

SCALE FOR PROJECT TOKENIZER (/PROJECTS/TOKENIZER)

You should evaluate 1 student in this team



Git repository

git@vogsphere-v2.1337.ma:vogsphere/intra-uuid-5d963ee8-3b5a-4995-814



Introduction

- Remain polite, courteous, respectful and constructive throughout the evaluation process. The well-being of the community depends on it.

- Identify with the person (or the group) evaluated the eventual dysfunctions of the work. Take the time to discuss and debate the problems you have identified.

- You must consider that there might be some difference in how your peers might have understood the project's instructions and the scope of its functionalities. Always keep an open mind and grade him/her as honestly as possible. The pedagogy is valid only and only if peer-evaluation is conducted seriously.

Guidelines

- Only grade the work that is in the student or group's GiT repository.

- Double-check that the GiT repository belongs to the student or the group. Ensure that the work is for the relevant project and also check that "git clone" is used in an empty folder.

- Check carefully that no malicious aliases was used to fool you and make you evaluate something other than the content of the official repository.

- To avoid any surprises, carefully check that both the evaluating and the evaluated students have reviewed the possible scripts used to facilitate the grading.

- If the evaluating student has not completed that particular project yet, it is mandatory for this student to read the entire subject prior to starting the defence.

- Use the flags available on this scale to signal an empty repository, non-functioning program, a norm error, cheating etc. In these cases, the grading is over and the final grade is 0 (or -42 in case of cheating). However, with the exception of cheating, you are encouraged to continue to discuss your work (even if you have not finished it) in order to identify any issues that may have caused this failure and avoid repeating the same mistake in the future.

- Remember that for the duration of the defence, no segfault, no other unexpected, premature, uncontrolled or unexpected termination of the program, else the final grade is 0. Use the appropriate flag.

You should never have to edit any file except the configuration file if it exists. If you want to edit a file, take the time to explicit the reasons with the evaluated student and make sure both of you are okay with this.

- You must also verify the absence of memory leaks. Any memory allocated on the heap must

be properly freed before the end of execution.

You are allowed to use any of the different tools available on the computer, such as leaks, valgrind, or e_fence. In case of memory leaks, tick the appropriate flag.

Attachments

 subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/95346/en.subject.pdf>)

Preliminaries

If cheating is suspected, the evaluation stops here. Use the "Cheat" flag to report it. Take this decision calmly, wisely, and please, use this button with caution.

Preliminary tests

- There is in the repository:
 - Check that the token name contains 42.
 - A README.md file in the root.
 - A non-empty folder with the name "code".
 - A non-empty folder with the name "documentation".
 - A last non-empty folder used for the deployment part.
 - No video should be available on the repository. One link is allowed.
 - If no work has been submitted (or wrong files, wrong directory, or wrong filenames), the grade is 0, and the evaluation process ends.

 Yes

 No

Mandatory Part

Readme.md

Inspect the contents of the readme.md file.

- This is a time to discuss the different platforms that can be used and so the person being evaluated must clearly indicate the reason for his choice in this file.
- There is an indication on the choice of the language used.
- A short description to help understand the contents of the repository is not mandatory but highly recommended.

 Yes

 No

Documentation

Inspect the contents of the document folder.

- You should be able to understand how the project works by using this documentation. You should be able to understand the actions possible with the created token but also the way to deploy this token if needed.
- A demonstration video is possible to explain how this token works.

 Yes

 No

Code review

Inspect the contents of the code folder.

- You have to find other files to create a token. You must be careful to understand the logic of the code with the comments as well as the choice of variable/function names.
- This is a moment of exchange to understand how the token works. We are aware that the language will not necessarily be understood by the evaluator but it is a good moment to try to understand with the help of the evaluator!
- You must evaluate the quality of the code as well as the explanations of the person being evaluated.
- Ask for an explanation of the security used in this token. The evaluator should be able to explain how ownership works and the privileges used.
- Be careful to find the functions indicated in the documentation.

 Yes

 No

Deployment review

Inspect the contents of the last folder.

- You must find tools that allow the deployment of this new token. It is of course forbidden to have an API key or a password in clear text used in this file. If you encounter a password/Api key you can use the appropriate flag and the evaluation stop.
- A simple explanation of the evaluated should help you understand how this part works. The documentation should also help you if needed.
- With the help of the person being evaluated, you must deploy this token.

 Yes

 No

Blockchain explorer

With the help of the evaluated person you must find the address allowing to reach the explorer used to set up the token. You should see a smart contract address but also the network used.

A ticker must be used and directly visible in the explorer.

Check that the token name contains 42.

The person being evaluated must be able to explain how the explorer works as well as an explanation of the possible actions.

With the help of the person being evaluated, you must perform actions with the token.

The goal is to verify that the token is working properly.

Additional contracts can be added if necessary. Ensure that the token is used according to the documentation.

 Yes

 No

Bonus

Evaluate the bonus part if, and only if, the mandatory part has been entirely and perfectly done, and the error management handles unexpected or bad usage. In case all the mandatory points were not passed during the defense, bonus points must be totally ignored.

MultiSig

Is a multisignature system available in the code and on the token that is visible to the explorer?

You should see a section in the documentation explaining how it works.

A brief explanation of this system is then requested from the person being evaluated.

 Yes

 No

Ratings


Don't forget to check the flag corresponding to the defense

 Ok

 Outstanding project

 Empty work


 Incomplete work

 Invalid compilation

 Cheat

 Crash

 Incomplete group

 Concerning situation

Conclusion

Leave a comment on this evaluation

Finish evaluation