Case Study Software Engineer

Context

At LANCH, we want to enable our restaurant partners to have full transparency on their operations and how to grow their revenues. One important driver of revenue is the search ranking (position of restaurant) on each platform. As such, it is very beneficial for LANCH and its partners to have transparency on the current ranking of a partner as well as having the possibility to analyze rankings retrospectively.

Challenge

Your task is to build an application that retrieves and logs the ranking of a partner on Lieferando. The application should support

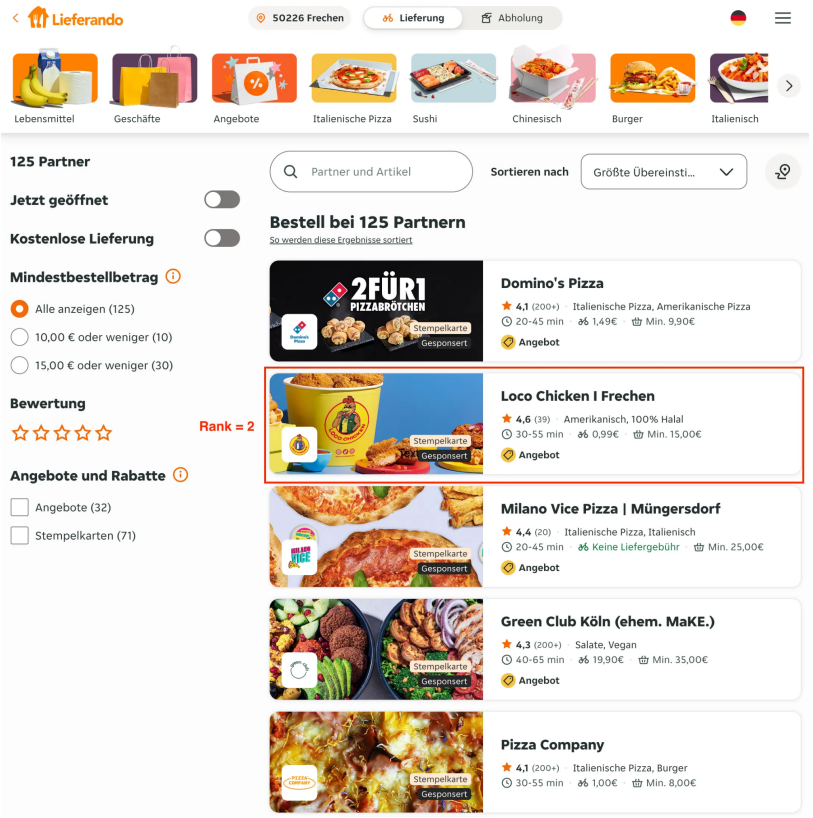
1. retrieving the current ranking ad-hoc through a REST API
2. logging the ranking of a given list of partners in intervals of at most 60 mins

Requirements specification

Lieferando Rank

Retrieve the rank as displayed to the end user on the Lieferando web app: https://www.lieferando.de/lieferservice/essen/. When searching, Lieferando requires you to enter an address. For this, use the address of the restaurant you are retrieving the rank for. You are expected to find a way to retrieve the rank of a given restaurant at the restaurants location from Lieferando's web app. If required, we encourage you to use web scraping technologies.

Retrieve the rank (where the top result has rank 1, followed by 2, etc.) without any search filters applied. When logging the result data in the DB, add any fields/information from the result page you consider worth logging in the given context.



Sample Lieferando results page. For “Loco Chicken | Frechen”, the rank is 2.

Containerization

Use Docker to containerize your application. Running `docker compose up` should be sufficient to get the application up and running on any machine.

REST API

The API should have exactly one endpoint:

`GET` `/rank/<restaurant_slug>`

```
{
    "restaurant_slug": "<restaurant_slug>",
    "rank": 4
}
```

The minimum response (see above) should be expanded with any fields you consider relevant.

You are *not* expected to implement authentication. You are neither expected to add routing (→ use http://localhost:8080 as base URL for the API).

Logging of ranks

For the below given list of restaurant slugs, the ranks should be tracked in intervals of at most 60 minutes. The results should be written to a database. This can be done with a simple Python and / or bash script, using cron is optional. You are not expected to retrieve the slugs to retrieve the rank for from the DB and are encouraged to hard-code them in your script for simplicity.

Slugs to retrieve ranking for:

```
[
    "loco-chicken-i-frechen",
    "loco-chicken-bielefeld",
    "happy-slice-suedstadt",
    "happy-slice-pizza-i-wandsbek-markt",
]
```

Tech stack

- Language - Please use Python for the main application logic
- API - Use any Python framework of your choice
- DB - Use a relational database with any dialect of your choice
- Containerization - Use Docker with docker compose

General guidelines

Presentation

- Please send us your results / git repo at least two hours before the presentation
- We want you to present the approach you chose on a high level
- Focus on outlining why you took certain design decisions instead of going too deep into the implementation
- We don’t expect you to prepare a fancy presentation - guiding us through your thoughts and application structure in a clear and organized manner will suffice

Evaluation

We are going to evaluate your submission using the following criteria:

- Ease of setting the application up and reproducibility
- Core functionality
- Handling of erroneous input / output
- DB utilization
- Code quality
- Design decisions and reasoning

Solution finding

Your job at LANCH will be a lot about finding solutions that satisfy the 80/20 rule. We also want you to demonstrate this ability in the case study. Focus on delivering a solution of good quality without overengineering it. We are expecting you to leverage current technologies like LLMs for development and code frameworks for quicker implementation.

We know that you are spending considerable time and energy in preparing this case study and appreciate that a lot! As such, we encourage you to add the resulting project to your personal portfolio.