**2e Systems d.o.o**

# BE developer candidate practical assignment

## Task summary

Develop a service that continuously loads and stores a subset of METAR data for the subscribed airports, and makes that data available to clients on request.

## Domain knowledge

To familiarise yourself with the domain, please consult the following resources:

- A description of METAR can be found on wikipedia: https://en.wikipedia.org/wiki/METAR
- ICAO airport codes description can also be found on wikipedia: https://en.wikipedia.org/wiki/ICAO_airport_code

## Required technologies

- The service must be implemented using Java 8 (or newer), and based on the Spring framework (preferably Spring Boot).
- Storing of METAR data must be done in a RDBMS, and managed using JPA.
- All components of the solution (the service, RDBMS, automated tasks) must run under the Linux operating system.
- The solution should be made available in a public source code repository (github, bitlab, bitbucket or similar)

## Task details

The solution consists of 3 components, described in the following sections.

### The service

The service is a Java application running continuously that provides, at a minimum, two RESTful endpoints, as follows:

- ***airport subscription/unsubscription and subscription listing endpoint*** provides support for POST, GET, and DELETE predicates; some possible examples are:

  "`POST /subscriptions`" with JSON payload '`{ "icaoCode": "LDZA" }`' for subscribing the Zagreb airport
  "`GET /subscriptions`" which would return all of the subscribed airports in JSON, e.g. '`[ {"icaoCode": "LDZA" } ]`'

"DELETE /subscriptions/LDZA" which would delete the subscription for Zagreb airport

- ***METAR data storage/retrieval service***, some possible examples are:

  "POST /airport/LDZA/METAR" with JSON payload '{ "data": "METAR LDZA 121200Z 0902MPS 090V150 2000 R04/P2000N R22/P2000N OVC050 0/M01 Q1020=" }' would store the data in the database
  "GET /airport/LDZA/METAR" would return the last METAR data record previously stored in JSON format

## Automated task

This consists of a job, scheduled to run in regular intervals, that retrieves a list of subscribed airports, queries an external service for METAR data, and sends the retrieved data to the METAR storage service.

The external service for METAR data retrieval to be used can be found at:
https://tgftp.nws.noaa.gov/data/observations/metar/stations/

The job should be implemented and scheduled from the Linux shell (command line), as a script or a list of commands.

## The database

The database contains, at a minimum, the following tables:

- the "subscriptions" table stores all of the subscribed airports
- the "metar" table stores METAR data for airports

# Extra tasks

Before implementing any of the following extras, please use the tagging functionality of your chosen source control system, and tag the basic version of the code that implements requirements specified above. After that, please also tag a new version after implementing any of the functionalities specified below:

- Extend the subscription endpoint to support activation/deactivation and the PUT predicate, e.g.
  "PUT /subscriptions/LDZA" with JSON payload '{ "active": "0" }' would deactivate the subscription for Zagreb airport
- Providing of filtering capabilities when returning airport subscriptions (so the client can request only active subscriptions, or only the airports matching certain letters)
- Parsing/splitting up of METAR data and storing each element in a separate database field (at a minimum: data timestamp, wind strength, temperature, overall visibility)
- Extending the METAR retrieval endpoint to enable retrieving only a subset of available data (e.g. only wind strength and temperature)
- Decoding of METAR data to natural language. Can be implemented before storing/persisting it to the database, or on-the-fly when returning it via the METAR retrieval endpoint