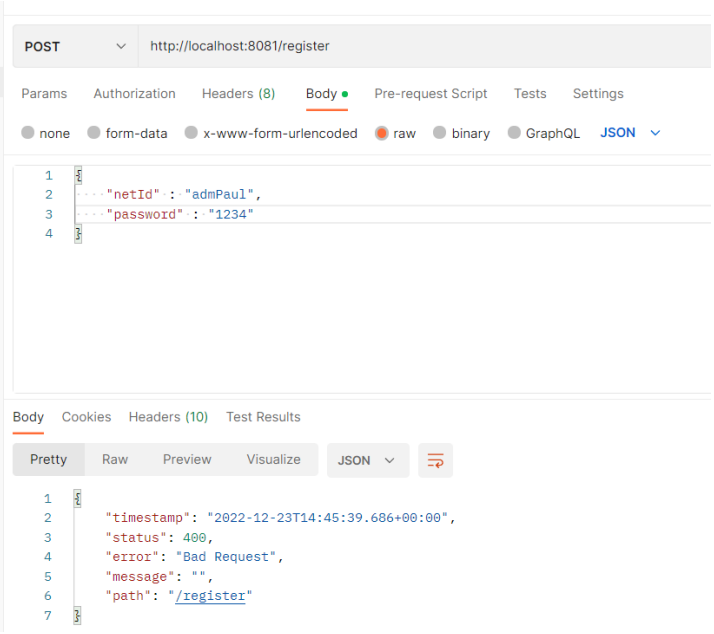
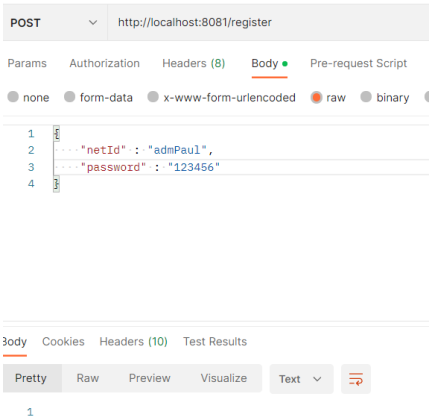
MANUAL TESTING MANUAL

1. Paul

- Paul is a sysAdmin who is setting up the system. The first thing he does then is to register and authenticate himself as an admin. For that, he tries to send a request to the ‘/register’ path in the authentication microservice, with his netId starting in ‘adm’.

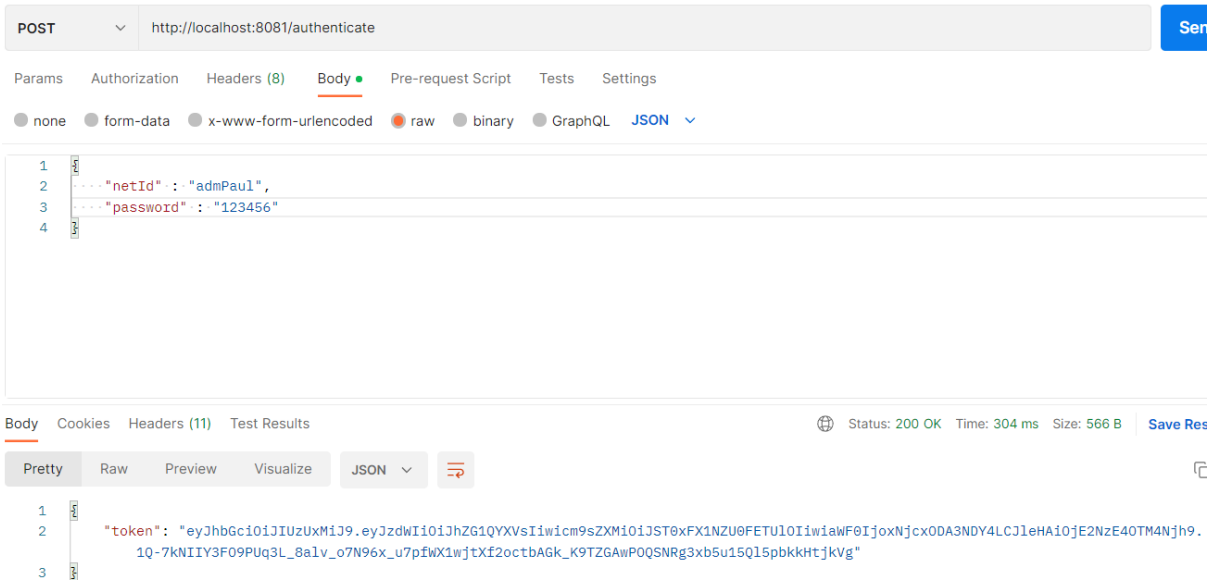


- On the first try of registering himself, a bad request is thrown. What happened here is that the password is too short. He tries again with a longer password, and succeeds in registering himself.



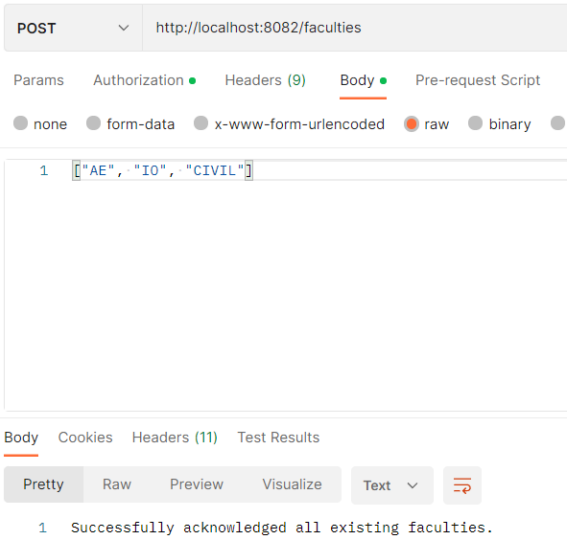
- Now, Paul needs to authenticate himself, and he does that by sending a request to

the ‘/authenticate’ path, with the same credentials as he did for the registration:

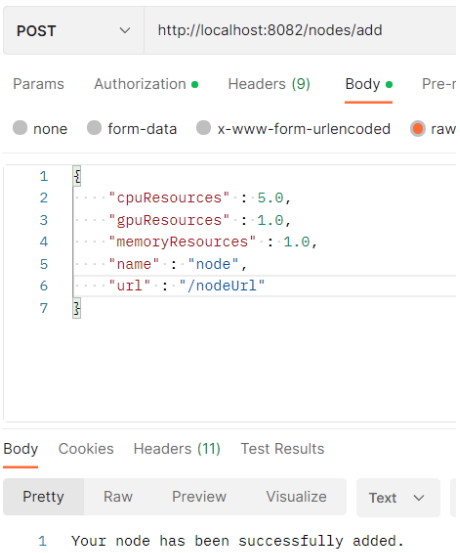


- By doing that, Paul received the token that he should use to access the system as a sysAdmin. This token should be used in all the requests made to the system, as a bearer authentication token.

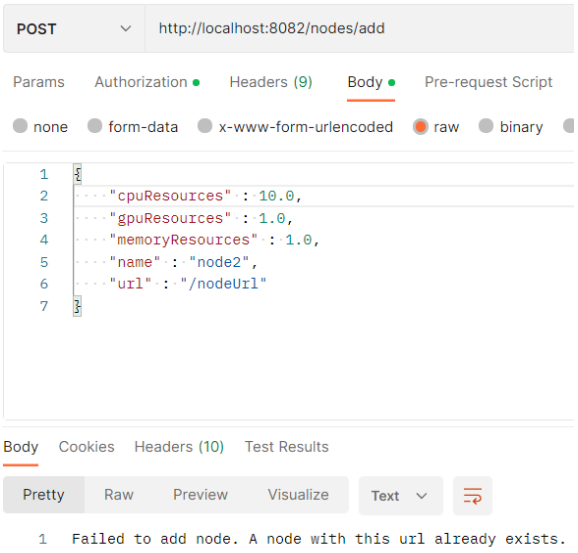
- Since Paul does not trust the user to add the existing faculties to the system, Paul decides to do it himself.



