

Home Challenge

Dear candidate,

Thank you for your interest and for dedicating your time to participating in our selection process. This Home Challenge is divided into two main parts. **We send you the best energy!**

Please submit your results by sending us a link to your **GitHub repository** containing:

- A separate folder containing test automation implementation and another one for the performance test implementation.

First task: API test automation

Your first task involves testing a RESTful API. We recommend using the Swagger Pet store Sample training tool. This tool is designed to provide hands-on experience with an API framework in a local environment. Please follow the instructions in the readme to set up your environment.

Here are the steps for this task:

1. Follow this link for getting information about the [Swagger Pet store project](#).
2. Set up the API Pet store and run it locally.
3. Propose a list of test cases for automation.
4. Automate the proposed test cases.
5. Provide a brief explanation of the solution you have implemented.

Please note that we are looking for a test automation solution that goes beyond a Postman suite, **please do not use postman as an automation framework**. We encourage you to **build a framework using the best coding/design practices**.

Second task: API performance test

The second part of the challenge involves conducting performance testing on the APIs exposed by the [Swagger series of Petstore APIs](#). Your task is to create scripts that can effectively test the performance of these APIs under different conditions.

You have the freedom to analyze and determine the coverage of test cases that you consider most relevant and impactful. The objective is to identify any potential performance issues that could impact the user experience or the overall functionality of the APIs.

Once the testing is complete, you will need to interpret the results and provide a comprehensive report.

You can find our technology stack below:

Our tech stack includes. Kotlin, Java, Docker, Cucumber, Gherkin, Gradle, Serenity, Rest Assured, Git, K6, Locust

We truly value if you use the same stack that we manage for this test, but feel free to use the stack that you know.