

Stackoverflow.com from lurker to active contributor

Mohammad Faani

<https://github.com/prohoney/StackoverflowLurkerToContributor>

Overview

- This talk is about my journey with Stackoverflow.com
- The potentially harsh community it has
- Common mistakes
- Why having an account makes a whole lot of difference
- Alternate ways of using the site
- Stackoverflow.com alternates

Docs are nice
But so are errors



3

My Journey

My Journey

- Duplicate question . . . Question closed!
- Couldn't comment



I was learning HTML. Through Codecademy on my own.
I had some questions about PHP

My question got closed. I wasn't exactly welcomed into the site.

My Journey



8 years.

This site sucks!

Everybody on Stackoverflow is a jerk!

I'm just trying to communicate how I felt. This is less of an issue for people with CS degrees. Yet even to this day, rarely I see my questions get closed...

My Journey



It wasn't exactly 17 years. It was closer to 2years. But then I found myself a mentor for iOS and was going through a course, I was going to meetups.

I had a more structured mind

First Question and point

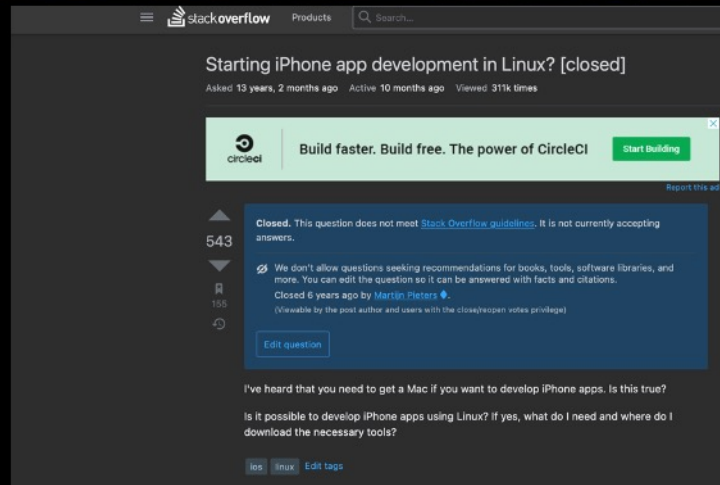
- [First Question and point](#)
- [First question that got good traction](#)
- [First Edit](#)
- [First Comment](#)
- 1000 points
- First accepted answer

Stackoverflow.com mistakes

- Asking about opinionated things
- Posting without code
- Posting image instead of a code
- Posting excessive code
- Incorrect/insufficient tagging
- Sharing link instead of code
- Ask a totally new question in a comment
- Ask too many questions in the comments

Before we dive deeper, let me discuss some beginner mistakes on Stackoverflow.com

Mistake – Asking for recommendations



Mistake - Duplicate

396 results

Relevance Newest More

1
Q: Swift fatal error (duplicate)
I want to go back to my previous controller when I hit logout button, but it gives me: fatal error: unexpectedly found nil while unwrapping an Optional value navigationItem.rightBarButtonItem = UIBarButtonItem(title: "Logout", style: .done, target: self, action: #selector(self.logout))
asked Dec 2 '16 by Vika

130
Q: Fatal error: unexpectedly found nil while unwrapping an Optional value (duplicate)
I was using a UICollectionView in Swift but I get when I try to change the text of the cell's label: func collectionView(collectionView: UICollectionView?, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell { let cell = collectionView.dequeueReusableCell(withReuseIdentifier: "Cell", for: indexPath) as! UICollectionViewCell cell.textLabel?.text = "Next line" fatal error: unexpectedly found nil while unwrapping ...
asked Jul 9 '16 by Vika

-1
Q: Thread 1 Fatal Error Optional Value Swift (duplicate)
completion { user, error in if let error = error { print(error.localizedDescription) } } I get this error: Thread 1: Fatal error: unexpectedly found nil while unwrapping an Optional ...
asked May 21 '16 by chul

21
Q: fatal error: unexpectedly found nil while unwrapping an Optional value (duplicate)
I am trying to run this code but I keep on getting this error: fatal error: unexpectedly found nil while unwrapping an Optional value I don't understand what it means or why I'm getting it ...
asked Jul 25 '14 by Arlo