- Now that the faculties were added, Paul proceeds to add some nodes to the cluster.

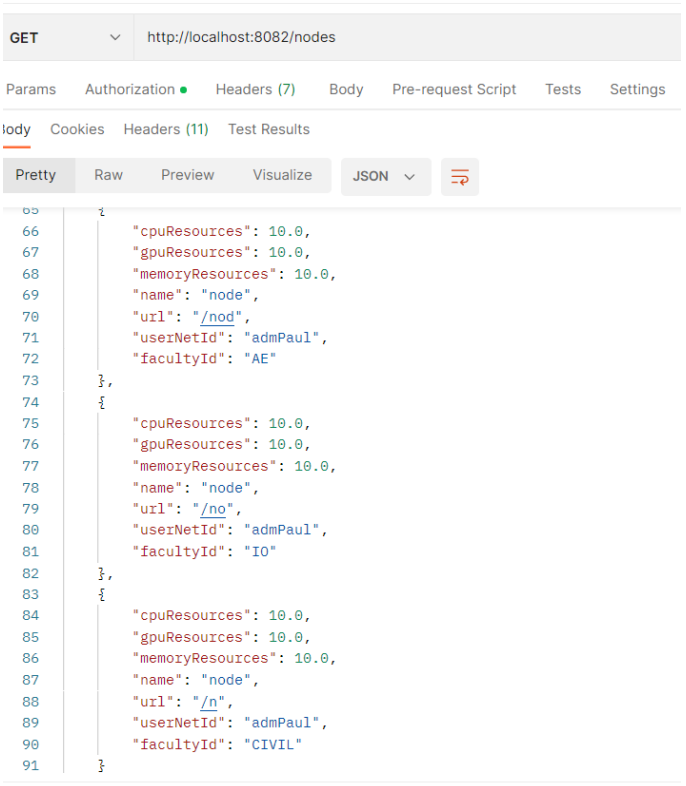


- He tries to add another node, but this time receives a warning:

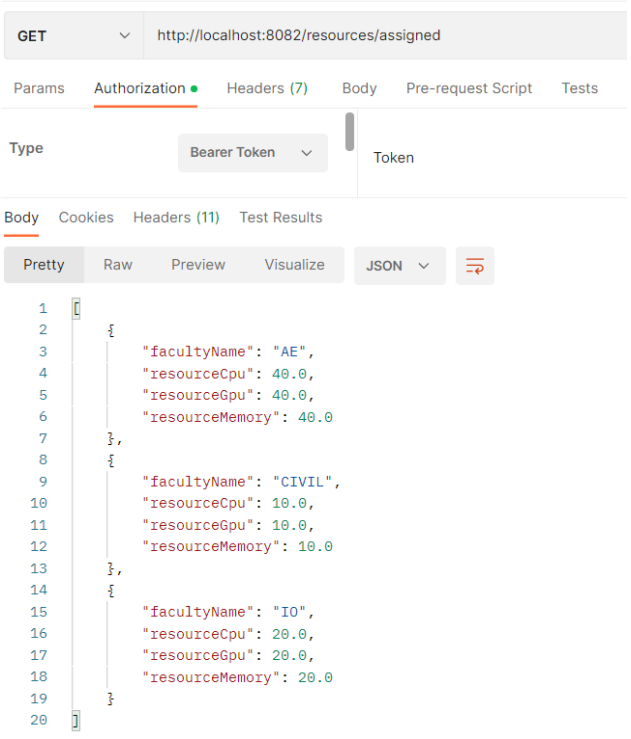


    - What happens is that the system already contains a node with this url, and therefore, cannot add another one with the same url. This is because the url is the primary key of the node.

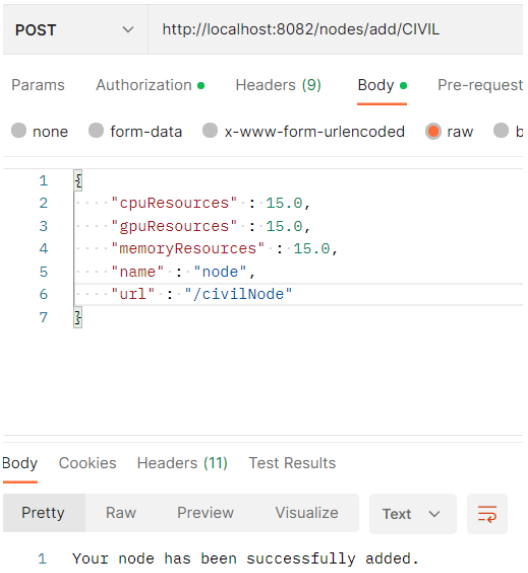
- Paul then proceeds to add more nodes to the cluster. He now wants to review all the nodes that the cluster currently contains, and sends a request to the ‘/node’ path to retrieve that:



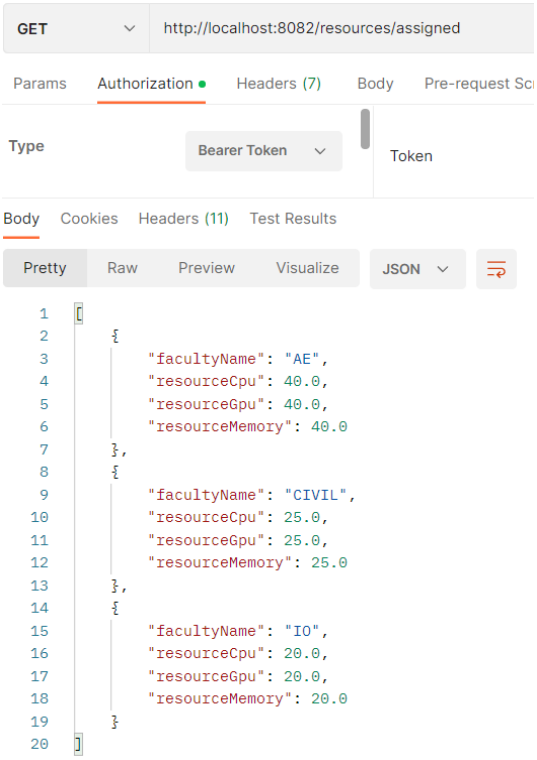
- Now, Paul wants to check the amount of resources per faculty. In order to do that, he sends a request to the '/resources/assigned':



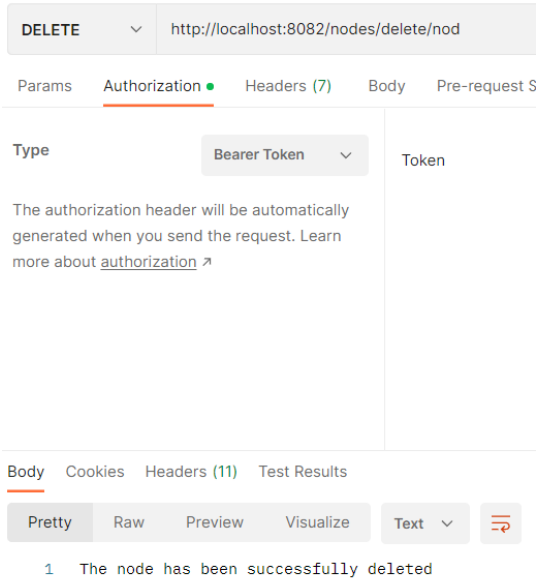
- Seeing this, Paul decides to add more resources to the CIVIL faculty, and does that through the '/nodes/add' url again, adding the facultyId to the url:



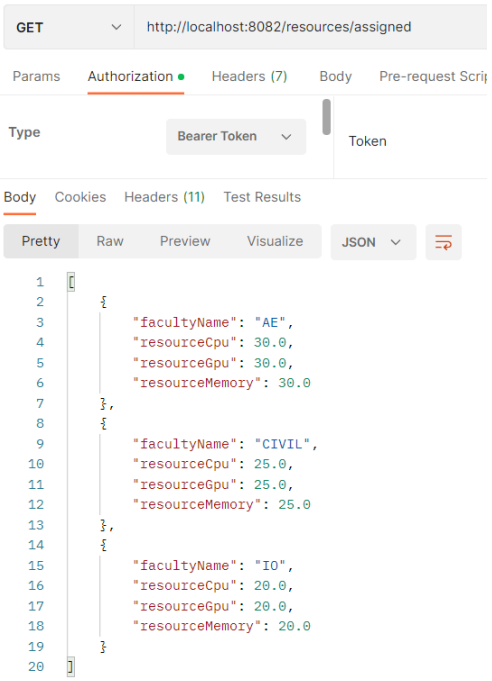
- To make sure that the node was correctly added to the ‘CIVIL’ faculty, Paul retrieves the resources assigned per faculty again.



