Lab 10 - Internet Access

Script Languages

Learning goals

1. Download content from Internet web sites.
2. Analyse downloaded web pages content.
3. Send emails.

Exercises

1. Preparation

In this lab you will create Python application to read the dataset in CSV format, aggregate data read and finally prepare a report as an Excel file.

1. Install and read about request module (https://docs.python-requests.org/)
2. Install and read about Beautiful Soup 4 module (beautiful-soup-4.readthedocs.io)
3. Read about builtin modules smtplib (https://docs.python.org/3/library/smtplib.html) and email (https://docs.python.org/3/library/email.html)

2. Sending emails

1. Develop a function to send an e-mail to a certain email address using the university mail server.
    1. Prepare a configuration file to store the required email settings (username, password, server address) and read it when your application starts. You should not store login and password in your code!
    2. Send the message when your app is started in the following manner:

       ```
       python app10.py --mail "Additional message to the someone”
            --topic "Email topic" --to "receiver@example.com"
       ```

    3. Include a current date and time in the body of the message.
    4. Use a tool of your choice to implement command line interface.

2. Interacting with a REST API

1. Read about the API of the Cat Fact (https://cat-fact.herokuapp.com/) service (https://alexwohlbruck.github.io/cat-facts/docs/)

2. Develop a function to read a number of facts about cats given by user as parameter to your application, e.g.

```
python app10.py --cat-facts 5
```

   Display the extracted facts on a screen.

3. Parsing HTML

Develop a set of functions to create a report with a list of researchers of the Faculty of Information and Communication Technology and their contact data.

1. Read and parse the content of the web page with researchers of our department with the last name starting with a given letter e.g.

```
https://wit.pwr.edu.pl/en/faculty/structure/employees?letter=P
```

2. Use the BautifulSoup module to parse web page and extract the list of researchers along with their email addresses. Develop a dataclass to store information about a teacher.

3. Print on the screen:

   1. a header (e.g. "The list of researchers - D"),

   2. a list of researchers' names along with their emails.

4. What will you do if there is no teacher with a last name starting with a given letter?

4. Print the list if your app is started in the following manner:

```
python app10.py --teachers P
```

5. Create an XLSX file with the same information and send it (as the attachment) to the given email address if the application is started in the following manner:

```
python app10.py --teachers P --send_to "some.email@example.com"
```

```
Utilize the code developed in the task 2.1
```

6. Use a tool of your choice to implement command line interface.