



HENNGE

CHALLENGE DETAILS

Introduction

Thank you for your application to HENNGE.

This challenge is intended for **frontend position** applicants.

Description

Suppose we have a system which archives emails of an organization. In the event of security incidents (such as leakage of sensitive information via emails), this system will be used by the organization's auditing department to retrieve and verify the emails of the concerning parties through its admin console. We would like you to design the user interface (UI) mockup for this system.

Mission 1: Design the UI using the given images

Assignment: Inside `frontend_resources.zip` (Please refer to resources section below), you will find a UI design file created by our designer (`result.pdf`) and accompanying images. Create the UI mockup using HTML, CSS, Javascript, etc. based on the design. You can use any framework or just vanilla JS to build this mockup.

Resources

- [frontend_resources.zip](#): UI design resources required for the challenge.

Mission 2: Extend the UI in order to implement a new feature

Assignment: In the search result section, extend the UI mockup so that user can inspect the body of each emails from the search result.

Mission 3: Indicate UI parts that can be improved

Assignment: From the provided design, state your opinion on UI parts that can be improved, including the reason why and how you would improve them.

Rules:

- Please use the images that we provide for the assignments.
- Please produce high-fidelity UI mockup for the assignments.
- Any fixed sample texts can be used as the data within the UI.
- For Mission 2: The extended design should allow user to inspect multiple email bodies at once.

What to include in the solution

- The final UI mockup.
- An `IMPROVEMENTS.md` text file with the list of your suggested improvements, using Markdown format.
- A `REPORT.md` text file with a report describing your efforts in the process of completing the missions above.
- Push all your solution files to a Github **private** repository, and then install our Github App (**PENDING**) permissions to read it.

Mission 4: Submit your solution

Upload your solution to Github

For the solution submission, please create a private Github repository and commit your solution to it. We ask you to include only the required files in it.

Give us permissions on your solution

For us to be able to get your solution, we need you to give us reading permission on your private repository. To do so, we need you to install our [HENNGE Admission Challenge Github App](#) in your private solution repository.

1. Enter to our Github App page: [Click here](#).
2. Click "Install".
3. Go to your user, and look for your repository.
4. Accept and install.

Note: Please do not publish your work at any public-facing URL.

Build your solution request

First, construct a JSON string like below:

```
{
  "github_url": "https://github.com/YOUR_ACCOUNT/GITHUB_REPOSITORY",
  "contact_email": "YOUR_EMAIL"
}
```

Fill in your email address for `YOUR_EMAIL` , and the private Github repository with your solution in `YOUR_ACCOUNT/GITHUB_REPOSITORY` . Be sure you have double-checked your email address; we will contact you by email.

Then, make an HTTP POST request to the following URL with the JSON string as the body part.

`https://api.challenge.henнге.com/challenges/003`

Content type

The `Content-Type:` of the request must be `application/json` .

Authorization

The URL is protected by HTTP Basic Authentication, which is explained on Chapter 2 of RFC2617, so you have to provide an `Authorization:` header field in your POST request

- For the `userid` of HTTP Basic Authentication, use the same email address you put in the JSON string.
- For the `password` , provide a 10-digit time-based one time password conforming to RFC6238 TOTP.

Authorization password

For generating the TOTP password, you will need to use the following setup:

- You have to read RFC6238 (and the errata too!) and get a correct one time password by yourself.
- TOTP's Time Step X is 30 seconds. T_0 is 0.
- Use HMAC-SHA-512 for the hash function, instead of the default HMAC-SHA-1 .
- Token shared secret is the `userid` followed by ASCII string value `" HENNGECHALLENGE003 "` (not including double quotations).

Shared secret examples

- For example, if the userid is "[ninja@example.com](#)", the token shared secret is "[ninja@example.comHENNGECHALLENGE003](#)".
- For example, if the userid is "[nijasamuraisumotorishogun@example.com](#)", the token shared secret is "[nijasamuraisumotorishogun@example.comHENNGECHALLENGE003](#)"

If your POST request succeeds, the server returns HTTP status code 200 .

Rules

- You do not have to disclose how you achieved this mission at this time. Do not hesitate to use source codes or tools on the net, but do the exploring process by yourself of course, do not ask your friend to help you. The only thing that matters is that it works!
- No bruteforce attacks, please!

Sample Request

```
POST /challenges/003 HTTP/1.1
Authorization: Basic bmluamFAZXhhbXBsZS5jb206MTU5NTk0MjU2MA==
Host: api.challenge.henнге.com
Accept: */*
Content-Type: application/json
Content-Length: 98

{"contact_email":"ninja@example.com", "github_url":"https://github.com/henнге/фронте
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Date: Wed, 26 Jun 2019 02:15:16 GMT
Transfer-Encoding: chunked

{"message":"Congratulations! You have achieved mission 3"}
```

Thank you for your challenge!

After successful completion of Mission 4, you will receive an email with a link to submit your CV and a Cover Letter. Once we receive them, our engineers will review your work and we will contact you via email with the result. Quick reminder: no deadline. Please note that our team will not answer to any inquiry related to the challenge. If you have any (other) questions, please reach us through recruit-engineer@henнге.com .