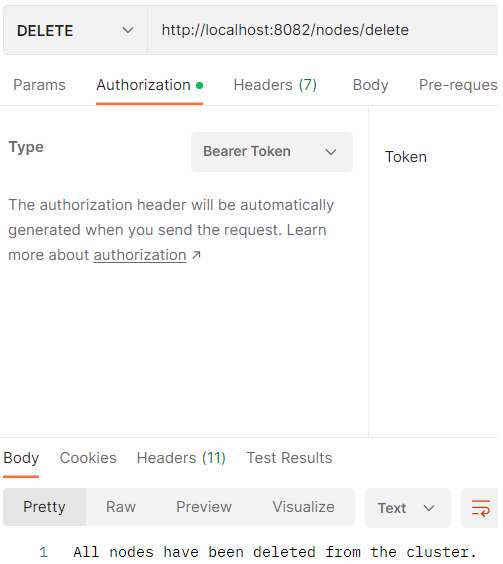
- Furthermore, Paul decides to remove one of the nodes assigned to the AE faculty.
  - First, he checks all the nodes in the cluster again, to see which ones are assigned to AE. He finds out that the one with the url ‘/nod’ is one of those, and chooses to remove that one.



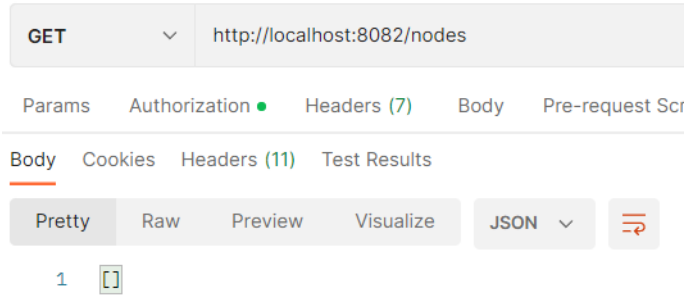
- Now he checks again the amount of resources per faculty, to make sure the deletion was successful and correct.



- Finally, Paul decides to remove all the nodes existing in the cluster by passing ‘nodes/delete’ as the url, without specifying a node url.

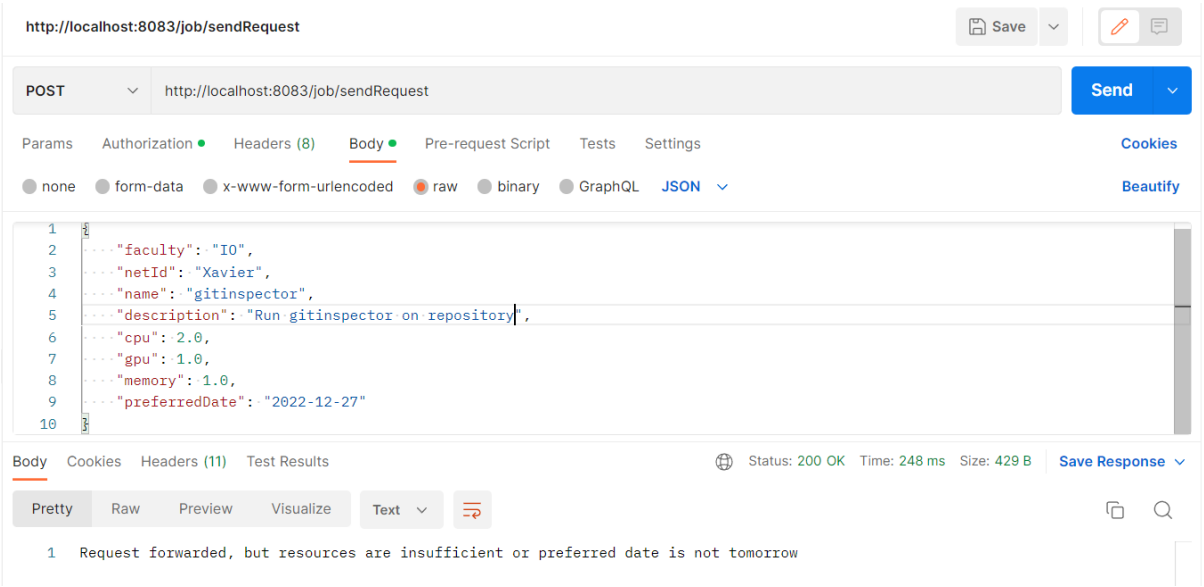


- And checks if the deletion of all nodes was indeed successful.