-4
Q: swift3 -- fatal error: unexpectedly found nil while unwrapping an Optional value (duplicate)
I get fatal error in Swift 3 when call this code line: let dummyInformation = getDummyInformationFromInternetArticle let centerLongitude let date let count let count Seems that type of centerLatitude ... error is as follows: fatal error: unexpectedly found nil while unwrapping an Optional value (SwiftCore.dylib: function signature specification of Swift3 ...
asked Mar 10 '17 by user1388448

Why points matter?

- Comments
- Voting
- Extra visibility
- Bounty

Why am I focusing on the point system? Not because it's cool or anything.

Visibility in deleted posts, downvotes, site analytics, etc.

Rather until you get to 50 points where you can comment like others, you don't really enjoy much using the site, so that's why.

Point system

- Question upvote: 10
- Answer upvote: 10
- Accepted answer: 15
- Comment: 0
- Accepting an answer: 2
- Successful edit: 2
- Downvote answer: -2

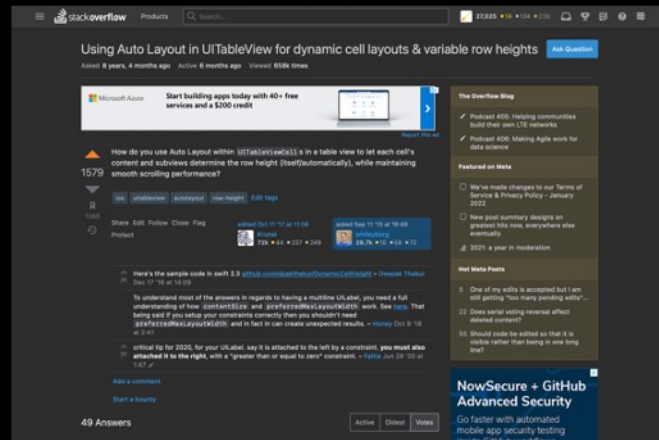
When do I post answers?

- [I write my own question and answer](#). Another [example](#)
- I find an [alternate answer / explanation](#) to a question/error message I ran into myself
- When I discuss something with folks on Slack/work and learn something new I try to [share it with the outside world](#)

I'm not actively searching on the site.

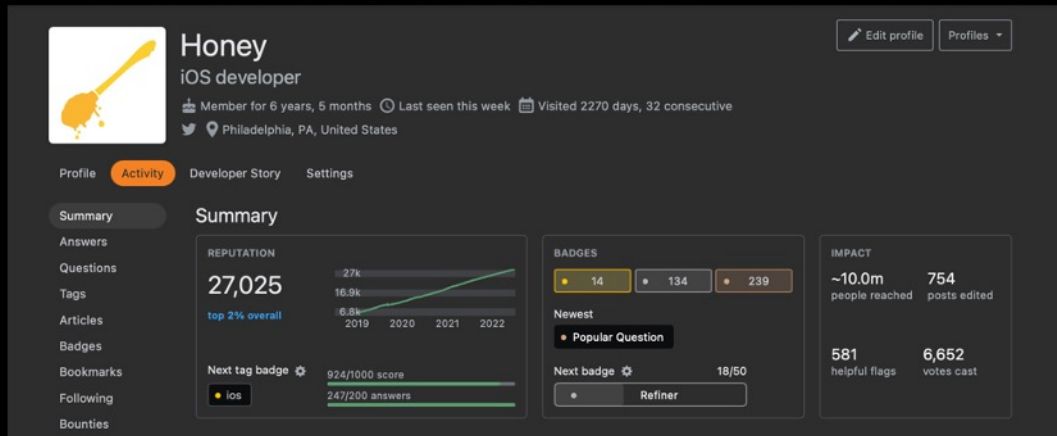
Why you should join
Stackoverflow.com

Comparison



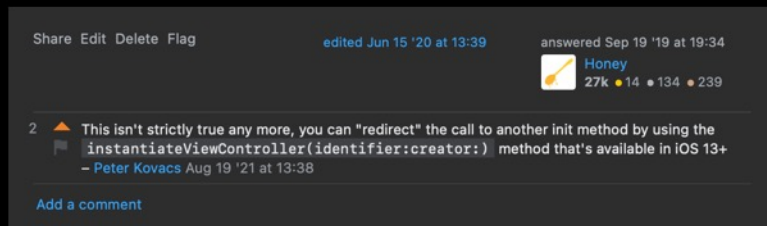
It's good to see how your answers compare with others

