# Development Challenge - SDET

## Context

The FAA (Future Assets Analytics) team at Axpo uses a web API to retrieve weather data from AEMET (Spanish Meteorological Agency) to analyze the feasibility of innovative projects. This exercise focuses on automating the signup process and validating API functionalities to ensure the robustness of the system.

The AEMET provides a free-licensed Rest API to allow the dissemination and reuse of the Agency's meteorological and climatological information. The documentation of this API can be found on: https://opendata.aemet.es/centrodedescargas/inicio.

## Objective

You are required to create automated tests for the signup process and API key retrieval and validate the data retrieval process using the API.

## Part 1: Automate Signup Process

### Goal:

- Automate the end-to-end signup process.

### Steps:

- Automate navigating to the AEMET API portal and completing the signup process.
    - Be aware of a reCAPTCHA, maybe a manual input here?
- Retrieve the API key sent to the user's email.
- Parse the email to extract the API key.

### Deliverable:

- A script or suite of tests.
- Ensure proper exception handling for cases like invalid credentials or missing emails.
- Log the flow of the process for troubleshooting.

## Part 2: Validate API Key and Data Retrieval

### Goal:

- Validate that the retrieved API key works as expected.

### Steps:

- Use the API key to make a request to the AEMET API endpoint `/api/antartida/datos/fechaini/{fechaIniStr}/fechafin/{fechaFinStr}/estacion/{identificacion}`.
- Validate the response:
    - Check HTTP status codes.
    - Verify the structure and content of the response data (e.g., fields like temp, pres, vel).
    - Ensure that the datetime field in the response respects the CET/CEST timezone.

### Deliverable:

- Automated tests
- A test report summarizing the outcomes.
- Use parameterized tests to validate different combinations of input parameters (start date, end date, station).

## Requirements

1. The code should:
    - Be structured and modular.
    - Include clear logging and exception handling.
    - Use best practices for automation and testing.
2. Provide a Git repository with:
    - The test scripts.
    - A README file with instructions on how to run the tests.
3. Bonus:
    - Implement data validation to ensure that the temperature, pressure, and speed values meet realistic thresholds.
    - Evidence how you might handle the situation where the data in a public test environment is constantly changing. You do not need to evidence this in your code, some bullet points outlining your approach will suffice.

## Assessment Criteria

- Code quality and adherence to best practices.
- Coverage and effectiveness of the automated tests.
- Clarity of the documentation and instructions.
- Ability to handle edge cases and errors gracefully.