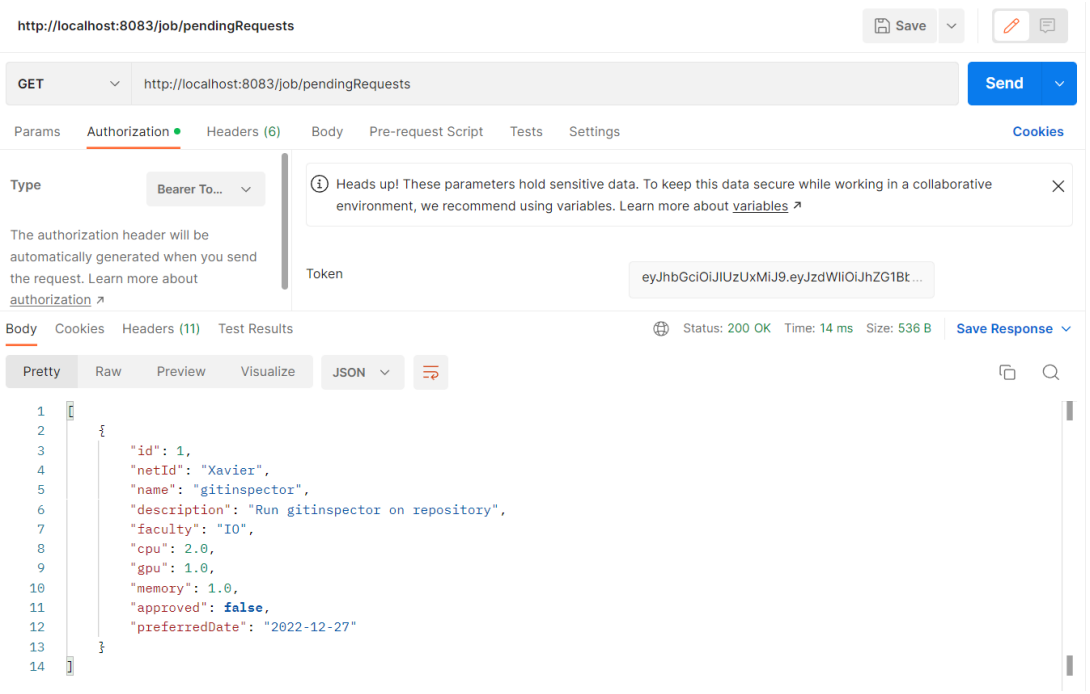


2. Xavier

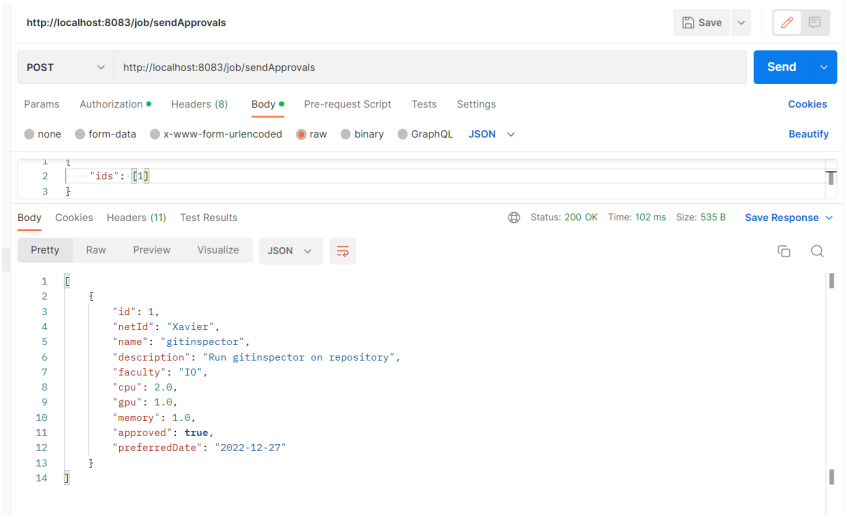
- Xavier is a PhD student at the faculty of Industrial Design. In his free time, he enjoys studying and submitting requests to Delft Blue the most; this is, coincidentally, exactly what he is doing now. He starts off by making a single request for a gitinspector script to be ran on his repository:



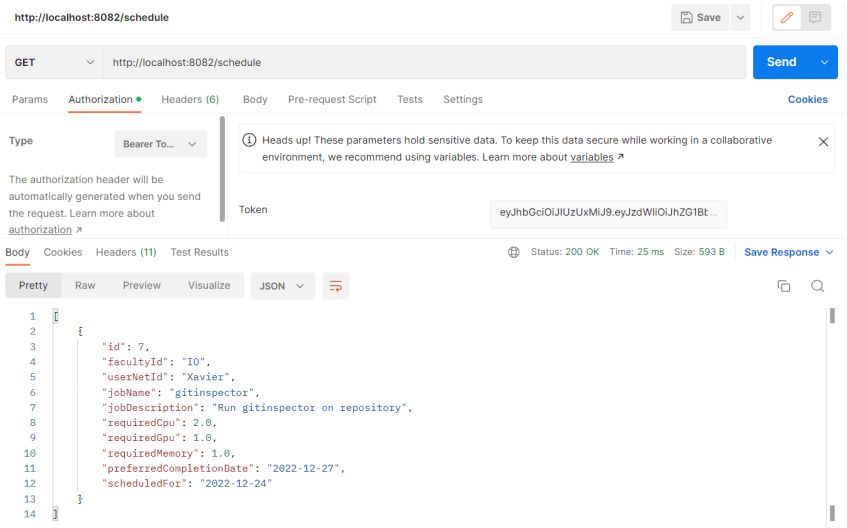
- The somewhat cryptic message doesn't phase him: he's been making so many requests that he knows by now that it simply indicates that his request still needs to be approved, despite the time being well after 6 pm. This is because his preferred completion date isn't tomorrow.
- A passing administrator assigned to IO notices Xavier's first request before he manages to flood the microservice with more.



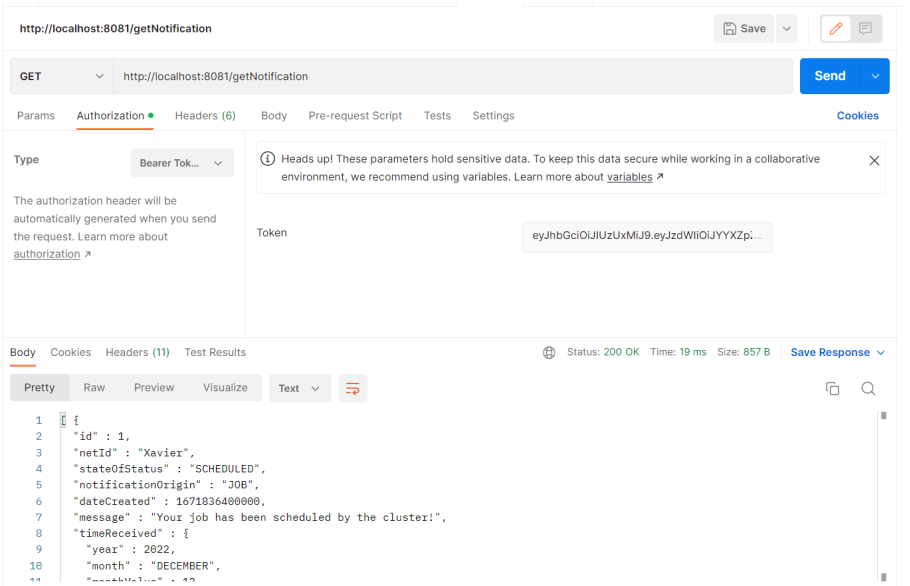
- Knowing Xavier and how much work he has, he chooses to approve it:



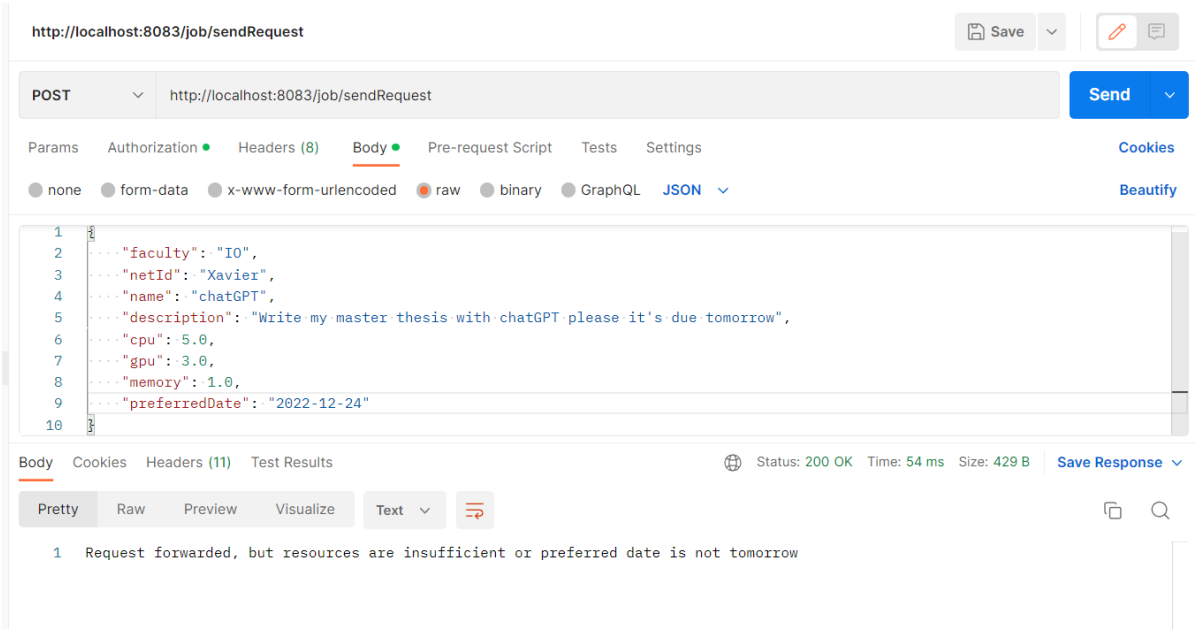
- To make sure that everything did work fine after recent code disturbances due to the now-infamous Mallory's actions, the administrator checks the cluster's schedule.



