

Home Challenge

Thank you for taking the time to do our Home Challenge. It consists of three parts:

1. Performing an exploratory testing session of a money management application.
2. Writing a set of automated checks for an Android application.
3. Writing a set of automated checks for a REST API application.

Submission rules:

- All 3 tasks are **mandatory**. Please pay attention to all required points.
- Please submit your results by sending us a link to your **GitHub**, **GitLab** or **BitBucket** repository named {firstname}-{lastname} containing:
 - A single **markdown file** containing answers for Task 1
 - Separate folders containing each of the test automation tasks

Note: If you prefer to submit a private GitHub repository instead of a public one, let our Recruiters know and they will come back to you with list of GitHub users to give access to.

Tasks

1 - Exploratory Testing Session of the Monefy app

Perform an exploratory testing session of the **Monefy** mobile application. Choose the platform you are more familiar with (iOS or Android). You can download the apps here:

*

[iOS](https://itunes.apple.com/us/app/monefy-money-manager/id1212024409?)

*

[Android](https://play.google.com/store/apps/details?id=com.monefy.app.lite)

Deliverables

Please perform the testing session and write in a **markdown** file:

- Exploratory testing charters to document your testing.
- Findings from your charters. Did everything work as expected? What bugs were discovered?
- Prioritisation of those charters with reason for given priority.
- What kind of risks do you need to mitigate for this type of application?

You should not spend more than 2 hours on both testing and documenting.

2 - Mobile App Test Automation

You are given a task of writing a set of automated tests for the Monefy Android application used in Task 1.

Deliverables

Choose the 3 most important E2E user flows based on your exploratory testing session and provide:

- A test automation framework that performs the tests covering chosen flows:
 - The solution has to be easy to scale and maintain. Please explain how you achieved this.
- An explanation of the provided solution inside the README file consisting of:
 - the setup
 - how to run tests
 - your approach and why you chose that particular tech stack
- A test execution report attached in the repository.
- Bonus points for dockerising the solution.

Feel free to use whatever frameworks, library, packages you like and any structure or pattern you are comfortable with.

3 - RESTful API Test Automation

You are given a task of testing a RESTful API and writing a set of automated tests for the service. Find the service and set up your environment as explained in this [guide](#).

Deliverables

- A test automation framework that provides tests for CRUD operations on PET, STORE or USER endpoints (choose one domain):
 - The solution has be easy to scale and maintain.
- An explanation of the provided solution inside the README file consisting of:
 - the setup
 - how to run tests
 - your approach and why you chose that particular tech stack
- A test execution report attached in the repository.
- Bonus points for dockerising the solution.

Feel free to use whatever frameworks, library, packages you like and any structure or pattern you are comfortable with.

Note: Make sure your test automation solution is **NOT a Postman suite** but a framework built using best practices in coding and framework design.

Evaluation

We will evaluate:

- How you approach and organise an exploratory testing session.
- Your attention to detail.
- Your testing skills.
- How you prioritise your work.
- Your problem solving skills.
- Your coding skills and usage of good programming patterns.
- Your knowledge of the selected tech stack and libraries.
- Your communication and documentation skills.

Tech at N26

N26 is completely hosted in the cloud. Our tech stack includes Kotlin, Java, Docker, GitHub Actions, Kubernetes, ArgoCd, Consul, Vault, Saltstack, Terraform, Spring boot, Kafka, AWS, Appium, RestAssured. These tools and technologies are fully embraced by our test frameworks. Bonus points if you recognise most of them.

Let us know if you have any questions.