You can market yourself differently












One can claim that their contributions to the community has reached 10M people

I get updated with the latest on my posts



One can claim that their contributions to the community has reached 10M people

Each message, upvote jolts my memory

> 32	added an image a note on how to create good slides Source Edit Rollback Link	edited Sep 4 '20 at 12:27  heavy 27k ● 14 ● 134 ● 239
> 31	added 10 characters in body Source Edit Rollback Link	edited Aug 3 '20 at 14:14  heavy 27k ● 14 ● 134 ● 239
> 30	added 250 characters in body Source Edit Rollback Link	edited Jul 21 '20 at 19:56  heavy 27k ● 14 ● 134 ● 239
> 29	edited body Source Edit Rollback Link	edited May 29 '20 at 2:19  heavy 27k ● 14 ● 134 ● 239
> 28	added another simple example - removed the references to subtasks as subtasking is not necessary Source Edit Rollback Link	edited May 2 '20 at 16:55  heavy 27k ● 14 ● 134 ● 239
> 27	deleted 9 characters in body Source Edit Rollback Link	edited Apr 23 '20 at 16:39  heavy 27k ● 14 ● 134 ● 239
> 26	improved wording Source Edit Rollback Link	edited Mar 31 '20 at 3:45  heavy 27k ● 14 ● 134 ● 239
> 25	improved example and formatting Source Edit Rollback Link	edited Mar 27 '20 at 18:10  heavy 27k ● 14 ● 134 ● 239
> 24	improved example Source Edit Rollback Link	edited Mar 27 '20 at 17:01  heavy 27k ● 14 ● 134 ● 239

on that subject. They force to me revisit and update my answer. It's like constantly being forced into a mirror, where you both see your beauties and faults. I make updates as I find better ways to explain things

Points

- “Recognition has a profound effect on the workplace, **encouraging more prosocial behavior**. The Positive Psychology Program defines “prosocial” as “promoting others' wellbeing, usually through altruistic acts.” This attitude in the workplace generates more teamwork, more productivity, and more communication between teams”
- Recognition motivates you by challenging your inner procrastination. It pushes you to write and contribute for the sake of community and your own personal brand.



<https://www.fond.co/blog/prosocial-behavior-psychology-rewards/>

It's about positive energy and feeling useful...

I often come back to my own answers

Usage:

- `git checkout HEAD~1` will actually GO/checkout to 1 commit/reference before
- `git reset HEAD~3` will uncommit your last 3 commits — without removing the changes, ie you can see all the changes made in the last 3 commits together, remove anything you don't like or add onto it and then commit them all again.
- `git reset --hard HEAD~3` will uncommit your last commit **and remove their changes**. It will completely remove those changes. For more see [here](#).
- `git diff HEAD~3` to look into the changes of the last 3 commits
- `git diff someFile HEAD~3` to look into the last 3 changes of a specific file
- `git revert --no-commit HEAD~3..HEAD` . Make a new commit by reverting last 3 commits: For more see [here](#)

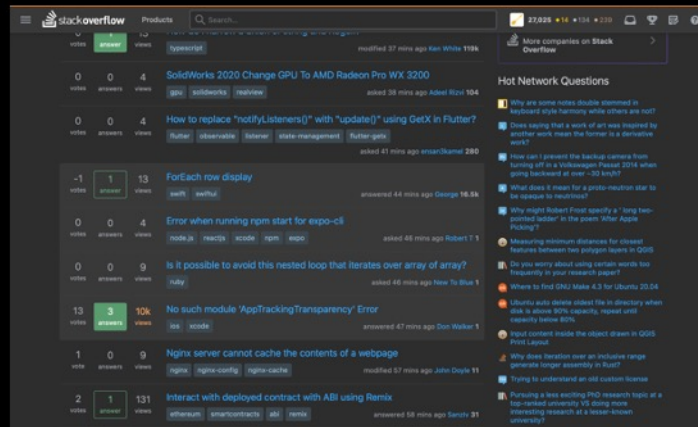
Developer Story

- [Rob Napier Developer Story](#)
- [Mohammad Developer Story](#)

