Data description:

Our data source is from Tencent charity. For each charity project, we are looking for project name, id, start date, end date, introduction, province, category, fund id, donor number, cumulate fund amount, target fund amount, project status, recipient name and its phone number, initiator name and its phone number. As for Users, we are interested in user name, id and password. If he is a donor, the amount he donated to each project will be recorded. So, there are 21 attributes in total. Almost all the project information can be crawled on the Internet except phone number of recipients and initiators. And all the information related to users can not be crawled. For those data can not be crawled, we generate by ourselves following some formats. In the data crawling, due to asynchronization, we get the json file url in network under developer mode and process it to get information. All the data will be stored in Excel files. In the project.xlsx, we add one column of duration which is a string contains its start date and end date. And for the column introduction, in order to encode correctly, we transform all the ‘“’,’”’(Chinese characters) to ‘[‘ and ‘]’ and delete all the hidden new line characters which may cause error when importing data to database. Lastly, we save all the Excel files as .csv format.

Difficulties we encounter:

For each charity project, they provide information such as name, current funds, status and so on. They are exactly the data we are looking for. Because they are visible in the page source code, we simply try to catch the info by raising a request. However, what we get is the framework of the page without data. We realize that data is asynchronous which means that they are put into the website in json format after the page rendered. Initially, we found the json file by clicking the network in developer mode. Next, we view the json file. The file has many useful attributes. But we open the link in web browser, all the characters are in the format of hexadecimal. And there possibly some errors in the file. If the format is wrong, the parser will occur an error so that all the info of that page will be lost. In order to make the use of every pages. We first translate hexadecimal code to UTF-8 format and take it as a String. Each pages contain 10 projects. And they all starts with the word “id”. So, we decide to split the string into 10 sub strings. Each indicates one project. And use try except syntax to handle the exception case. If we fail to catch the info because of the format error, the rest info in the same page can still be recorded. In order to crawl in total 100 pages. We change our request by replacing the page number and timestamp at the end of the URL.