

# Instructions

This week will be more administration centric.

## Trivia

You'll have to add authentication, and more importantly user groups (companies separation). This simulates situation in which different platform clients are hosted on the same server, and their data need to be separated, so that they cannot receive data not owned by them. This approach will require authentication implementation, as well as specifying special "owner" fields in appropriate models. Moreover, we require that all EP be rewritten using django rest framework. Additionally, filters (from specified library) and pagination will be used. As you've already noticed robots sometimes send fault messages, which shouldn't have been handled by your contemporary code. This time, in order to have live feedback on such situations, you must implement websocket connection.

# Tasks

1. Add websocket (use library from links)
2. Fault log must be sent (propagated) using websocket
3. Rewrite all EP to rest framework adding EPs:
  1. EP detaching communication device from the robot
  2. EP attaching communication device to the robot
  3. EP detaching communication device from one robot and attaching the same device to the other
  4. Add model for a storage of robot modification history and write EP which will only allow modification of specified robot (any number of fields - update only the ones passed in request), then add entry to the robot modification history based on such modification
  5. EP for only creation of communication device (check existence of such robot in the database first, you can use unique on name, use validator)
  6. EP for only removing a robot, telemetry and location must remain in the database
  7. EP for only removing location for a specified robot on a specified day
  8. EP for setting specified temperature for a whole day on single robot as well as on a set of robots
  9. EP which will create company based on provided NIP using api (use requests library)
4. Add pagination on robots EP returning all robots
5. Add filters on EP getting specified robot for every field (contains)
6. Add filters on EP getting telemetry for timestamp field (lte, gte, range)
7. Implement user groups using build in Group model from django. Read about extending django models - one to one relation from your company model to the Group model. All EP must return data only for the currently logged in company.

8. Add authentication methods (token, session) from rest framework
9. (OPTIONAL) create report using xhtml2 for the specified robot including latest telemetry data and latest location data
10. (OPTIONAL) using signals send mqtt message to topic /debug/msg after the telemetry data got saved in the database (post save)

# useful links

1. <https://www.django-rest-framework.org/>
2. <https://channels.readthedocs.io/en/stable/>
3. <https://www.django-rest-framework.org/api-guide/authentication/>
4. <https://www.django-rest-framework.org/api-guide/pagination/>
5. <https://xhtml2pdf.readthedocs.io/en/latest/usage.html#using-xhtml2pdf-in-django>
6. <https://docs.djangoproject.com/en/4.0/ref/signals/>
7. <https://docs.djangoproject.com/en/4.0/topics/auth/default/>
8. <https://django-filter.readthedocs.io/en/stable/>
9. <https://www.gov.pl/web/kas/api-wykazu-podatnikow-vat>
10. <https://pypi.org/project/requests/>