

Test tasks for QA Engineer at relayr

Instructions

Please use Python for the second task; stick to one framework and do not spend more than 2-3 hours of your time. We are mostly interested in seeing the way you approach the problem and your coding skills. Rather than coming up with a complex but not finished solution, it's better to create a simple but finished and well-documented one. We would like to see a well-structured and maintainable code that is easy to read and understand. You could also add some comments in the code describing in simple words what you would do differently or what would you add if you had more time. Provide your solution in a git repository of your choice if possible or as a PDF document for Task 1 and as a zip archive for Task 2.

Task 1

Imagine the following situation: you are joining a cross-functional team which builds a front-end application using REST APIs. You are a first QA engineer and need to establish a QA process in the team.

1. What would you do in your first few days of work? Where would you start?
2. Which process would you establish around testing new functionality?
3. Which techniques or best practices in terms of code architecture and test design would you use in your automated tests?

Task 2

How would you approach test automation of an HTTP API of some service? Choose a service from this list <https://github.com/toddmotto/public-apis> or any other of your choice. It is better to pick one without authorization as it is easier to test.

Document several test cases. Implement one or two automated tests based on the test cases. Write the tests as if they were a part of a real project.

Task 2 alternative

Pick it instead of Task 2 if you think you are more proficient in frontend automation than in backend automation and this task will reveal your skills better. How would you test the search UI functionality of your favorite website (e.g. <https://medium.com>, <https://www.google.de>)?

Choose your favorite website which has search functionality. Document several test cases. Implement one or two automated tests based on the test cases.