

Web and Mobile Support for LibreOffice

In the past 10 years, TDF has invested into providing web and mobile support for LibreOffice. Although there was some releases, the products were eventually abandoned for non-technical reasons. Although ~5 years have been lost since then, there is still room for providing usable and competitive web and mobile application.

This proposal is written to address the needs that have been discussed over the past decade. The goal is to hire in-house developers to continue the development of web and mobile support for LibreOffice.

TDF has successfully hired developers in the areas of a11y, RTL/CTL/CJK, which both lead to significant improvements in these areas that were neglected by other developers during several years. Building on this background, and the focus areas of the suggested hiring, it is expected that the staff can handle the hiring in short term, and help the hired developers to start in relatively short time.

Focus Areas of the Developers

The underlying technologies used in LibreOffice Online, and also mobile version of LibreOffice is the same: LibreOfficeKit. Therefore, it is helpful to discuss things in a single document, rather than discussing things in different proposals.

LibreOfficeKit itself is written in C/C++. It is the back-end technology used to open, render and deliver document tiles to LibreOffice Online and also mobile versions of LibreOffice. LibreOfficeKit is used as a back-end library.

On the other hand, different front-end technologies have been used in web and mobile, which is platform-specific. For each of them, different programming language is used, but any of these languages are well-known and widely used on that specific platform.

The below table is the proposal in short:

Focus areas for the proposed developer hiring

Focus Area	Programming language / Technology	Hire count
Back-end for Online	C/C++	1
Front-end for Online	JavaScript, TypeScript	1
Android	Java	1
iOS	Objective C, Swift	1

LibreOffice Online

LibreOffice Online source code repository is frozen since 2020, around 5 years ago, and is now atticized. The freezing was supposed to be temporary. It is now stored as a read-only repository, where no contributions is possible:

<https://git.libreoffice.org/online/>

Since then, there has been various improvements in LibreOffice itself, LibreOfficekit, and online counterparts.

LibreOffice online consists of various modules, but it can be divided into two different parts:

1. Server side: mostly written in C/C++
2. Client side: mostly written in JavaScript/TypeScript

Therefore, two different roles have been defined for online:

First, one back-end developer, which should know C/C++. And second, one front-end developer, focusing on JavaScript/TypeScript development.

Android Version

Currently, TDF provides two different Android applications for LibreOffice, which are for two different purposes:

1. LibreOffice Viewer (with experimental editing support)
2. LibreOffice Impress Remote

The plan is to consolidate these two applications, and improve the editing capabilities, promoting it out of experimental status to a working by default status.

1 developer hiring is proposed in this focus area.

iOS

Apple iOS is another popular mobile platform, and there is a high demand for providing LibreOffice application on this platform.

Although there is no iOS application from TDF at the moment, the underlying architecture and technologies are mostly similar to what is used in LibreOffice Android viewer. Therefore, an early beta is expected to be released in short term.

1 developer hiring is proposed in this focus area.

Conclusion

The goal is to hire 4 in house developers to cover LibreOffice web and mobile support. The Executive Director, or other staff member appointed by the ED, shall direct management for the developers to ensure they effectively focus on their development areas, will follow the progress and address eventual issues.

The development work of the in-house developers will include fixing bugs and implementing features in full, fixing bugs that are blockers for community contributors and to help evaluating which complex tasks should be tackled by external specialists.

Regular weekly exchanges between the team and ESC will help the team to independently define the tasks to be performed so that the most relevant topics will be prioritized and overlaps will be avoided.

While the main areas of relevance are listed in this document, there may be situations where the in-house developers will work with the team to help fixing urgent issues, such as implementing security patches or app stores required changes, if skill sets and experience allow.

The in-house developers will have the same rules, rights and conditions as any other volunteer or corporate contributors. Overlaps or prioritization that find no clear conclusion within team will be evaluated by our Executive Director with the support of our mentors.

In-house developers will follow agile development methodologies, including short-term goals and delivery times.

In case the developers need to spend time mentoring community developers, the expectations may be adjusted by the Executive Director with the assistance of our current mentors.

In-house developers should be able to provide performance levels comparable to other developers working in the same area after an on-boarding and mentoring period of approximately 6 months.

In-house developers will greatly benefit from the experience accumulated by the team while also providing our current mentors a way to improve their mentoring techniques within the team to better serve other developers. The new members of the team may also provide mentoring services, once deemed ready by our current senior mentors, which will be scheduled in a way that will be compatible with their primary role as in-house developers.

In-house developers should be able to present their activities, discuss their achievements and possible obstacles on a regular basis in team and/or ESC meetings depending on the relevance of the topics to be discussed.

It is expected from the developers to set together with the team the planning of their activities and to participate to weekly team meetings.

TDF's staff will evaluate and report to our ED any attempt made by community members and/or third parties to propose tasks for our in-house developers that are distracting them from the work assigned to

them by the ED. If ED suspects the request is driven by a personal or commercial interest, the ED will consult the Board to decide about eventual further actions.