

Gestione Cliente

Il microservizio Gestione Cliente si occupa principalmente di gestire l'interazione di un cliente col sistema e di gestire la sua relativa comanda di ordini.

La comunicazione con gli altri microservizi avviene tramite Message Broker come segue:

- Il microservizio GestioneCliente comunica verso GestioneComanda tramite il topic Kafka NotifyOrderEvent.

Le API riguardano l'interazione del cliente con il servizio

GET Get Menù

```
localhost:8080/cliente/menu
```

API di Get con la quale si richiede il menù (lista di piatti disponibili)

Risposta:

Lista di oggetti piatto:

- `id` : identificativo del piatto
 - `idIngrPrinc` : identificativo dell'ingrediente principale
 - `descrizione` : descrizione del piatto
 - `prezzo` : prezzo del piatto
 - `tpreparazione` : tempo medio di preparazione del piatto
-

GET Get piatto from menù

```
localhost:8080/cliente/menu/car123
```

API di Get con la quale si richiede uno specifico piatto dal menù

Parametri della query string:

- `idpiatto` (obbligatorio): Identificatore del piatto da recuperare

Risposta:

- `id` : identificativo del piatto
 - `idIngrPrinc` : identificativo dell'ingrediente principale
 - `descrizione` : descrizione del piatto
 - `prezzo` : prezzo del piatto
 - `tpreparazione` : tempo medio di preparazione del piatto
-

GET Get new Comanda from Cliente

```
localhost:8080/cliente/tavolo1/comanda/new
```

API di Get con la quale si richiede una nuova comanda per un determinato cliente

Parametri della query:

- `idCliente` (obbligatorio, stringa) : identificativo del cliente

Risposta:

- `id` : identificativo della comanda
 - `idCliente` : identificativo del cliente
 - `codicePagamento` : codice di pagamento
 - `totaleScontrino` : costo totale dei piatti ordinati a scontrino
-

GET Get Comanda attiva from Cliente

```
localhost:8080/cliente/tavolo1/comanda/attiva
```

API di Get con la quale si richiede la comanda attiva per un dato cliente

Parametri della query:

- `idCliente` (obbligatorio, stringa) : identificativo del cliente

Risposta:

...response...

- `id` : identificativo della comanda
- `idCliente` : identificativo del cliente
- `codicePagamento` : codice di pagamento
- `totaleScontrino` : costo totale dei piatti ordinati a scontrino

POST Post new Order by Cliente

```
localhost:8080/cliente/order/add?idcliente=tavolo1&idpiatto=car123&urgenzacliente=0
```

API di Post con la quale si salva un ordine per un dato cliente all'interno del sistema

Parametri della query:

- `idcliente` (stringa, obbligatorio) : identificativo del cliente
- `idpiatto` (stringa, obbligatorio) : identificativo del piatto
- `urgenzacliente` (numero, obbligatorio): Attributo urgenza del cliente(0 : espresso non urgenza, 1 : espresso urgenza, 2 : default)

Risposta:

- `id` : Identificatore dell'ordine
- `idComanda` : Identificatore della comanda di cui l'ordine fa parte
- `idPiatto` : Identificatore del piatto ordinato dal cliente
- `stato` : Stato dell'ordine (0: Ordine preso in carico, 1: Ordine in coda di preparazione, 2: Ordine in preparazione, 3: Ordine preparato)
- `t0ordinazione` : Istante temporale in cui viene effettuata l'ordinazione
- `urgenzaCliente` : Attributo urgenza del cliente (0 : espresso non urgenza, 1 : espresso urgenza, 2 : default)

PARAMS

<code>idcliente</code>	tavolo1
<code>idpiatto</code>	car123
<code>urgenzacliente</code>	0

GET Get order

```
localhost:8080/cliente/order?id=2
```

API di Get con la quale si richiede un ordine specifico

Parametri della query:

- `id` (numero, obbligatorio) : identificativo dell'ordine richiesto

Risposta:

- `id` : Identificatore dell'ordine
- `idComanda` : Identificatore della comanda di cui l'ordine fa parte
- `idPiatto` : Identificatore del piatto ordinato dal cliente
- `stato` : Stato dell'ordine (0: Ordine preso in carico, 1: Ordine in coda di preparazione, 2: Ordine in preparazione, 3: Ordine preparato)
- `tOrdinazione` : Istante temporale in cui viene effettuata l'ordinazione
- `urgenzaCliente` : Attributo urgenza del cliente (0 : espresso non urgenza, 1 : espresso urgenza, 2 : default)

PARAMS

id	2
----	---

GET Get Order status

```
localhost:8080/cliente/order/status?id=2
```

API di Get con la quale si richiede lo stato di un ordine specifico

Parametri della query:

- `id` (numero, obbligatorio) : identificativo dell'ordine richiesto

Risposta:

- `ordine` : Identificatore dell'ordine
- `stato` : Stato dell'ordine (0: Ordine preso in carico, 1: Ordine in coda di preparazione, 2: Ordine in preparazione, 3: Ordine preparato)

preparazione, 3: Ordine preparato)

PARAMS

id	2
----	---

GET Get orders from cliente

```
localhost:8080/cliente/tavolo1/orders
```

API di Get con la quale si richiede una lista di tutti gli ordini per un dato cliente

Parametri della query string:

- `idCliente` (obbligatorio, stringa): Identificatore del cliente in questione

Risposta:

Lista di oggetti Ordine:

- `id` : Identificatore dell'ordine
- `idComanda` : Identificatore della comanda di cui l'ordine fa parte
- `idPiatto` : Identificatore del piatto ordinato dal cliente
- `stato` : Stato dell'ordine (0: Ordine preso in carico, 1: Ordine in coda di preparazione, 2: Ordine in preparazione, 3: Ordine preparato)
- `tOrdinazione` : Istante temporale in cui viene effettuata l'ordinazione
- `urgenzaCliente` : Attributo urgenza del cliente (0 : espresso non urgenza, 1 : espresso urgenza, 2 : default)