enconso GmbH - Hölderlinweg 17b - 61350 Bad Homburg

enconso GmbH
Hölderlinweg 17b
61350 Bad Homburg
T: +49 6172 868449
F: +49 6172 868447
M: +49 173 9506459
E: info@enconso.com

Application for Curry-Wurst Shop

Dictionary

- *Order* - information about *menu items* to be cooked, count of each *item*, order arrival time and expected time. Example order: Curry Wurst x 2, Coca-Cola x 1, Fries x 3, arrived at 13:00, should be cooked until 14:00
- *Menu items* - list of *recipes*, that can be cooked in shop.
- *Recipe* - contains information about dish: name, time for cooking, number of *ingredients* required for cooking of dish.
- *Ingredient* - unit of product, required for cooking. Contains name of *ingredient* and amount of time, required for delivery of one unit from *warehouse* to *kitchen*.
- *Cook* - performs cooking of a dish via *recipe*. Cooking requires all *ingredients* from *recipe* on *kitchen* and some time. A *cook* also can order *ingredients* from *the warehouse*.
- *Kitchen* - all *cooks* are on kitchen and can use only *ingredients* from it.
- *Warehouse* - infinite *ingredient* storage.
- *Worker* - move *ingredients* from *warehouse* to *kitchen*. Each *worker* can move only one type and one unit of *ingredient* at once (*optional: workers can more than one unit at once, defined by their capacity*). This action requires time, described in the *ingredient*.
- *Cooking plan* - description, how *orders* are distributed: at which time and which *cook* starts cooking and which *workers* perform a delivery of required *ingredients*

Description

Functions

Application should model the work of a shop, which sells curry wursts to customers. The customer places an *order*. *Order* should be cooked by *cook* in a given time.
The Cook works on a *kitchen*. The shop can have multiply *cooks*, which are working parallel. If there are not enough *ingredients* on the *kitchen*, *the cook* can ask the required amount of them from *the warehouse*. If all *cooks* are busy, *order* should wait, until it will be a *cook* available.

The warehouse is managed by *workers*. It can be more than one *worker* in the warehouse. All of them are working parallel. If all *workers* are already busy, *ingredients* can not be delivered.
Cooking time of the *order* is a summary of time, required for cooking and delivery of all ingredients (including waiting time for free *cooks* and *workers*).
The *order* can be accepted, if it can be processed in given or smaller time.

Requirements

API

An application should provide next endpoints:

1. Make an *order*. Endpoint should receive the *order*. Application should respond “OK”, if the *order* can be cooked in the given time (with attention to other accepted orders) and accept this *order*.
2. Display a *cooking plan.* Endpoint should respond with a *cooking plan* for currently accepted *orders*.
3. *Optional:* CRUD API for:

a. *Cooks*
b. *Workers*
c. *Recipes*
d. *Ingredients*

Technologies

Required technologies: Spring Boot 2.3+, Lombok, JPA + H2, Web API, Swagger (springfox), Git
Optional: Frontend via Angular 2+/React, Docker, Liquibase

Additional

The source code of the resulting application should be uploaded to Github, Gitlab or other free git-server.

enconso GmbH	T: +49 6172 868449	Geschäftsführer	Amtsgericht Bad Homburg	Postbank
Hölderlinweg 17b	F: +49 6172 868447	Igor Friedman	HRB 11364	Konto: 25205606
61350 Bad Homburg	M: +49 173 9506459		Steuernummer:	BLZ: 50010060
	E: info@enconso.com		003 232 41053	IBAN: DE93500100600025205606
	www.emconso.com		DE266199003	BIC: PBNKDEFF