**Preliminary Questions**

1. What is your philosophy on TDD? Do you use it? **TDD is invaluable. It not only allows you to build and deploy software with a degree of confirmable reliability and integrity. It also saves debugging time in the long run. Yes, I use the heavily iterative form of TDD.**
2. What is your experience at working with remote teams? **I have done a large amount of remote work, for contract positions and full-time office positions that involved large amounts of remote days.**
3. How much experience (in months/years) would you say you have directly related to Rails? **I have 3 years of everyday Rails experience. Before the 3 years, I used Rails off and on for another 2 years.**
4. What is your experience with working on larger scale enterprise applications that are usually more complicated and complex than simpler web based applications? **I have designed, engineered, and built everything from simple scripts to large PCI compliant money (revenue sharing) distribution systems.**
5. What is your experience with JavaScript? **I have used JS on the web and as a standard development tool for many years. Everything from vanilla AJAX to prototype/jQuery and other JS libraries.**
6. Any particular JavaScript library you prefer? **jQuery, however I am flexible**
7. Do you have any open source contributions? **I have made contributions. I used to do a lot of development for a game bot. (I basically took over development after the lead abandoned the project.) I’ve also submitted patches to some Erlang database wrappers.**
8. What resources do you use to stay current? **Reddit, Blogs, RSS Feeds, Twitter, and my personal favorite is recorded conference talks.**
9. What are the five technologies that you are most proficient at and what are the top five that you are most interested in? **Proficient: vim, bash/zsh, unix/freebsd, ruby, rails. Interested: ruby, rails/sinatra, nosql dbms, iOS/Android, RubyMotion.**
10. You've determined that a critical piece of code needs to be refactored.  What approach do you take? **Depends. However, after ensuring that tests cover the code in question, generally I would use extract method and extract class to take overly complicated methods/classes and break them into smaller, modular, interchangeable parts. I would also extract any service layers or non-functional parts to create a clear separation between isolatable/easily testable pieces and more complex pieces.**
11. Describe the project that you are most proud of, and why? **I am most proud of my vim/cli note taking system. I am proud of it for a few reasons. 1. I take notes, manage todo lists, etc. without ever breaking my workflow. (By switching to some program or tool that isn’t in that workflow). 2. I’m a firm believer in knowing the tools of your trade thoroughly and using them efficiently. I believe this demonstrates that well.**