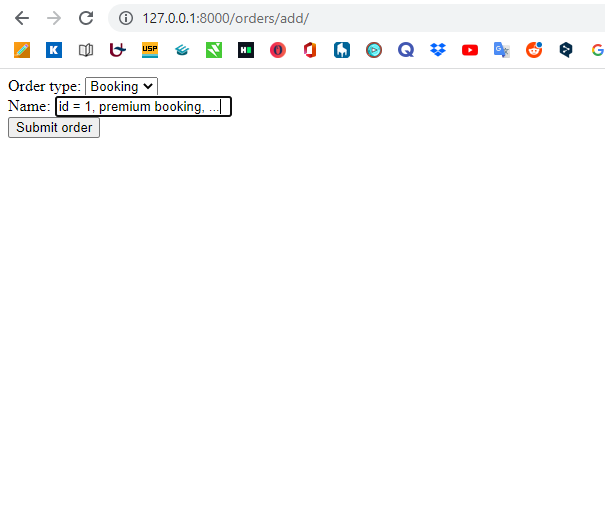
Homework 8

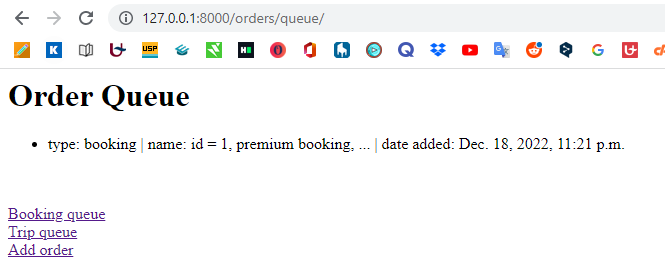
# Adding order to order queue

First, we will need to add orders. An order can be either a booking or a trip. We are using a very simplified booking and trip where for each, we only add the name. The datetime will be autogenerated. The interface that we will be using is a very simple HTML GUI. We could have also used a rest API (or others) but this way it is simpler to demonstrate.

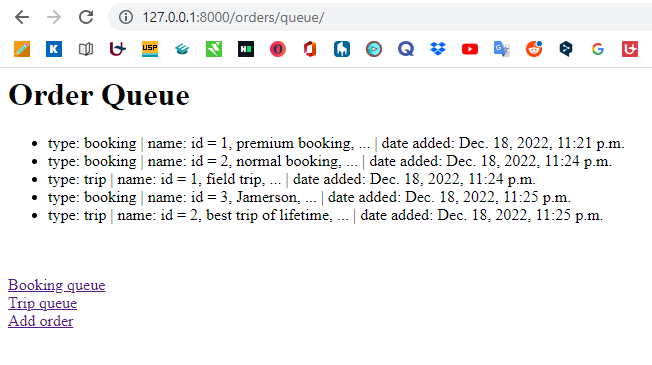
Endpoint: /orders/add/



If we now click on ‘submit order’, the order will be added to the order queue. We will also be redirected to a very simple html GUI of the order queue. Again, we used this for simplicity.

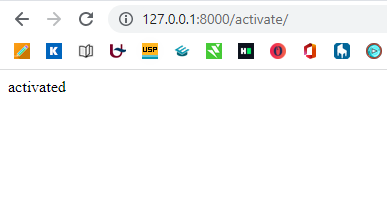


Lets add another few orders. We now have the following order queue.

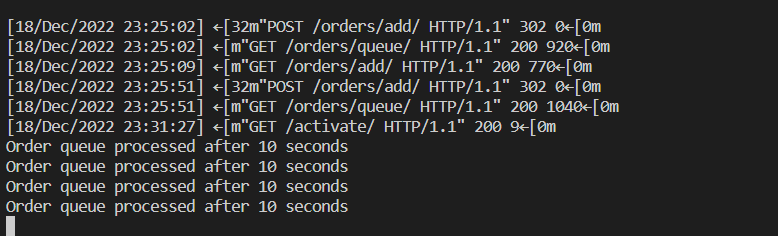


# Activating order processing

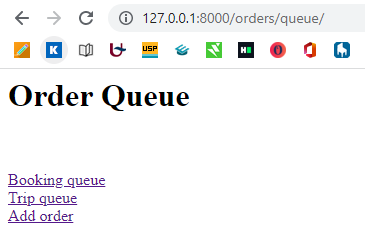
Now, if we want to activate our order processing process, we can use the endpoint /activate. This will run a while true-loop that will constantly (every 10 seconds) empty the order queue and put its items in respectively the trip queue and the booking queue. The refreshing of 10 seconds is arbitrary.

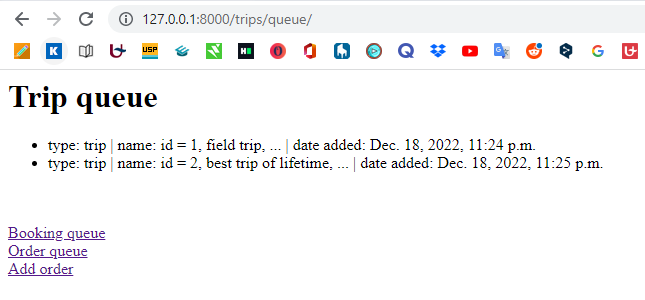


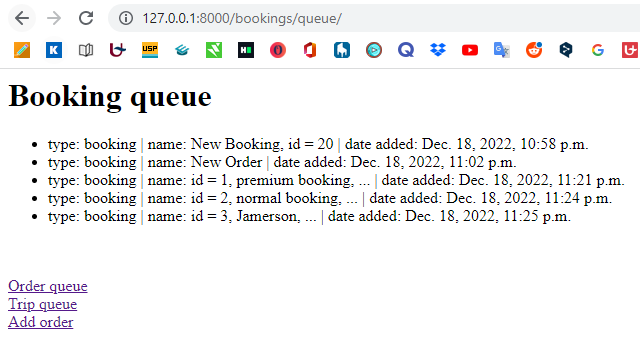
Our console gives the following.



2 things will have happened. Foremost, the order queue will have been emptied by the process. Secondly, the trips and booking queue will have been filled with the orders.







# Processing from individual queues

The last thing that we now want to implement is an individual consumer that processes the individual trips and bookings. For simplicity purposes, we won’t go into detail of processing, but we will show how it can be done.

If we want an individual processor to get a trip or booking out of the queue, they can use the endpoint /trips/dequeue/ or /bookings/dequeue/. This will give a http response in json format of the booking/trip that was first in line (added first). We’ll demonstrate using curl.

bloom@David-PC MINGW64 ~

$ curl -X GET http://127.0.0.1:8000/trips/dequeue/

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 92 100 92 0 0 15587 0 --:--:-- --:--:-- --:--:-- 18400{"type": "trip", "name": "id = 1, field trip, ...", "date\_added": "2022-12-18T23:24:34.444"}

bloom@David-PC MINGW64 ~

$ curl -X GET http://127.0.0.1:8000/trips/dequeue/

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 103 100 103 0 0 35677 0 --:--:-- --:--:-- --:--:-- 51500{"type": "trip", "name": "id = 2, best trip of lifetime, ...", "date\_added": "2022-12-18T23:25:51.460"}

bloom@David-PC MINGW64 ~

$ curl -X GET http://127.0.0.1:8000/trips/dequeue/

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 17 100 17 0 0 7172 0 --:--:-- --:--:-- --:--:-- 17000no trips in queue

We can see that our trip processor requested all the trips in the queue. The queue is now empty.