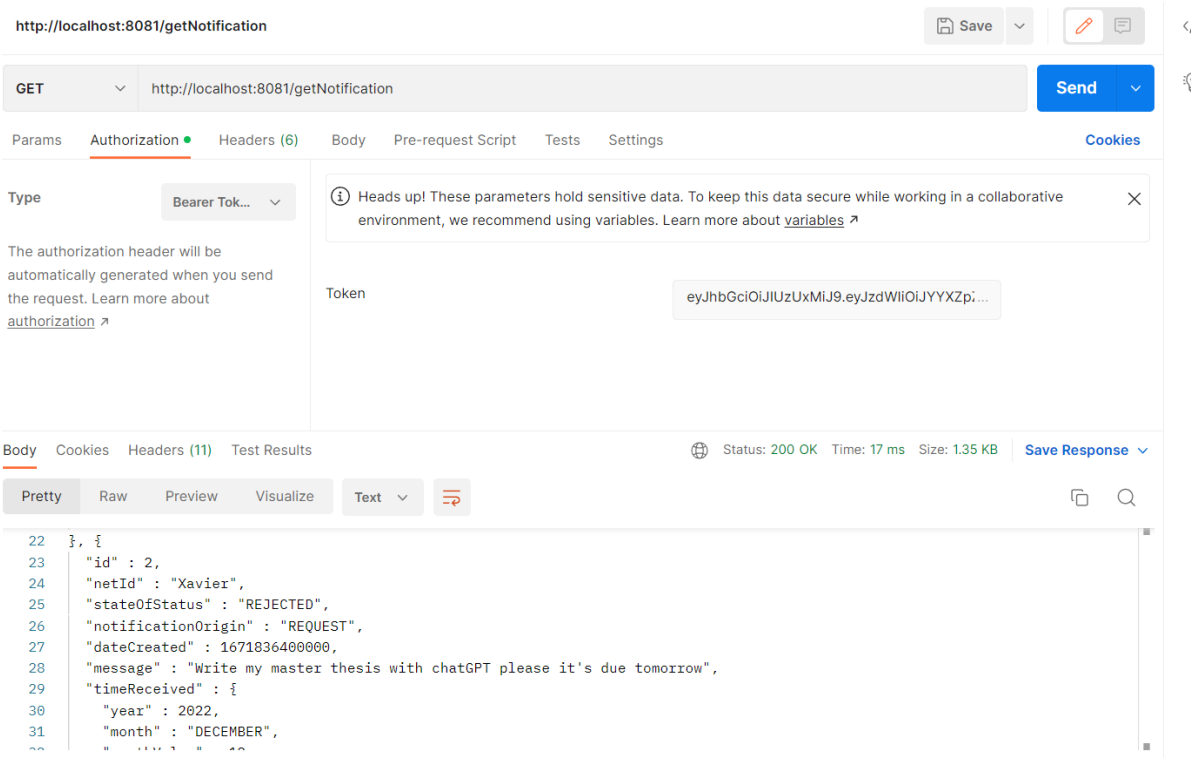
- Xavier himself, meanwhile, checks his notification folder to see if his request has been approved. It has!



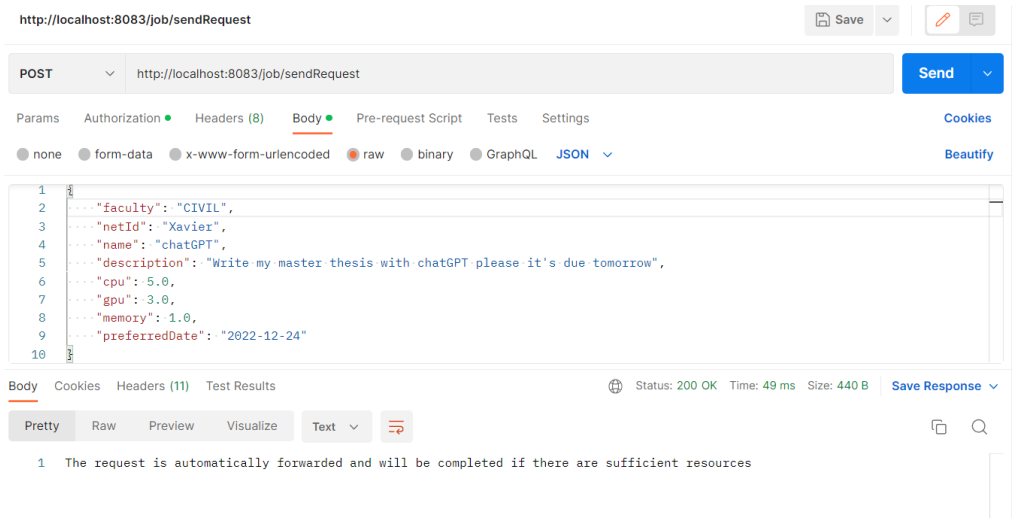
- Xavier continues to post more and more requests, including one for tomorrow. As it's past 6 pm, he counts on it being approved automatically.



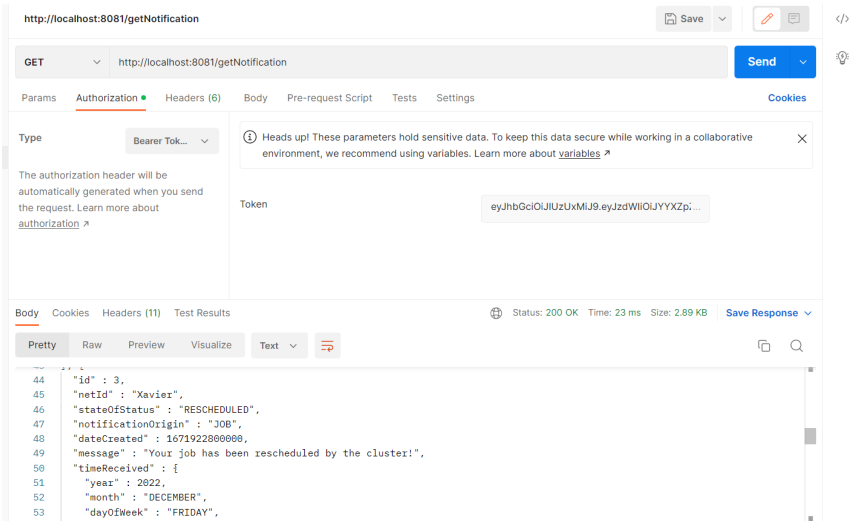
- Unfortunately, there are not enough resources for IO for tomorrow available, as the notifications quickly show



- This is not going to Xavier's plan. Fortunately, he has an ace up his sleeve: he has secretly maintained his link to the Civil Engineering faculty from the time of his minor! Therefore, he decides to make the request again, through CIVIL:

