Lab09. Internet Access

Script Languages (INZ002025)

Wojciech Thomas

Summer 2020/2021

1 Learning goals

After this lab you should be able to:

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

2 Exercises

Artefacts to be uploaded to ePortal: - file: `app9.py`

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. Develop a function to send an e-mail to your teacher using the university mail server.
   - Use the following settings for your university account:
     - Username: university student's email address,
     - Password: your e-mail password,
     - SMTP Server: `smtp.gmail.com`.
   - Read login and password from external configuration file. Do not store your credentials in the code! Do not upload your password to the ePortal.
   - Put a current date and time in the subject of the message (with some nice words ;)).
   - Check, if the mail application correctly displays sender and recipient of the message (eg. you do not get `undisclosed recipients`).
   - Provide a content of the message using an parameter `--mail`
   - Send the e-mail only, if your app is started in the following manner:

     ```
     python app9.py --mail "My message to the teacher"
     ```

Warning: Before testing, in your Google Account turn on the option *Security / Less secure app access*. After successful sending e-mail to your teacher turn it off.

2. Develop a function to display some facts about cats.

- Study documentation of the Web API
- Print on the screen facts about cats downloaded from the web page: https://cat-fact.herokuapp.com/
- Print a particular number of facts, when your app is started in the following manner:

  ```
  python app9.py --cat-facts 5
  ```

3. Develop a function to create a report with a list of researchers of W-8 department with last names starting with a given letter.
   - 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://wiz.pwr.edu.pl/pracownicy?letter=D.
     - Use the BautifulSoup module to parse the web page and extract the list of researchers.
   - Print on the screen:
     - a header (e.g. "The list of researchers - D"),
     - a list of researchers' names along with their emails, eg.: `John Doe - john.b.doe@pwr.edu.pl`
   - What will you do, if there is no teacher with a last name starting with a given letter?
   - Print the list of teachers, only if your app is started in the following manner:

     ```
     python app9.py --teachers D
     ```