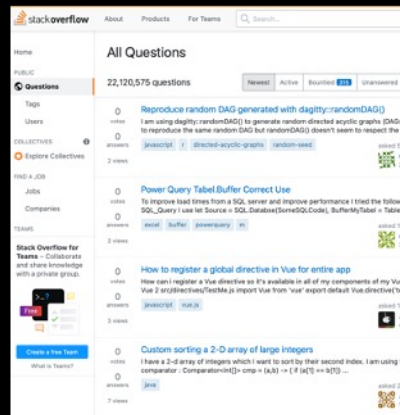
It's best in class for a dev-friendly timeline of all your accomplishments, best posts, book reads, presentations, experience.

Not even LinkedIn comes close to this. LinkedIn adds a whole lot clutter

Why join? – Personalized content



When you haven't joined



Other reasons to join

- My primary reason for answering SO questions isn't to help people or to gain rep or to show off what I know. My favorite questions are ones that I don't know the answer to either. They force me to go research. Maybe even write some code. And when I'm done, I know something I didn't know before. I don't answer SO questions because I'm a good developer. I'm a good developer because I spent so much time answering SO questions.
- Now I've been learning new languages where I'm not an expert, and I've found I'm starting to answer a few more SO questions. They're smaller communities (Scala and Go) and there's a lot more opportunity to again find questions that are interesting and I don't know the answer to already. Being less of an expert makes me more of a help. And being more of a help lets me learn faster.



Rob Napier

<https://meta.stackoverflow.com/a/253289/5175709>

Alternate ways of using the site

Chat

3 Q: swift3 - How to protect secret key

I am new to iOS swift 3 development. Now, I am working on a project which needs encryption, message authentication codes(MAC) and message digests. These algorithms require secret keys. I know that it is a bad practice to hard-code the keys inside the code, but the key = "secretkey123" does...

There always to check it? If you're worried about an amount of time on the order of the lifespan of the universe, then I guess so.

Alexander what do you do with a secret key? Plaintext/secret every API call you make? And then the server will also decrypt it using the same key it has given you?

Alexander Well your secret key can't be in the app binary, at all. If it's plaintext (like it was the case for longchat, lol), then it's trivial to see it. If you store your secret key in a hashed form, then you have access to your secret key (even you can't control it). If you use the hashed form of your secret key for encrypt/decrypting, then your hashed secret key is effectively your new secret key, and it's stored in plaintext, which fails. Encrypting the secret key doesn't work, because you only "lock the car down the road", because then you have to figure out how to store the encryption key.

But I think in my future, the key is to have the server provide the secret key after validation, over an encrypted channel (using a DH handshake to derive the encryption key for that channel). Plaintext/secret every API call you make? This is a separate question, of how to use the secret key after you have securely acquired and stored it. That secret key should never leave the device. It should be used to locally sign/encrypt (and going further, that the server will decrypt using the same key).

Alexander "What your secret key can't be in the app binary, at all." Isn't that what is being asked and recommended? I'm not trying to trigger a debate, rather just to confirm that there's a difference of opinion. In your post though I see a different answer from Rob Hopper too. So you disagree because obviously if a secret key is hashed, then all of them are. To your point though I see a different answer from Rob too. So you disagree because obviously if a secret key is hashed, then all of them are. Also doesn't your last sentence confirm my initial comment?

Alexander This conversation is suffering from the ambiguity of the term "key". There are two definitions in use in this conversation: API keys, and encryption keys. They have different threat models. API keys you want to keep secret from anyone but the developer, the app, and the API that requires them. As no happens, you can't really do much there. I was talking about encryption keys. There's no harm done if a user can figure out what encryption key is being used by one of their apps. It just needs to be protected from everyone else.

