

First glance look at this project it is like a select statement from a data source. Mainly idea is like get the Json string for the data (since XML is too heavy, my machine is not that powerful, that's the reason I choose Json instead of XML) and then iterator each json object to check if it is qualify. To check if it is qualify I check the current hour is larger or equal than the start hour in 24 hours and less than the end hour in 24. But there is a tricky thing under this idea, I evaluate the data and see there are some trucks which is open overnight for example, 20:00 open 2:00 close, if someone run the program at 23:00 then this food truck will not be selected. So here I choose two ways 1. If current day is the same as in the data then simply add 24 to the end time; 2. If current day is the next day show in the data (ex, today is Thursday if the data show Wednesday and end hour is less than start hour) then check if current hour is less than end hour. And here I choose less is because running the program will take time if it is already 14:00 and it will close at 14:00 then after running the program the food truck is gone, it will be more real. After check it is qualify get the applicant's name and put in a set (handle duplicates) then display

Second part I will discuss about the differences between this command line demo and a full scale web app. If I was asked to build a web app, first thing is create a data source to store all the data. Since it has json version, I suggest using Mongo. Two reasons for that: 1. This web app is read access much more than write access. 2. Take advantage of spring mongo api can easily handle pagination problem. To select qualified result, I provide two choice, 1. Select current day and yesterday's result and check the hours in servers. 2. Before insert raw data to mongo make some changes if there is overnight insert two records rather than one. And meanwhile change the time date from string to date or int. I prefer second choice since it will handle the selection in DB side which will be fast since they have multi-processing and cache which is improve the performance. Since so far the data volume is not huge, so no MQ is ok, if the data is hundreds of GB than we need to inject a MQ let the query become asynchronies. Meanwhile we can also inject a load balancer if we have multiple servers to handle request.