AKQA Coding Test Candidate Briefing

Candidate briefing

Hello!

Please find below a coding problem we'd like you to solve. The purpose of this test is not to get a "right" or "wrong" answer - it's to see how you code. So, show us what you've got!

You can spend as much or as little time as you'd like (as long as you submit your solution before the deadline), but make sure you tell us how long that was.

You should submit your solution in a way that makes it easy for us to use. You should zip up the code, and include Visual Studio .sln files if using .Net; if you're using Java, we'd expect to see an Ant or Maven build script. Unless you include documentation to tell us otherwise, we will expect to be able to build and run the solution with no changes. It should be quick and easy for us to verify that the input does indeed lead to the expected output. Please use English as the primary language in your solution to enable us to verify the behaviour.

An acceptance test is fine as a means to demonstrate that the solution fulfils the requirements. This could be via a console application, using a unit test framework, or any other means that make it easy for us to verify. **Note:** we don't want to be hosting web applications or performing complex configuration tasks just to verify your solution.

What are we looking for?

We want to see you can create production-quality code. This means naming conventions, coding style, sensible design and meaningful commenting. If you feel you can give your work to a brand new colleague with a minimal of hand-over, you've probably got it right.

We are fans of BDD and really big fans of TDD.

We realize that writing even a trivial application, and having it "production ready" is a lot of work - so it's okay to leave "to do" comments in your code. Show us the intent, but don't write too much boilerplate. Show that you're coding something that needs to be maintainable.

What are we not looking for?

We don't expect you to use every design pattern you've ever heard of - only apply patterns when it makes sense to do so.

We don't expect you to build a user interface - a command line application or acceptance test is fine.

We're not expecting you to have optimized the solution for performance or memory size - readability is more important than performance.

The whole thing built in the shortest amount of time - it's not a race. We'd like to see what you'd do under normal conditions - if you run out of time, just say so.

Make sure your solution does actually fulfil the functional requirements. One of the reasons that we like BDD and TDD is that it can be faster to get the behaviour right. With decent test coverage, refactoring to a better design can be a practical proposition later. If you find yourself in this position, a TODO comment or two can show us your intent.

Please Don't Copy Work

There are solutions to this test on the Internet, posted by previous unsuccessful applicants. If we find that you have copied your solution, we will not consider you for interview.

For the same reason, please do not share your work with others applicants or on the Internet.