Collect :

The first thing that we are going to talk about is the Collect function, since it is the first function that is executed in our program. The function is the one that is responsible of the data-gathering process. It uses four of the five libraries we've covered, and it demanded the most development time due to the continuous discovery of new challenges with the collected data.

When our program launches, the Collect function initiates checks if data is already stored in the triple store. In the presence of existing data, the function is quietly executed in the background via a thread and collects information from cooperative websites before updating the triple store. In parallel, the rest of the code proceeds without waiting for the update to conclude.

In the absence of stored data, the Collect function is executed in the foreground, and temporarily pausing the program until it completes. Its primary objective is to aggregate diverse data in RDF format from various websites.

The process starts by obtaining a JSON file from the Copcycle website, which is then transformed into an RDF graph using the RDFLib library. then, the program identifies and acquires multiple restaurant websites from each cooperative URL that is present in the JSON file. Using the BeautifulSoup library, we extract the Json-LD from these restaurant websites and store the information in a dictionary with cooperative and restaurant URLs as keys.

Some JSON-LD files didn't have the minimum price for a restaurant order. In fact these values were hidden in a JavaScript code, and required using the Selenium library to extract them. The library allowed us to load each restaurant's website, retrieve the minimum order price, and add it into the Json-LD file stored in our dictionary.

The next step was merging both RDF graphs into one. This combined graph was then transformed into a SPARQL insert query that enables the addition of all the data to the triple store. To accomplish this, we created a function translating graph values into a SPARQL query format and sending it to the triple store using the SPARQLWrapper library.

At the end the function lets us collect all the information available from the web sites and store them into a triple store.

Describe :

The describe function is the main interface between the user and the program, it guides the user, and allows him to choose the preferences of his research. It also allows him to save his preferences for future research by only inputting his name.

The function gives the user 3 ways to make their research; a manual way where he inputs his preferences without saving them, an automatic way where he can use the preferences he already saved, and an assisted way where he is asked to give an answer for some preferences that are going to be stored in the triple store and used in the future.

The function uses a simple structure that is based on inputs and conditions to gather information and validate it, and it communicate with the Query function