So my statements only conflict with Rob's if you confuse API keys and encryption keys. Otherwise, they're making two different statements within two separate domains.

Alexander Is encryption key == secret key?

Alexander I don't follow what you're saying. I'm just trying to understand how API key and secret key work together... How they are stored?

Alexander right. So API key is merely an identifier. But one that you don't want anyone else to know of, otherwise they could potentially get a free ride.

Exactly

so where does secret key come in to play?

So who are I (the dev) defending against?

I'm defending against other developers, which could be the user of my app. Ideally, only my app, and the API endpoint will ever have this API key, although that doesn't seem to be possible. This is what risk was talking about. Now consider a totally different situation, where I have an end-to-end encrypted chat app, where 2 users can use the app to communicate using an encrypted channel.

I thought Rob was talking about Secret key

What happens if the encryption key for that channel leaks?

My whole point is that "secret key" is an ambiguous phrase

It's in 5 minutes. I have to move

Alexander let me take a step back, how is the API key passed from client to server? It's just sent as part of the payload?

yes, hopefully through an encrypted channel, like in the case of HTTPS over TLS (HTTPS)

ok

so back to your chat question: how is that end-to-end encryption done?

So what happens if the encryption key for that channel leaks?

My whole point is that "secret key" is an ambiguous phrase

or it's encrypted with the other user's public?

what?

Is it encrypted by the encryption key is it symmetric?

Bounty

4 Answers

Active Oldest Votes

▲ This is a common misunderstanding among iOS developers.

56 ▼ Here's one of my "golden rules" for Auto Layout:

✓ **Don't bother about "updating constraints".**

You never need to call any of these methods:

- `setNeedsUpdateConstraints()`
- `updateConstraintsIfNeeded()`
- `updateConstraints()`
- `updateViewConstraints()`

except for the very rare case that you have a tremendously complex layout which slows down your app (or you deliberately choose to implement layout changes in an atypical way).