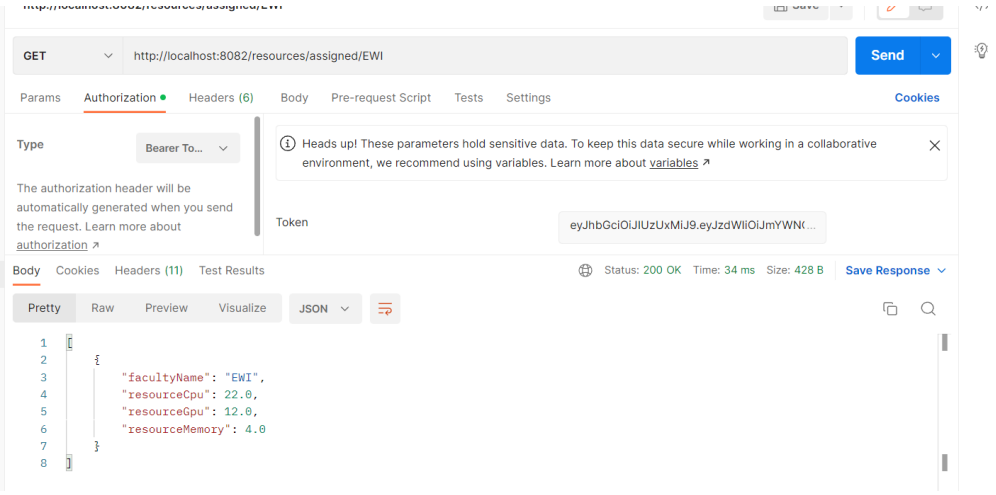


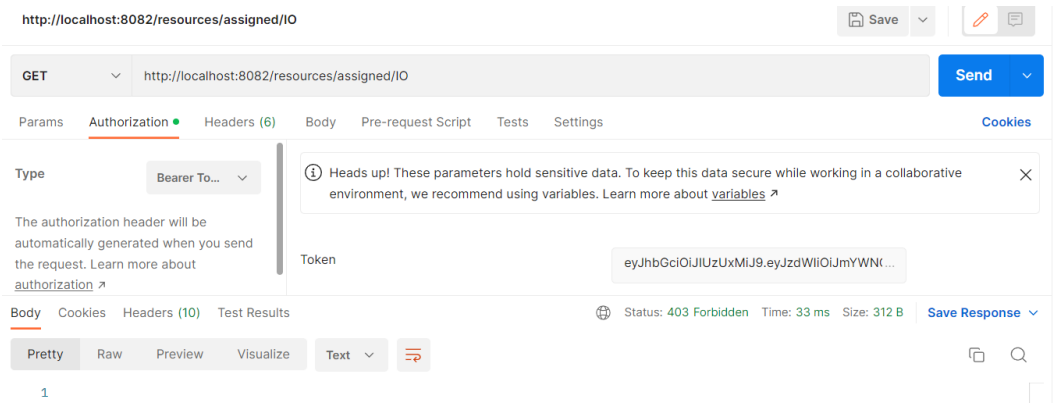
- This is a message Xavier recognizes and that restores his hopes in his ability to finish his master thesis. The story of how he became a PhD student before getting his Master's degree is one for another day and another microservice project.
- Xavier wakes up and decides to check notifications once more. To his shock, he realizes that Delft Blue was attacked overnight, and all nodes belonging to CIVIL deleted one by one.



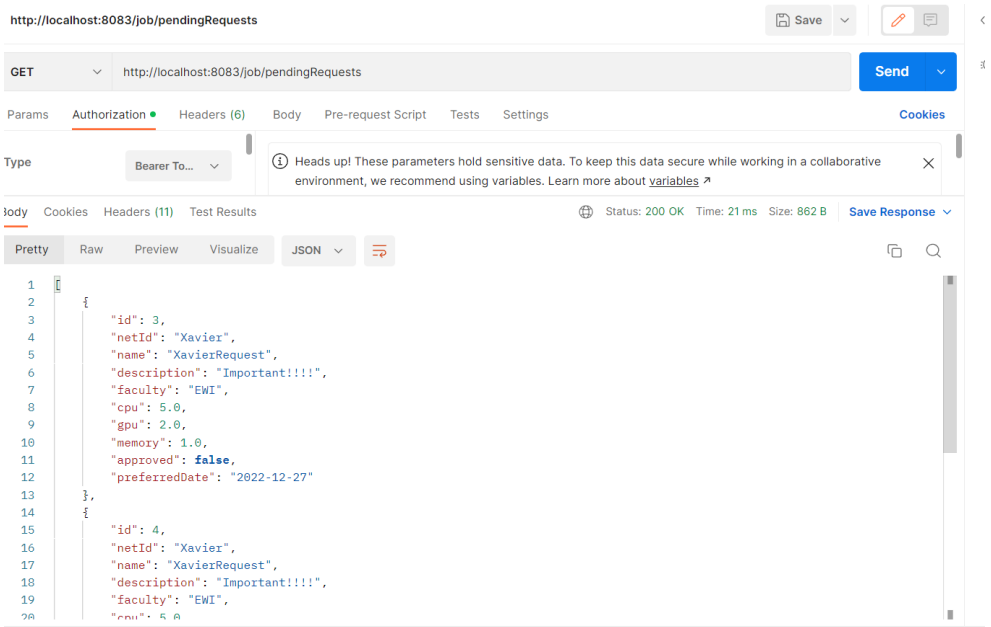
3. Prof. Nguyen

- Professor Nguyen is a kind-hearted faculty member of EWI. As usual this time of day and year, he opens the Delft Blue system up to inspect resources available to his faculty, as well as approve some student requests.

