Get better at programming by helping others on Glitch

Nicolas Grenié Nov 8 '17 · 4 min read EDIT MANAGE

#beginners #devrel #javascript #serverless

I started to learn programming about 15 years ago. In a time without Github and very limited resources to learn. A lot of my early learnings were made through imitation of existing websites. I would check their source code, copy-paste parts of it on mine and try to understand what it was doing by modifying it.

After few years of practice, I had enough basic knowledge to help other beginners with their questions on bulletin boards. This was a great way to give back to the community that welcomed me when I began. I really enjoyed it because I got to put myself in someone else's shoes, understanding their issues and walk them through a solution. This teaches you pedagogy and empathy while you are re-enforcing your coding knowledge.
Now that I think about it, those early days were probably the foundation of my Developer Advocate career 😇

Contribute and give back 🤝 🤗

When you just started to learn a technology it could be really intimidating to give back to the community. It's not easy to submit your first Pull Request on Github. Or answer your first question on Stack Overflow. People have created initiatives

like YourFirstPR or Hacktoberfest to lower the barrier to entry and make PR less scary.

I feel that contributing and giving back got a lot easier with tools like Glitch.
I already mentioned Glitch in previous posts, especially from the Developer Advocate perspective.
With the *remix* feature, you can clone a project and add your own sauce to it.
If you ever get stuck you can ask the community for help with the *Help* button.

If you are interested to help others, you can check often the Glitch homepage and see if people need help. You can join people's project, offer help and start directing them to find a solution. At the end, they will thank you with a 💖 on your profile.

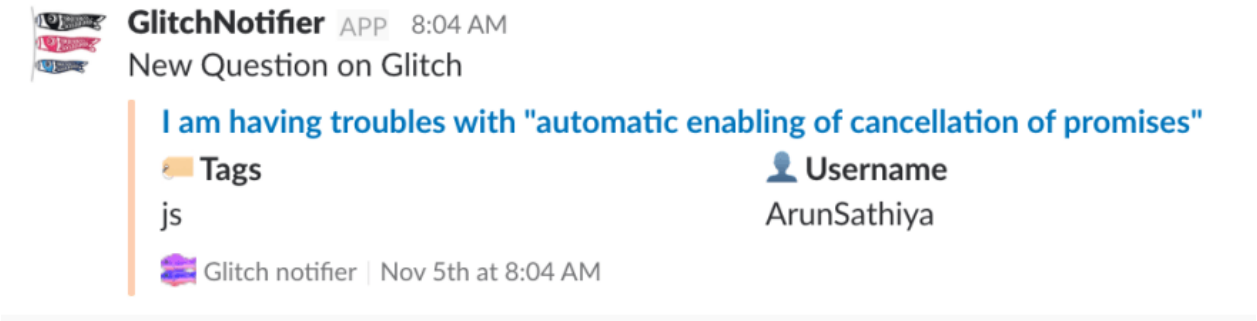
But questions are still rare, and popup at random times, so you might not satisfy your desires of community collaboration.

I wanted to get more involved so I created a small tool to be notified whenever someone asks a question.

Meet: Glitch Notifier.



I am a heavy user of Slack, so naturally, I wanted to get notified directly on Slack.



Glitch Notifier is open for the community to use. You log in with Slack, you choose the tags you want to follow, and add the Notifier app to your Slack team. You will then receive notification any time a question matches your criteria.

You can contribute to the project by remixing it.

I've been testing this tool for a few weeks now. I helped about 10 people and received 💖 8 times. There were about 100 questions asked in a month, on a variety of topics. I remember helping someone set up their environment variables or someone building a chatbot in Messenger...

No matter your experience or your skillset, you are able to help someone in the community!

If you are interested to learn how it was built, keep reading 😉

How is it built? 💻🛠️

Like in the old days, looking at page's source gives you a lot of information. That's how I found out that Glitch has an undocumented API.

So far I've found 3 endpoints:

- GET https://api.glitch.com/projects list projects
- GET https://api.glitch.com/users list users
- GET https://api.glitch.com/projects/questions list current open questions

The last endpoint is the one we are interested in. It returns an array of questions or an empty array when no question is asked.

To check periodically (every minute) on this endpoint to see if there are open questions, I have set up a small Lambda function (github repo) using Serverless.

If there are questions, I send them to a specific endpoint (`/questions/hook`) on our Glitch app.

There, we will check if we already have the question in our database. We use FaunaDB, as our cloud database.

If the question is not yet in our database we should notify our users.

Doing a Map request on our database we find which users are following the question's tags, and we notify them on Slack.

Here is the corresponding code

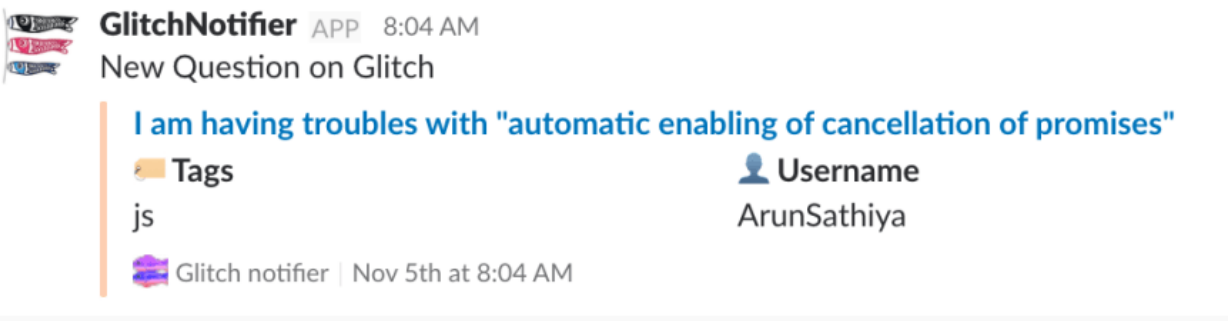
```
var notifyPeople = function(question) {
  var users_in_db = client.query(
```

```
            q.Paginate(
            q.Union(
              q.Map(question.details.tags,
                function(tag) {
                  return q.Match(q.Index("user_by_tag"), tag)
                }))));


users_in_db.then(function(result){ // request match results
   var usersRef = result.data
   usersRef.forEach(function(userRef){
     var user = client.query(q.Get(userRef));
     user.then(function(u){
       slack.sendQuestionToSlack(u.data.incoming_webhook.url, ques
     })
   })
}).catch(function(err){ // does not exist
  console.log("err", err)
})
```

To notify people on Slack we use an incoming webhook integration, with a bit of formatting so it looks nicer.

Result in a Slack channel:



Hope you like it, let me know if you have any question or comment.

Photo by Štefan Štefančík on Unsplash



Nicolas Grenié EDIT PROFILE

Passionate Hacker, Developer Evangelist at Typeform, Love APIs, Serverless and Bots

@picsoung picsoung picsoung nicolasgrenie.com