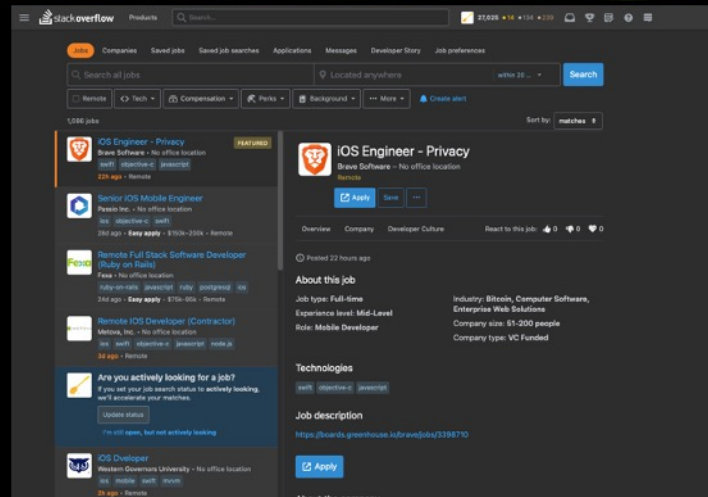
The Preferred Way to Change Your Layout

Normally, when you want to change your layout, you would activate / deactivate or change layout constraints directly after a button tap or whichever event triggered the change, e.g. in a button's action method:

```
@IBAction func toggleLayoutButtonTapped(_ button: UIButton) {
    toggleLayout()
}

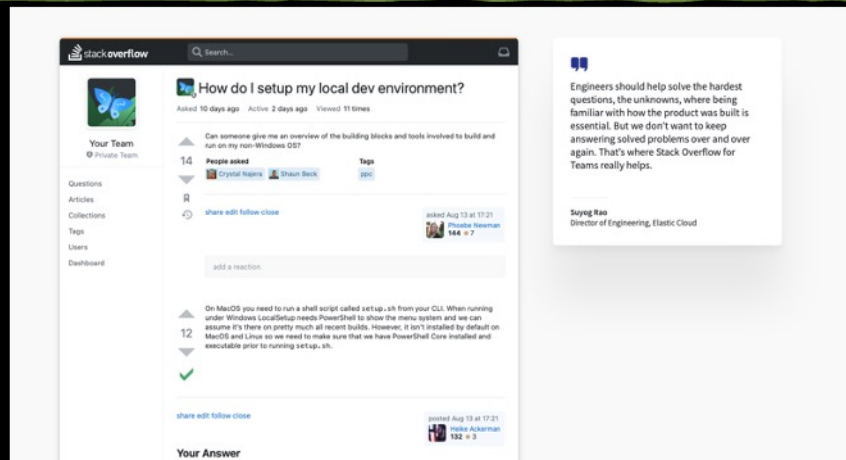
func toggleLayout() {
    isCenteredLayout = !isCenteredLayout
    if isCenteredLayout {
        centerXConstraint.isActive = true
    } else {
        centerXConstraint.isActive = false
    }
}
```

Jobs



Amazing ways to filter for job posts. Often you can apply all the way within SO

Private communities



Power users

- [Core Location Power User](#)
- [Encryption Power user](#)

Soft skills

- The fear of being judged shouldn't lead us to use Stackoverflow
- New users
- Upvote and recognizing efforts
- Be gentle with mistakes
- Every improvement counts

Being able to ask questions at work, helps others to learn. It saves us time by asking an internal person who knows the whole system

Be welcoming to new users on Stackoverflow.

- Offer edits.
- give feedback. Don't downvote for new members.
- some are youngsters who don't understand the value of structured questions. They might say "super thanks" or have terrible English.

Similar to SO we can potentially react negatively at work with new members...

Upvoting, giving positive feedback is always energizing. Giving bounty to a well written answer

Make suggestions rather than strong corrections

Every improvement counts. I correction to a tag, formatting makes it a whole lot easier/faster. You can do the same for the READMEs, Confluence pages, Jira tickets, add labels, improve structures

Stackoverflow.com alternatives

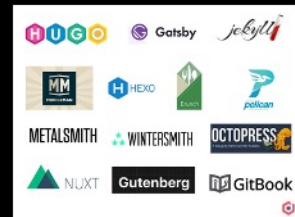
- GitHub
- Quora
- Slack
- Medium
- Personal Blog



Quora



Medium



GitHub - Some people just write their best work on their own GH Pages. Open source projects are a lot like islands. Depending on the project, they have a very unique CI/CD setup a custom dev environment. Your best bet is to look at their own docs. Their recent commits, their issues and try to break into their Slack or get in touch with one of the main contributors. Open source projects usually have very good documentation until they don't. e.g. their devs have their environments setup differently from docs and don't care to update.

Quora - When your question is open-ended. E.g. Swift vs Kotlin.

Slack - When you want to speak to someone or show live code, etc.

Medium – Great for blogging. Can also make money. Not owned by you...

Blog - (is right for personal growth, but it's also difficult to create the context). I remember someone lost all their company income because they were 100% relying on Facebook to allow their things to work. You can run into the same problem with Stackoverflow, because you don't own it

Quick Static site generator demo

[Hugo Quick Start](#)

Benefits of hugo is that it's a single binary as opposed to something that has node dependencies.

It's also super fast.

But I'm definitely not the expert on this. I just posted my first ever post in December.

<https://gohugo.io/hosting-and-deployment/hosting-on-netlify/>

...

baseURL: 'http://mfaani.com/'

languageCode: en-us

title: TIL with Mohammad

theme: paperMod

params:

ShowReadingTime: true

ShowShareButtons: true

ShowBreadCrumbs: true // to show parent pages of a subpage

ShowCodeCopyButtons: true

```
languages:
  en:
    languageName: "English"
    weight: 1
    menu:
      main:
        - name: Blog
          url: archives/
          weight: 5
        - name: About
          url: about/
          weight: 10
        - name: Search
          url: search/
          weight: 15
        - name: Tags
          url: tags/
          weight: 20
```

...

Thank you