

Google

Python(174)

python-fire

Python Fire is a library for automatically generating command line interfaces (CLIs) from absolutely any Python object.

Stars: 6910      fork: 339      updated: 2017-10-25

yapf

A formatter for Python files

Stars: 5713      fork: 384      updated: 2017-10-25

roboto

The Roboto family of fonts

Stars: 2896      fork: 243      updated: 2017-10-25

seq2seq

A general-purpose encoder-decoder framework for Tensorflow

Stars: 2823      fork: 566      updated: 2017-10-25

grr

GRR Rapid Response: remote live forensics for incident response

Stars: 2343      fork: 418      updated: 2017-10-24

nogotofail

An on-path blackbox network traffic security testing tool

## **Implement a repos viewer for GitHub repositories**

### **Goal:**

1. By complete the task, does **not** mean you getting hired automatically, it's just a tool that help to determine your skill level and experience.
2. Experience is only matter when it's up to date and constantly improve, whether it can meet today's challengers. Sorry, we don't count outdated knowledge as experience, so more years behind doesn't mean automatically it's better, if your skill is not keep up and improve with time. Your mileage may vary.
3. Try to do the best you can, everybody need to start somewhere. It's not end of the world, if you can't solve it. Good developer always keeps trying and improving, no matter of the outcome, remember that.

### **Requirements:**

1. Implementation in swift language with latest stable swift version.
2. Using Git commits and branches for implementation.
3. Short summary that describes design process, architecture decision, methodology, problem solving process, through and reflection regarding the implementation process. It doesn't need to be long, just precise, brief and to the point description of how you complete the task. What obstacle do you encounter under implementation, and what did you do to resolve the issues when it's occurred.

4. Research online if you are getting stuck, read documentation, google, check out sample code is allowed, inspiration and learning is part of the problem solving process. But please do not copy, paste code, if you don't understand what it does and its impact on the solution, you need to know exactly what you are doing and what you are trying to achieve by doing it.
5. Solving the task in 1-2 weeks' time frame from the day you receive the task, first part is the minimum, second, third part should only proceed after first part is finish and documented, if you have time to spare.

**First part: Implement basic data fetching, viewing logic without using third party frameworks, libraries with some exceptions.**

**Third party libraries that can be used for all parts of assignment:**

RxSwift

Realm

OAuth library of choice

GraphQL parse when using v4 GitHub api (for v3 GitHub REST api **NO** third-party http library/framework is allowed)

**!!! Other than these libraries/frameworks in exemption, using other third-party frameworks in first part of assignment is consider failing to solve task.**

## Languages

Python	172
Java	150
C++	136
Go	119
JavaScript	115
Dart	50
C	49
HTML	26
Objective-C	24
Shell	19

When typing owner/user name in search field, all repos belong to that owner/user/organisation listed as result in the list view below.

Group repos by language into sections, sorted by language with most repos on top with descending order.

Inside each language group section, sorted repos with most stars from top of the section to bottom with descending order.

See Result\_example.txt for what data could look like in list view.

Choose appropriate font, colour, symbol, icons to make UI representation more appealing.

**Second part: If you have plenty time, you can expand solution with these extra functionalities. (Optional)**

1. Caching result for offline access, database or custom http cache. If device is online, always showing the fresh result online. If device go offline, show any cache results on the device.
2. Implement type ahead search for owner/user look up.

Remember to commit your first part of implementation to git before starting second part, mark first part by tag or using branches.

**Third part: If you are getting bored, feel first and second part is too easy and not enough challenge for you. You could... (Optional)**

1. Implement detail view that display each repo's information in another screen when click and push from list view.
2. Display detail view as split view on iPad.
3. Improvement to the UI and usability.
4. Expand app with more functionality.
5. Sky is the limit. using your imagination and creativity...

Remember to commit your second part of implementation to git before starting third part, mark second part by tag or using branches.

For third part, third party library/framework limitation is lifted, you can choose free as you see fit... But ONLY if you solve the first and second part first.

With great power comes great responsibility, and you need to be able to justify your choice and trade off of using third party code, evaluate its quality, benefit and drawback, so choose wisely.

### **Note before start:**

1. Using force unwrap is consider an error in our company, linter will ensure project won't compile ... But why, if you have work long enough with swift project, then you know why :) But what if..., there is no but... If you don't understand crash and burn mean, then you will learn it the hard way some day.
2. Be careful with how you handle errors and exceptions.
3. Readable, reusable, flexible and easily maintainable is equally important traits as getting job done itself.
4. Keep an eye on memory leak and crashes.
5. Making complex problem simple is hard, and making it even more complex is easy, complexity on complexity will add even more complexity, your choice and approach can affect quality and maintainability in the long term.
6. Easy solution is not always the best solution, but try not to overcomplicate simple solution unnecessary, unless there is a good reason for it and effort is justified. Keep trade off in mind, clarity, efficiency and maintainability is the key.

**Info:**

1. Commit your solution to your open GitHub repository and share the link to implementation.
2. In case you are in time constraint, choose quality over quantity that best showcase your problem solving skill.
3. Second part and third part is optional, only if you have time left after solving the first part.