hour.

Half True, therefore I agree

(Dec 1, 2019 09:01 PM)sdk Wrote: [\[CITE\]](#) You can make a very popular open source product or suite of products, Github now has sponsor like system for highly used softwares / components. Cheers, I will google it!

MavenGradle - December 2, 2019, 9:29 am

(Dec 1, 2019 05:34 PM)Charger Wrote: [\[CITE\]](#) You don't.

People see you contributing and reach out to hire you.

By you I mean the dude in India writing code for \$.15 an hour.

(Dec 1, 2019 09:01 PM)sdk Wrote: [\[CITE\]](#) You can make a very popular open source product or suite of products, Github now has sponsor like system for highly used softwares / components.

(Dec 2, 2019 08:30 AM)dreamerboy Wrote: [\[CITE\]](#) Half True, therefore I agree

As an open-source developer myself (in my spare time), I 100% agree with [Charger](#) and the fact of the matter is that donations don't really happen all that often. I have repositories that have over 100 stars yet I only receive maybe \$10 - \$20 USD a year from donations. Not even enough to run a VPS for a month. The real money is actually in getting noticed and recruited to do various jobs.

Open-source development is more or less a way for you to display your skills to future employers and in some cases can even help skip the job interviews entirely (which is usually the case when I

apply somewhere).

dreamerboy - April 21, 2020, 9:18 am

(Dec 2, 2019 10:29 AM)MavenGradle Wrote: [\[CITE\]](#) As an open-source developer myself (in my spare time), I 100% agree with [Charger](#) and the fact of the matter is that donations don't really happen all that often. I have repositories that have over 100 stars yet I only receive maybe \$10 - \$20 USD a year from donations. Not even enough to run a VPS for a month. The real money is actually in getting noticed and recruited to do various jobs.

Open-source development is more or less a way for you to display your skills to future employers and in some cases can even help skip the job interviews entirely (which is usually the case when I apply somewhere).

Thank you for your example and clarification, this was useful data for me ;)

_Cryptomic - April 26, 2020, 11:19 am

(Dec 1, 2019 04:10 PM)dreamerboy Wrote: [\[CITE\]](#) Any suggestions where to start with open-source development as a first timer? I have experience working with Git and experience in JavaScript, Android, React and Structured Text (PLC programming)?? How can you monetize (make profit) and live off contributing towards open-source projects, because I have seen lots of experienced developer switching to open-source developement?

Github is your buddy.

Dood - April 26, 2020, 5:13 pm

Realistically, the only ways to work on open source and get paid are contracting jobs which require you to make modifications to open-source software for a specific purpose or to work for a company that manages and develops open source projects (e.g. Kubernetes team at Google, React team at Facebook, Kafka team at LinkedIn or Confluent, etc.).

I'm the latter; I work on one of the Apache Software Foundation open-source projects full-time (not top level, probably graduating from the incubator within a year). Most of my time is spent contributing to the open-source codebase for the project, but I also spend a good amount of time working on internal company stuff for our specific use case of the project.

As for how to start contributing, many beginners just look around the project docs and check for spelling issues or grammar issues and such; once you find one of these, you can follow the submission guide (most projects on GitHub will allow you to just make a pull request, but check to be sure) and wait for someone to approve your pull request/patch. If you want to make contributions to the actual code, I recommend playing around with the software first. It'll give you a good feel for how everything works, and it may give you a better understanding of how the codebase works. I would also recommend taking a look at the design docs for the software, if you want to make core changes.

dreamerboy - April 27, 2020, 4:36 am

(Apr 26, 2020 05:13 PM)Dood Wrote: [\[CITE\]](#) Realistically, the only ways to work on open source and get paid are contracting jobs which require you to make modifications to open-source software for a specific purpose or to work for a company that manages and develops open source projects (e.g. Kubernetes team at Google, React team at Facebook, Kafka team at LinkedIn or Confluent, etc.).

I'm the latter; I work on one of the Apache Software Foundation open-source projects full-time (not top level, probably graduating from the incubator within a year). Most of my time is spent contributing to the open-source codebase for the project, but I also spend a good amount of time working on internal company stuff for our specific use case of the project.

As for how to start contributing, many beginners just look around the project docs and check for spelling issues or grammar issues and such; once you find one of these, you can follow the submission guide (most projects on GitHub will allow you to just make a pull request, but check to be sure) and wait for someone to approve your pull request/patch. If you want to make contributions to the actual code, I recommend playing around with the software first. It'll give you a good feel for how everything works, and it may give you a better understanding of how the codebase works. I would also recommend taking a look at the design docs for the software, if you want to make core changes.

Cheers, thank you for sharing your experience