Extract Textual Data Articles

Here is a step-by-step process on how to scrape article extracts using Python:

1. **Install the required libraries**: Before starting with the scraping process, we need to install the libraries such as **BeautifulSoup**, r**equests**, and **nltk**. We can install these libraries by using the pip command.

Package	V**ersion**
pandas	1.5.3
beautifulsoup4	4.11.1
requests	2.28.1
nltk	3.8.1

e.g. !pip install <package> == <Version>

2. **Understanding the textual data article structure**: Before scraping the data, it is important to understand the structure of the textual data articles on the website and what data we want to extract from it.
3. **Get the URL of the article**: In order to extract data from an article, we need to get the URL of the article we want to scrape. We can get the URL of the article by visiting the URL and copying the URL from the input.xlsx.
4. **Send a request to the article**: Use the requests library to send a request to the article. The request will return the HTML source code of the article page, which we can use to extract data.
5. **Use BeautifulSoup to parse the HTML source code:** Use the BeautifulSoup library to parse the HTML source code of the article page. This will help us to extract data from the HTML source code in a readable format.
6. **Extract the data from the URL:** Use the BeautifulSoup library to extract the data from the LinkedIn profile. We can extract data such as URL_ID, article, paragraphs etc.
7. **Store the extracted data:** Store the extracted data in a structured format such as a CSV file. We can use the Pandas library to store the data in a CSV file.
8. **Repeat the process for multiple URL links**: To extract data from multiple articles, we can repeat the same process for multiple URL links.
9. **Save the data:** Save the data in a structured format such as a CSV file for further analysis.