

22.01.2021 // Antwort Repair-Cafe

„Sehr geehrte Damen und Herren,


eignet sich Ihre neue Plattform auch für Repair-Cafes? Ich suche schon lange so ein Ticketsystem, damit ich z.B. nicht mit meinem kaputten Breitbild-Fernseher mit Bus & Bahn zum nächsten Repair-Cafe fahre um dann zu erfahren, dass keiner Zeit für mein Problem hat oder sich nicht auskennt und ich unverrichteter Dinge wieder nach Hause fahren muß. Die Dingfabrik in Köln ist sowieso der Meinung, dass so ein Cafe als Schwerpunkt "Cafe und Kuchen" hat, weil es ja alles Freiwillige sind, die das machen? Wenn die freiwillige Feuerwehr auch diesen Standpunkt vertreten würde, dann würde Ihr brennendes Haus nicht mehr gelöscht, oder wenn andere Dienste wie z.B. das DRK, dass auch vorrangig von Freiwilligen organisiert wird, so denken würde, dann würden keine Blutspendetermine mehr laufen um Leben zu retten, oder die Corona-Impfungen könnten nicht stattfinden, weil die Ehrenamtlichen gerade Kaffee trinken und Kuchen essen. Aber die Betonköpfe von der Dingfabrik lassen sich leider nicht umstimmen und lehnen mein Engagement ab. Daher versuche ich schon länger etwas Alternatives auf die Beine zu stellen durch ein Repair-Wiki oder eine Plattform, wo man seine Dienste bereitstellen kann und Leute darauf reagieren können. Aber da ich nur eine Privatperson bin, weiß ich zur Zeit nicht wie ich es zeitlich, räumlich und personell organisieren soll? Sind Sie auch der Meinung, dass man weiter der Umwelt schaden sollte und massiv Elektroschrott produzieren soll, weil es kein Recht auf Reparatur gibt? Oder würden Sie auch für den Umweltschutz eintreten und mir helfen meine Idee in die Tat umzusetzen? Leider kenne ich mich mit "Cap'n Proto" nicht aus. Warum schreiben Sie es nicht mit PHP oder Python, dann könnte ich Sie unterstützen. Über eine freundliche und konstruktive Antwort würde ich mich sehr freuen.

Mit freundlichen Grüßen

Meinungen:

- Oh weih, ist der Mensch frustriert... (TMu)

- Er hat ein soziales Problem mit der Dingfabrik - das wird FabAccess nicht lösen. - selbst wenn die Dingfabrik (oder er) irgendwann FabAccess einsetzt. (-"-)
- Sehe ich auch so (TheJoKILa)

-  Freundliche und konstruktive Antwort > Diese Mail

Kurzantworten: eignet sich Ihre neue Plattform auch für Repair-Cafes? Prinzipiell ja, sofern RepairCafes Maschinen zur Nutzung anbieten und Einweisungen dafür geben. Allerdings eher nicht, für dass was Sie vorhaben ... wir werden aber nicht in das Termin-Management von Repair-Cafes eingreifen und das wird FabAccess auch nicht darstellen.

Ticketsystem Für ein Ticket System gibt es bessere Alternativen, Jira oder RequestTracker zum Beispiel.

Sind Sie auch der Meinung, dass man weiter der Umwelt schaden sollte und massiv Elektroschrott produzieren soll, weil es kein Recht auf Reparatur gibt? Nein

Oder würden Sie auch für den Umweltschutz eintreten und mir helfen meine Idee in die Tat umzusetzen? Können wir leider nicht (... wir sind mit FabAccess ausreichend beschäftigt... und nicht in Köln.)

Warum schreiben Sie es nicht mit PHP oder Python, dann könnte ich Sie unterstützen. 0. Weil ich will. * PHP / Python ist nicht besser als \$Language, auch nicht als Rust. * Rust ist nicht besser als \$Language, auch nicht als PHP. * Aber ich schreibe die software nunmal anstatt mich drueber auszuhaeulen wie schlimm \$Repaircafe ist. Also ist es Rust.

