InflightIT

Imagine, nós executamos

Challenger

Create a queue system that delivers all features required to handle a queue in an on-site customer service. This system is capable of managing a queue, prioritizing customers and resetting it at the end of the day.

When a customer arrives at the store his name is added to the queue with a priority. Priority is used to prioritize people with disabilities, pregnant women and senior citizens and therefore the higher the priority, the higher its precedence.

Priority is assigned by an employee at the time of the client's check-in and its value ranges from 0 to 10. 0 means no priority (end of the queue) and 10 means high priority.

[Bonus] At the end of the day, employees are able to clean the queue (reset) in order to remove not attended customers from the list and avoid not carrying over them to the next day.

Add customer

At the end of this operation, the queue should be ordered by priority and insertion order.

```
POST: {host}/api/v0/addCustomer
```

Payload type

```
{
   name: String,
   priority: Number
}
```

Examples

```
{
   name: "John Doe",
   priority: 0
}
```

Result type

```
{
   result: true | string,
}
```

Examples

Success message

```
{
   result: true
}
```

Error message (missing name)

```
{
   result: “Missing name”
}
```

Get next customer (by priority and FIFO *first in first out*)

```
GET: {host}/api/v0/getNextCustomer
```

Result type

```
{
   result: Client | null,
}
```

Examples

Queue has one or more clients

```
{
   result: {name: “John Doe”, priority: 0}
}
```

Queue is empty

```
{
   result: null
}
```

[BONUS FEATURE] Reset queue

This feature is performed in two steps. First it is asked for a code (authorization) in order to reset the queue and then we use that code to effectively reset the queue.

Part 1

```
GET: {host}/api/v0/ask-to-reset
```

Result type

```
{
   code: string,
}
```

Examples

```
{
   code: “2CF24DBA5FB0A30E26E83B2AC5B9E29E1B161E5C1FA7425E73043362938B9824”
}
```

Tips

- Use an hash generator in order to generate the code

Part 2

`GET: {host}/api/v0/reset?code={code}`

Result type

```
{
   result: boolean,
}
```

Examples

Queue reseted successfully

```
{
   result: true
}
```

Queue not reseted (wrong code provided)

```
{
   result: false
}
```

Test examples

```
POST: {host}/api/v0/addCustomer {“Rottenmeier”, priority: 0}
POST: {host}/api/v0/addCustomer {“Clare”, priority: 10}
POST: {host}/api/v0/addCustomer {“Sesemann”, priority: 0}
POST: {host}/api/v0/addCustomer {“Uncle”, priority: 5}
POST: {host}/api/v0/addCustomer {“Peter”, priority: 0}
POST: {host}/api/v0/addCustomer {“Mrs. Sesemann”, priority: 7}
```

Queue snapshot (descending sort by priority and insertion time)

```
1. {“Clare”, priority: 10}
2. {“Mrs. Sesemann”, priority: 7}
3. {“Uncle”, priority: 5}
4. {“Rottenmeier”, priority: 0}
5. {“Sesemann”, priority: 0}
6. {“Peter”, priority: 0}
```

```
GET: {host}/api/v0/getNextCustomer
RESULT: {“Clare”, priority: 10}

GET: {host}/api/v0/getNextCustomer
RESULT: {“Mrs. Sesemann”, priority: 7}
```

Queue snapshot (descending sort by priority and insertion time)

```
1. {“Uncle”, priority: 5}
2. {“Rottenmeier”, priority: 0}
3. {“Sesemann”, priority: 0}
4. {“Peter”, priority: 0}
```

Tech stack

- NestJS
    - nodejs
- Mongoose
    - MongoDB

InflightIT

Guimarães

Avepark - Parque de Ciência
e Tecnologia S.A, Sala 211

4805-017 Barco Guimarães

Esposende

Largo Rodrigues Sampaio,
nº 37

4740-218 Esposende

www.inflightit.com 936100891 info@inflightit.com

/inflightit /inflightit /inflightit /company/inflightit