João Freitas

Scripts should be written using the project main language

24 Mar, 2024

This is an hot-take I've been feeling for a long time.

Almost all projects I've worked on have scripts we wrote to automate a **repetitive process**. Consecutively, the majority of the scripts became obsolete and unmaintainable after some weeks because we either didn't need them anymore or they have become too **complex** to be looked at. Afterward, these scripts are either refactored or they stay forever in the *scripts graveyard*.

Never forget to pay your respects for all the scripts we left to rot 🫡 ([Source](#))

I think one of the major reasons that led to this outcome is that we wrote the scripts in `Bash` or `Python`. Don't get me wrong, I'm a big fan of the two, but I feel like teams can **win** more if they write scripts using the **main** language their project is developed with. From the start, *time* and *effort* required to write them is reduced since:

- The learning curve is **minimal** since you already know the corners of the language
- Internal language APIs can be leveraged, which drastically changes the **mental model** to write the script (for the better)
- Scripts feel more **natural** and eventually **maintainability** increases. Team members are **familiarized** with the language!
- Development machines **compatibility** increases. Windows users can finally run all scripts. 🤭

The downside of going in this direction is the main language **runtime** support. Not every language is multi-paradigm or was designed to support scripting. For example, writing scripts

on `JVM` languages would require additional effort to build a **toolchain** that compiles and runs files on the fly, with a short start time. But if you work on a web frontend project (`JavaScript`), the power is in your hands.

At the end of the day, it is up to teams to talk and establish an **agreement** on which language fits best their needs and **standardize** their scripting language. If they can use the main language, awesome. If they can't, a higher-level scripting language with native support (e.g., `Python`) should be adopted, since it provides the means to increase maintainability in the long run.