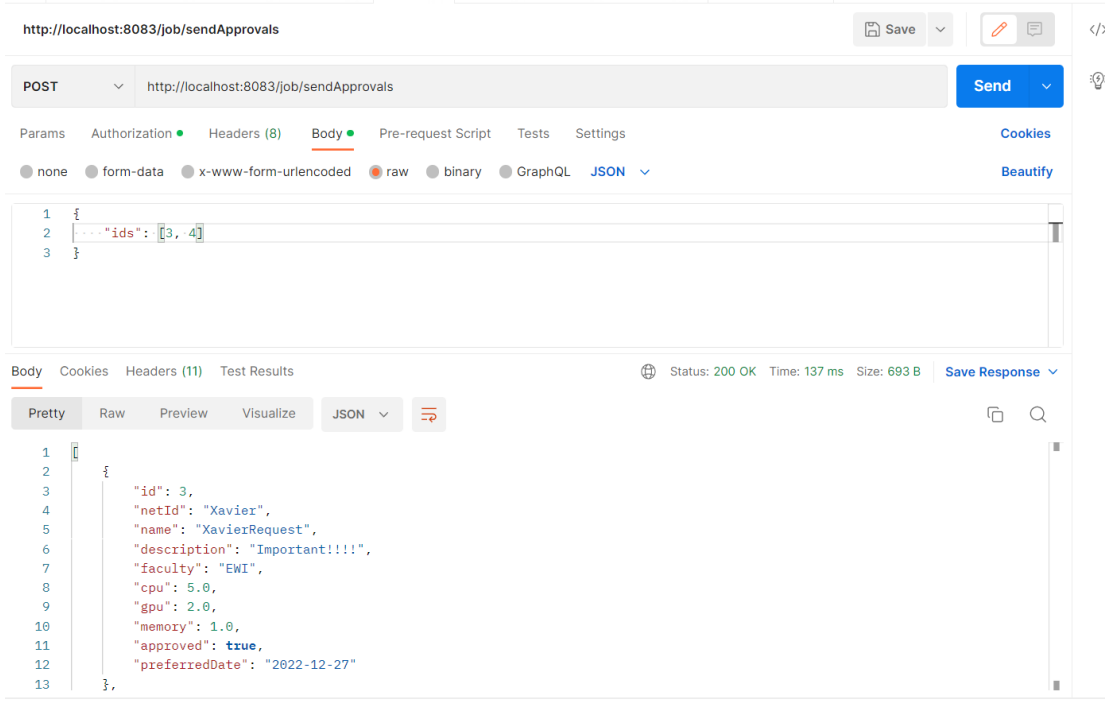


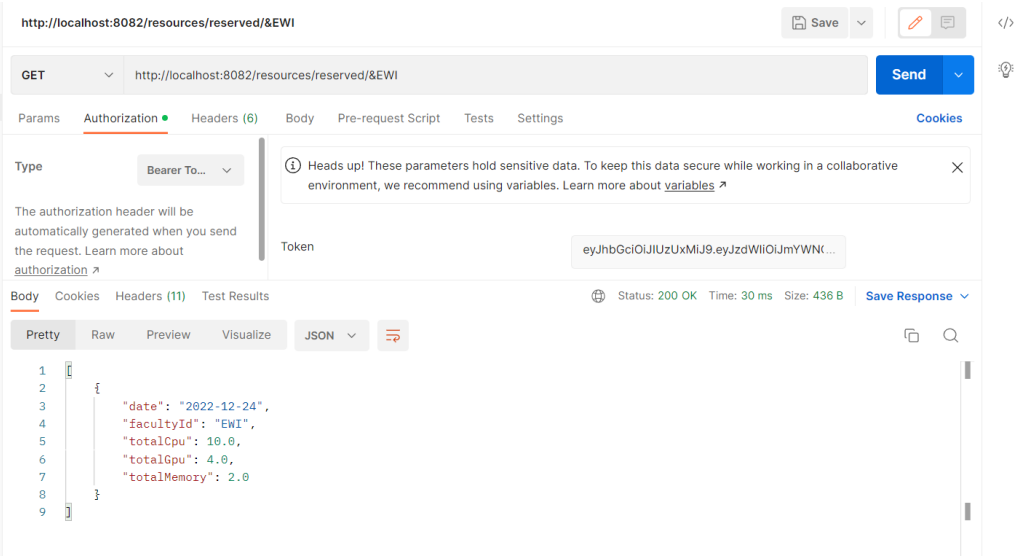


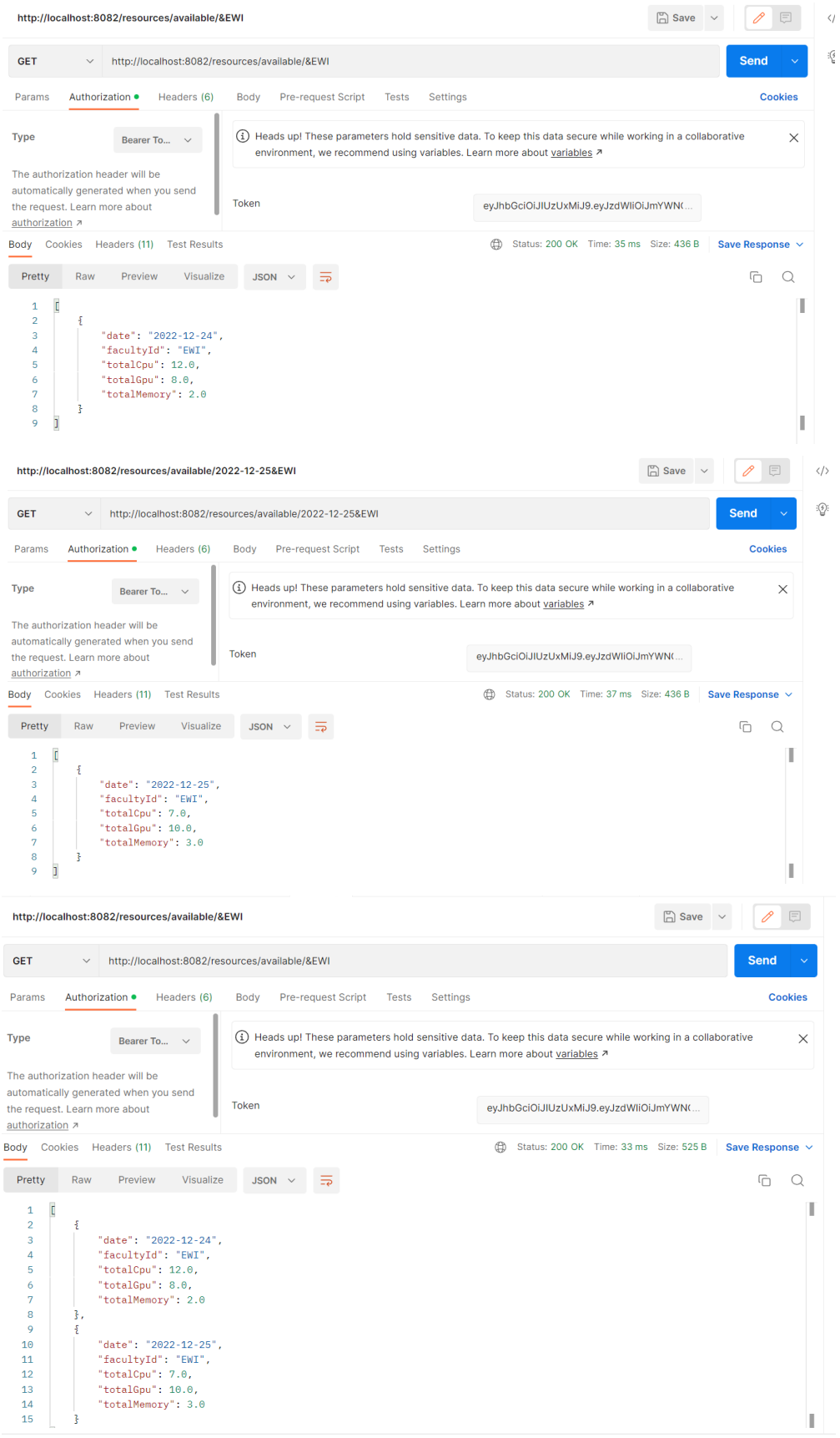
- Unfortunately, the professor can only see resources of his own faculty. If only he were an administrator…
- Seeing that there is a lot of resources assigned and few reserved, prof. Nguyen checks whether he has any job requests to approve:



- There’s quite a few jobs that the faculty member can choose from. Knowing Xavier well (though wondering why he is allowed to make requests to EWI, being an IO student…), prof. Nguyen approves his requests.



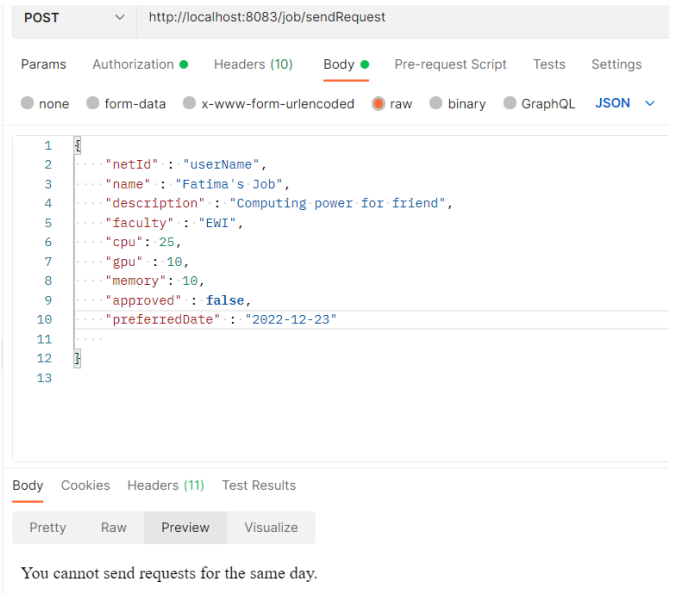




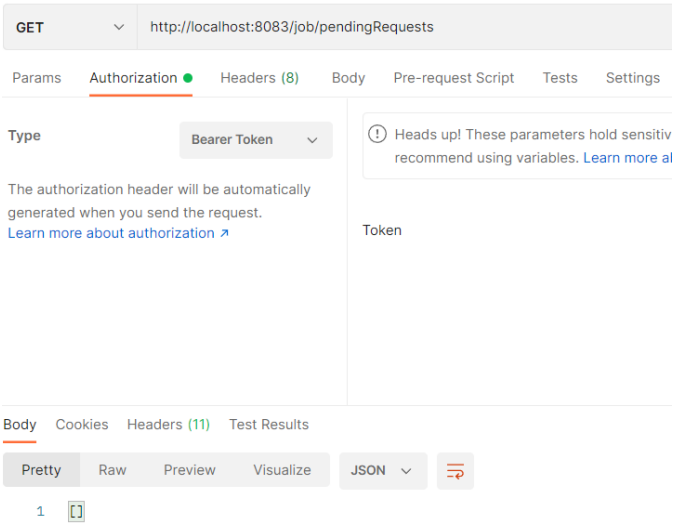
- Satisfied, prof. Nguyen closes Delft Blue.