1. Warum nicht PHP/Python sondern spezifisch Rust?

- Geschwindigkeit
 - Und damit Effizienz. Die Tatsache das wir Python verwenden traegt inzwischen nicht unwesentlich zum Klimawandel bei.
- Sicherheit
 - Gegen Bugs die in Netzwerksoftware oft Exploits verursachen, z.b. double free, buffer overflow (vgl. Heartbleed), buffer underflow aber auch gegen Logik-Bugs durch starke Typen
- Stabilität
 - Sowohl auf ABI-level, Rust hängt am Ende nur noch von der glibc ab
 - Die Software selber wird dadurch weniger warscheinlich abstürzen
- Ich kann Rust besser als Python und wesentlich besser als PHP

2. Cap'n Proto ist keine Programmiersprache sondern eine Schemasprache und definiert die Schnittstelle sinnvoll damit externe [Personen] Clients sinnvoll schreiben können und die nicht ständig in die Luft fliegen weil irgendwas

irgendwo eine Sache im Api ändert die inkompatibel ist

3. Es wird eine Python Implementation der API geben, mit der kann dann ein eigener Client geschrieben werden

Antwort

Sehr geehrter Herr M....,

unsere Software FabAccess ist zur Verwaltung von FabLabs und offenen Werkstätten gedacht, um so den Umgang mit Maschinen sicherer zu machen. Für den Einsatz in Repair-Cafes eignet sich unsere Software nur, wenn in diesem Umfeld Maschinen oder Gegenstände verliehen werden.

Für das Vorhaben was Sie anstreben bietet sich aus unserer Sicht eher ein Open Source Ticket System wie Jira oder RequestTracker an. Teilweise wird so etwas auch z.B. auf Discourse abgebildet. Auf einer solchen Plattform könnte Ihre Idee aufgebaut werden.

Wir unterstützen natürlich den Ansatz von Repair-Cafes, mehr zu reparieren und weniger zu entsorgen. Manche von uns sind auch in RepairCafes direkt aktiv und beteiligt. Allerdings kann unsere Software bei diesem Problem nicht helfen.

Leider können wir Ihnen bei Ihrem Projekt nicht helfen, aber wir hoffen wir konnten Ihnen einen Weg zeigen, wie Sie mit Ihrem Projekt starten können.

Die Toolwahl wurde tatsächlich am Projekt-Start ausführlich diskutiert und ist für unser Projekt den Kenntnissen unserer Entwickler gefolgt, um in der kurzen Zeit der Förderphase einen langfristig schnell und stabil laufenden und möglichst von Programmierfehlern freien Prototypen zu erhalten. Der Rust-Compiler verhindert an der Stelle Fehler, die in anderen Programmiersprachen gerne übersehen werden und produziert dabei sehr effizienten Maschinencode ohne viele externe Abhängigkeiten.

Als Info für Sie: Bei Capt'n Proto handelt es sich nicht um eine Programmiersprache die wir verwenden, sondern um die Schemasprache mit der wir unsere API definieren. Für unseren Anwendungszweck, eine möglichst stabile und erweiterbare API zu bauen eignet sich Capt'n Proto daher besonders gut. Über diese Herangehensweise können wir anderen Open Source Entwicklern wie Ihnen eine Möglichkeit geben unsere Software zu erweitern, wie zu Beispiel einen eigenen Client zu schreiben.

Wir hoffen wir konnten Ihre Fragen alle beantworten und wünschen Ihnen noch viel Glück und Erfolg bei der Umsetzung Ihrer Idee, die Repair-Cafes zu revolutionieren.

Mit freundlichen Grüßen

Version #1

Erstellt: 15 Oktober 2024 10:41:33 von Mario Voigt (Stadtfabrikanten e.V.)

Zuletzt aktualisiert: 14 Dezember 2024 18:23:07 von Mario Voigt (Stadtfabrikanten e.V.)