

Programming task for LYON Future

The purpose of the task is developing a small service using a technology stack similar to LYON Future development and deployment platform. Solution can be developed using your favorite IDE. Share your code with us by using a public repo like github.

Pre requirements

Download XML file from PeppolDirectory - <https://directory.peppol.eu/public>. This dataset represents European enterprises and their capabilities to receive electronic invoices by Peppol infrastructure

Requirements

- Solution has to be built in Java
- Source code must compile
- Use a build script, preferable maven
- Postman collection for testing all of the API methods
- Deployment must be possible in Tomcat or Wildfly (your choice)
- Provide deployment information (how to run it)

Your task is to develop a rest-based HTTP service for doing various queries on the XML dataset. Solution must be developed in Java8 or 11. You don't need to think about security. You can design the architecture as you like

Task 1: getById

Develop a service (method) for returning one entity from the dataset based on id. Id in the dataset is the value of the /root/businesscard/participant/@value. Format of the id is <ICD>:<business number>. E.g value="9908:918039899" means ICD 9908 (which is Norway) and 918039899 which is the enterprise number for a company called GULENG AS.

Parameters:

Input: 2 parameters ICD and enterprise number

Output: Business card for the enterprise as a JSON.

Task 2: searchByName

Develop a service (method) for returning a list of entities from the dataset based on a name search. The service shall search in the dataset and return a list of matches. "Name" is located in the dataset at the element /root/businesscard/entity/name/@name. This service must do a wildcard search for names. E.g. if "CAR" is the input "SUPER CARS", "CAR WASH" AND "CARLINGS" will result in a hit.

Parameters:

Input: Name to search for

Output: JSON containing Name, Enterprise number and country code.

Task 3: `getByName`

A service similar to `getById`. This service shall return one business card from the dataset based on an exact match in the tag `/root/businesscard/entity/name/@name`

Parameters:

Input: Name to search for

Output: JSON representing the value in one `/root/businesscard/`.