4. Fatima

- Fatima, as a first time user of the platform, is quite unsure of how and where to start. She wants to use a part of the cluster's computing power for a friend who is not part of a faculty, so she is not sure which API's to use. She starts off by making a request to the Request-Microservice. She thinks it is nice to get the job as soon as possible, so she makes the preferred-Date in the request the same day as today.:

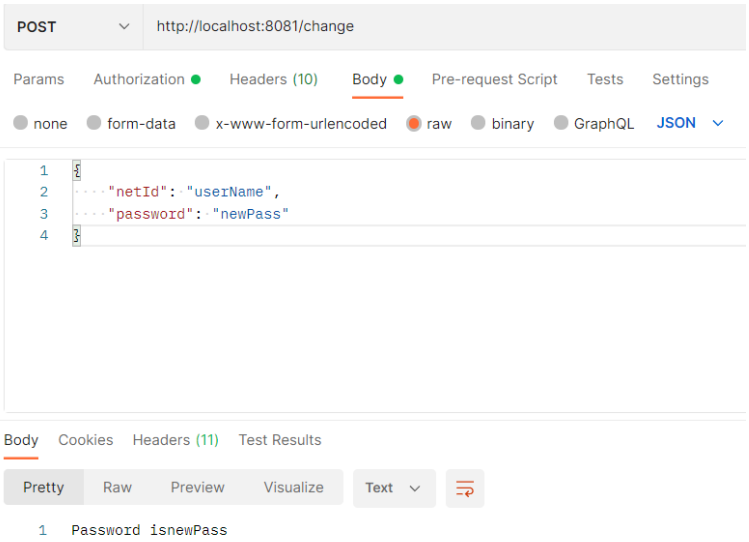


- She didn't read the response, and assumes the request has been made successfully. She checks the pending requests from EWI's faculty computers, and sees the following:



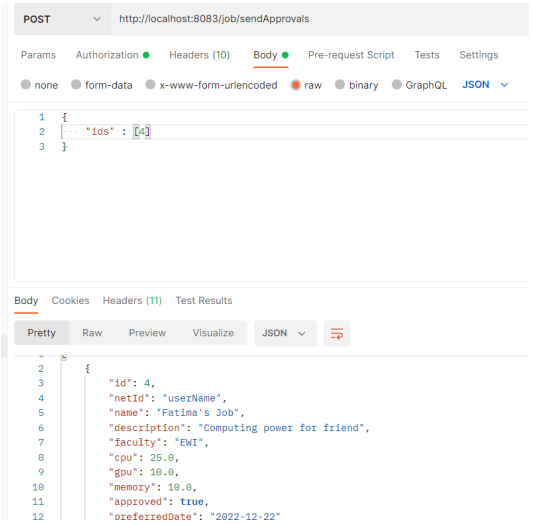
- She is confused as to why she is seeing an empty list. She sends the same request multiple times, in case there was an error in the system, but still her request is not appearing. She goes back and sends the original request multiple times, but to no availl.

- She thinks there is something wrong with her account, so she wants to change her netId password with the hope that it fixes it. She makes the following request:

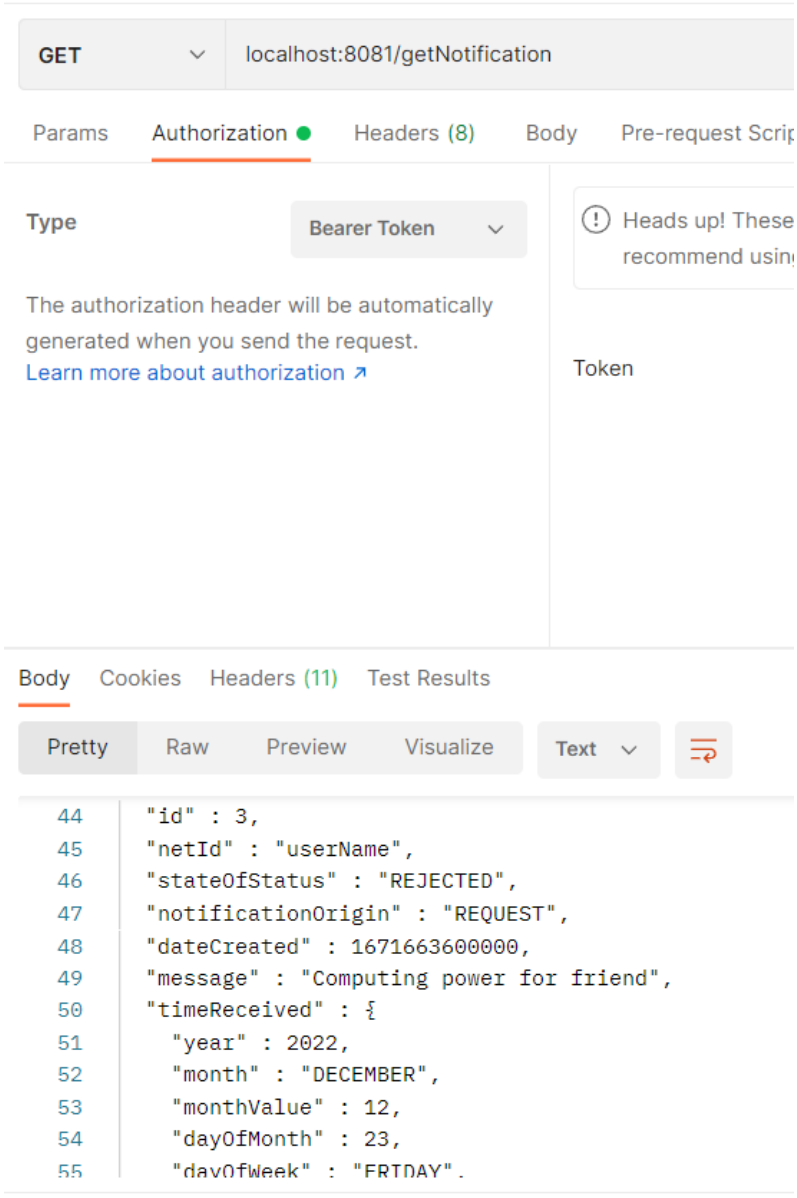


- She successfully changes her password, and tries to send the original request once again. However, since a new day (the 23rd of december) has arrived, the preferred completion date is now in the past, and the system accepts that as a valid date, and assumes there is no rush in the requested job. So she makes the request again and it passes:

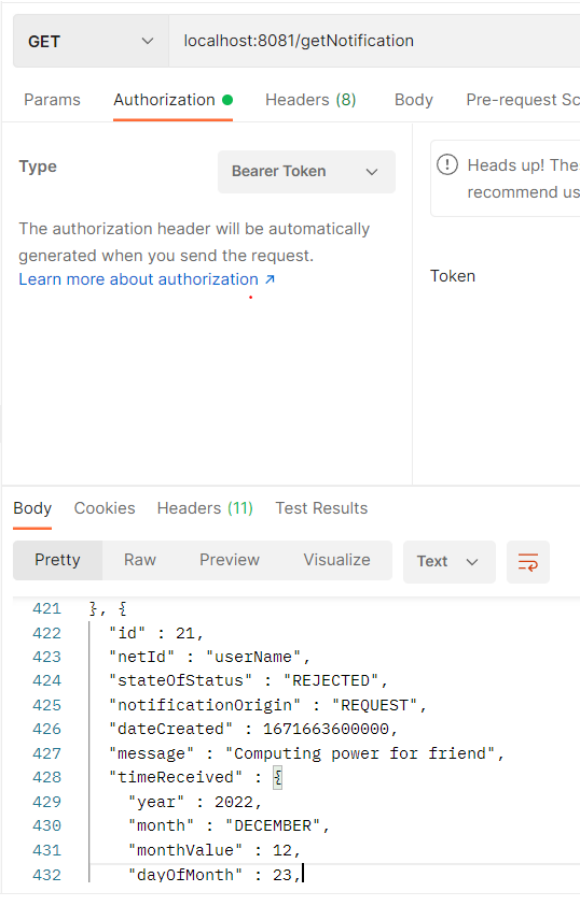
- Fatima now accepts the request from the faculty computer as follows, assuming it will be correctly processed for her friend:



- The next day she asks an administrator for the progress on her job. They answer that she can check herself by looking at her notifications. Fatima did not know how, and the admin tells her exactly how. She makes the following request:



- She suddenly sees her request had never been sent to the cluster! It turns out her request had been cancelled. Fatima did not know why, but thought that sending more requests could fix the problem, and that 1 of them was bound to go through. She spams her original request multiple times. She then looks at her notifications again.



- She looks at her notifications, and sees 21 failed requests. She gives up and tells her friend that she can’t do the job for her.