

Car Rental Agency

Our project consist of 2 microservices, one document DB (mongoDB) and also one MQ (RabbitMQ).

Customer-service

The role of customer-service is to provide users with the possibility of

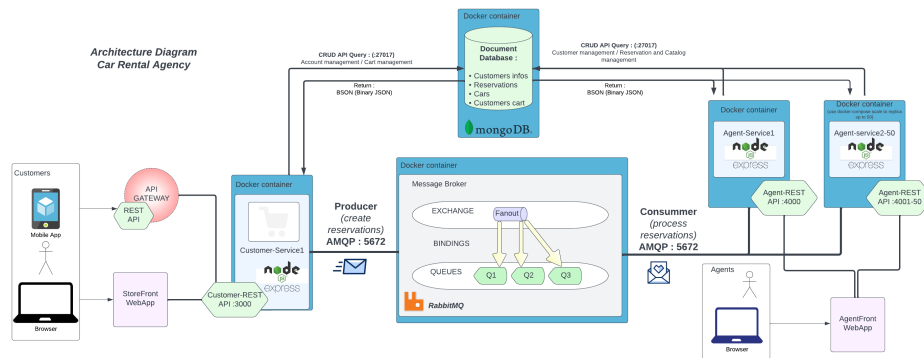
- Customer registers for a new account
- Customer retrieves account information
- Customer can update his informations
- Customer can retrieve the detials about his reservations
- Customer can delete his account
- Customer lists cars from the catalog
- Customer searches by criteria from the database
- Customer can retrieve info about specific car
- Customer adds car rentals to the cart
- Get his cart content
- Clear his cart
- Submit his cart content

The role of Agent-service is to provide users with the possibility of

- Agent checks whether a customer (identified by his email) is registered
- Agent can retrieve reservation of a customer (identified by his email)
- Agent adds new cars to the catalog
- Agent can create a reservation for a customer (identified by his email) from agency
- Agent can lists cars from the catalog
- Agent checks availability for a specific car
- Agent can check all exisiting reservations

See below the description of the applications

Architecture Diagram :



Project setup

```
docker-compose up -d
```

(Optionnal) Scale the agent service (up to 50)

```
docker-compose up --scale agent-service=<1-50>
```

Api Specification

📁 Customer-service end-points

End-point: Create an account

Method: POST

```
localhost:3000/api/account/create
```

Headers

Content-Type	Value
Content-Type	application/json

Body (raw)

```
{
  "name" : "John",
  "surname": "Doe",
  "email": "JohnDoe@test.com",
  "password": "azerty123",
}
```

```
"address" : "6 avenue de la République",  
"city" : "Villejuif",  
"zipCode" : "94200",  
"country" : "France"  
}
```

End-point: Update account info

Method: PUT

```
localhost:3000/api/account/update
```

Body (raw)

```
{  
  "email": "JohnDoe@test.com",  
  "password": "azerty123",  
  "city" : "Paris",  
  "zipCode" : "75012"  
}
```

End-point: Get account information

Method: GET

```
localhost:3000/api/account/infos
```

Body (raw)

```
{  
  "email": "JohnDoe@test.com",  
  "password": "azerty123"  
}
```

End-point: Delete account

Method: DELETE

```
localhost:3000/api/account/delete
```

Body (raw)

```
{  
  "email": "JohnDoe@test.com",  
  "password": "azerty123"  
}
```

End-point: Get reservations account

Method: GET

```
localhost:3000/api/account/reservations
```

Body (raw)

```
{  
  "email": "JohnDoe@test.com",  
  "password": "azerty123"  
}
```

Collection: Rental system

End-point: Get available cars

Method: GET

```
localhost:3000/api/rentalSystem/catalog
```

End-point: Get a specific car (brand/model/numberOfSeat)

Method: GET

```
localhost:3000/api/rentalSystem/specificCars
```

Body (raw)

```
{  
  "model" : "A3"  
}
```

End-point: Get infos about a cars

Method: GET

```
localhost:3000/api/rentalSystem/63cfbb223e0257761df4a0d
```

Collection: Cart management

End-point: Submit a reservation item to the Cart

Method: POST

```
localhost:3000/api/cart/
```

Body (raw)

```
{
  "email": "JohnDoe@test.com",
  "password": "azerty123",
  "carID" : "63e22d423468f5ebc0e01e6c",
  "startDate" : "2021-03-20",
  "endDate" : "2021-03-27"
}
```

End-point: Delete an Item of your cart

Method: DELETE

```
localhost:3000/api/cart/
```

Body (raw)

```
{
  "email": "JohnDoe@test.com",
  "password": "azerty123",
  "reservation_id" : "63e22a51a942795aac385702"
}
```

End-point: Retrieve cart informations

Method: GET

```
localhost:3000/api/cart
```

Body (raw)

```
{
  "email": "JohnDoe@test.com",
  "password": "azerty123"
}
```

End-point: Submit cart

Method: POST

```
localhost:3000/api/cart/submit
```

Body (raw)

```
{
  "email": "JohnDoe@test.com",
  "password": "azerty123",
  "paymentMethod": "creditCard",
  "creditCardNumber": "2345234523452345"
}
```

Collection: agent-service

Collection: RentalSystem

End-point: Add new car to catalog

Method: POST

```
localhost:4000/api/car/catalog
```

Body (raw)

```
{
  "email": "agent1@car.com",
  "password": "123456",
  "brand": "Citroen",
  "model": "C4",
  "numberOfSeat": "5",
  "pricePerDay": "80",
  "available": true
}
```

End-point: Create a reservation from an agency

Method: POST

```
localhost:4000/api/car/reservation
```

Body (raw)

```
{
  "email": "agent1@car.com",
  "password": "123456",
  "carID" : "63da2f3e5683291f48a3be1b",
  "customerEmail": "JohnDoe@test.com",
  "startDate" : "2020-04-24",
  "endDate" : "2020-05-4",
  "paymentStatus" : "Paid" ,
  "paymentMethod" : "creditCard",
  "ReservationStatus" : "Pending"
}
```

End-point: Check availability of a car

Method: GET

```
localhost:4000/api/car/availability
```

Body (raw)

```
{
  "email": "agent1@car.com",
  "password": "123456",
  "carID" : "",
  "startDate" : "2022-04-24",
}
```

```
"endDate" : "2022-05-4"
}
```

End-point: List all existing reservation

Method: GET

```
localhost:4000/api/car/listReservation
```

Body (raw)

```
{
  "email": "agent1@car.com",
  "password": "123456"
}
```

End-point: List all existing cars

Method: GET

```
localhost:4000/api/car/catalog
```

Body (raw)

```
{
  "email": "agent1@car.com",
  "password": "123456"
}
```

Collection: CustomerManagement

End-point: Check if client is registred

Method: GET

```
localhost:4000/api/customer/isClient
```

Body (raw)


```
{  
  "email": "agent1@car.com",  
  "password": "123456",  
  "customerEmail": "JohnDoe@test.com"  
}
```

End-point: Retrieve Client reservations

Method: GET

localhost:4000/api/customer/reservations

Body (raw)

```
{  
  "email": "agent1@car.com",  
  "password": "123456",  
  "customerEmail": "JohnDoe@test.com"  
}
```
