Objective

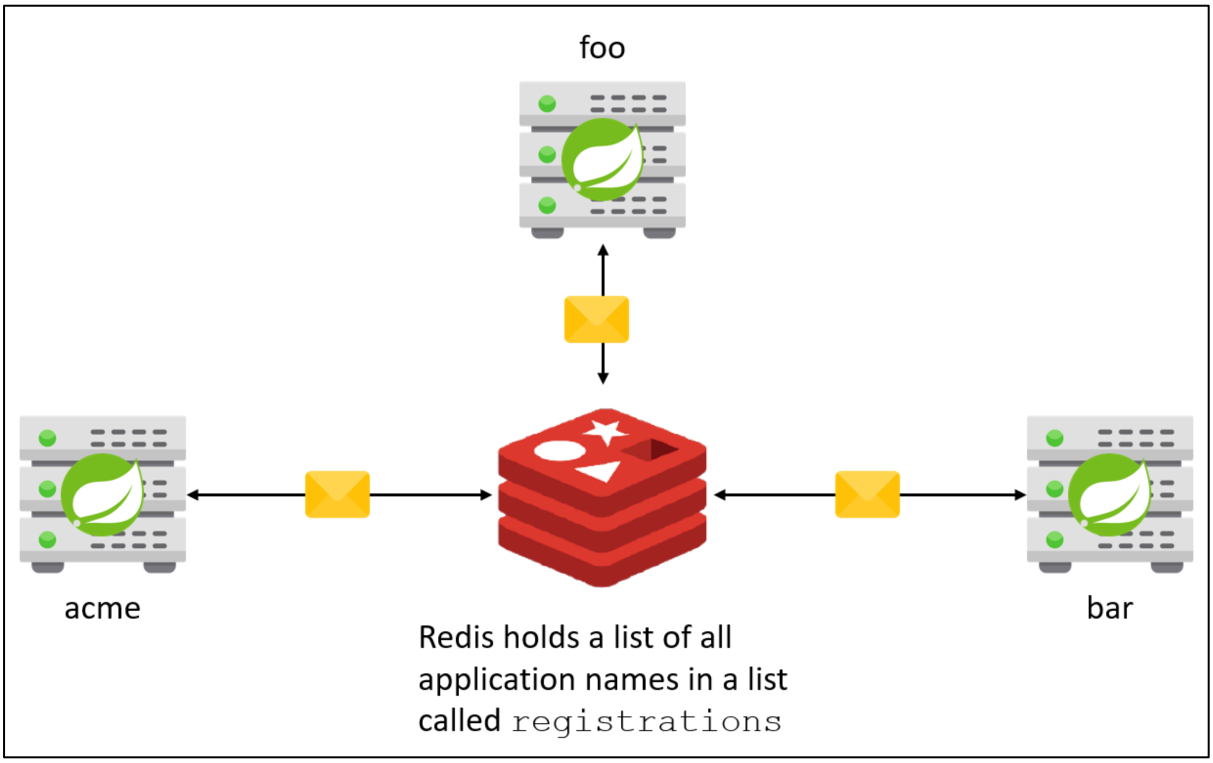
The objective of this workshop is to create route orders to different Spring Boot application using Redis Queue

Setup

a. Provision a new Redis instance or us an existing instance.
b. Generate a SpringBoot application and include the following dependencies:
   i. SpringBoot Dev Tools
   ii. Spring Web
   iii. Thymeleaf
   iv. Redis and Jedis
   v. JDBC and MySQL
   vi. JSON-P
c. Create a branch of day 24 workshop

Workshop

In this workshop, you will be writing a Spring Boot application to send orders to other application over with Redis pubsub


foo
acme
bar
Redis holds a list of all application names in a list called `registrations`

Modify day 24 application to get a name from the command line when the application starts up; for example

```
mvn spring-boot:run -Dspring-boot.run.arguments=acme
```

or

```
java -jar target/application.jar acme
```

sets the application name to `acme` .

Add this name to a list in Redis called `registrations`.

The application should listen to a list of its own name; for example, if the application name is `acme`, then the application should listen to a list called `acme`.

Modify the create order page to include a drop-down list of all the names from the `registrations` list.

When the order is posted, instead of saving it to MySQL database, convert the message to a JSON object of the following structure

```
{
  "customer_name": <a name from registrations>,
  "ship_address":  "....",
  "notes": "...",
  "tax": <tax>,
  "line_items": [
    {
      "product": <prod_id>,
      "unit_price: <unit price>,
      "discount": <discount>,
      "quantity", <quantity>
    }
    ...
  ]
}
```

Push the order to the list of the `customer_name`; for example if the customer name is `foo`, then push the order to the `foo` list.

When the consumer, the Spring Boot application, receives (pops) the JSON order, it should save it to the orders table according to day 24 workshop.