Lab 06 - Error handling

Script Languages

Learning goals

1. Use exceptions.
2. Store and retrieve data in JSON file.

Exercises

1. Preparation

In this lab you will develop two applications: first to create a configuration file for your second application.

1. Read about types of HTTP request methods:

   https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

2. Read about JSON format:

   https://en.wikipedia.org/wiki/JSON

2. Using JSON files

1. Write the application that asks a user for configuration parameters and store them in the JSON file. Set the appropriate encoding for the config file. Parameters to be stored:
   - name of the webserver log,
   - name of one of the HTTP request methods (to be used as a filter for displaying)
   - logging level used by the application,
   - number of log lines to be displayed at once,
   - your own parameter (devise a clever setting for your app).

3. Exception handling

1. Use code from previous labs. Modify it accordingly to read configuration settings from the JSON file created with application implemented in task 2.1. Loaded settings should affect the way the program works e.g. set logging level accordingly. Use exceptions to handle these situations:

    - configuration file does not exist,
    - configuration file is not a correct JSON file,
    - configuration file does not contain values your program needs,
    - logfile with the given name does not exist.

Use builtin exceptions (https://docs.python.org/3/library/exceptions.html) and implement custom exception for selected cases. Use `try/except/else/finally` statements.

2. Write a function to print all requests containing resource given as a parameter (e.g `index.html):`

    - `V`alid HTTP request header starts with the line containing request method and resource name eg.: `GET /index.html HTML1.1,`
    - Print only requests with method given in configuration,
    - Handle malformed requests using exceptions. Application should process only valid requests, and display a summary of invalid requests at the bottom of its output.
    - The resource name should be given be the user during runtime of the application by using `input()` method,
    - Stop every number of lines defined in configuration and wait for a user to press a key. What would you do if the number was less or equal to zero?

3. Write your own function processing web server log controlled by your parameter.