SOCKET HANDLE

--------------------------------------------------------------------

Đăng nhập thành công, xác thực socket bằng token:

client emit 'storeInfo', data:

{

token: String,

customerId: String,

driverId: String

}

-------------------------------------------------------------------

Client emit 'heartbeat' khoảng vài s 1 lần, nếu trong vòng 10s mà client không gửi heart beat thì server sẽ disconnect socket

---------------------------------------------------------------------

test: Client emit 'location' , data: driver

{

lat: Number,

long: Number

}

-> server sẽ gửi lại location 3s 1 lần (để test thôi)

--------------------------------------------------------------------

Sau khi đăng nhập và xác thực socket thành công, server emit sự kiện 'welcome':

- Customer nhận sự kiện 'welcome': emit sự kiện 'customerHello' với dữ liệu:

{

city: String

}

-> server nhận sự kiện 'customerHello', gửi lại cho customer danh sách (khởi tạo) driver đang online trong city: emit 'helloCustomer' với dữ liệu:

{

drivers: [{

socketId: String,

location: {...}

}]

}

- Driver nhận sự kiện 'welcome': emit sự kiện 'driverHello' với data:

{

city: String,

location: {

...

}

}

-> server nhận sự kiện 'driverHello', gửi lại cho driver danh sách (khởi tạo) customer đang tìm xe trong city: emit 'helloDriver' với data:

{

customers: [{

socketId: String,

productInfo: {..} (tọa độ bắt đầu của gói hàng lưu trong productInfo )

}]

}

-> server emit cho các customer trong city sự kiện ‘newDriver’:

{

socketId: String,

location: {...}

}

->

---------------------------------------------------------------------

Driver định kì gửi tọa độ lên server: emit ‘driverLocationUpdate’, dữ liệu:

{

location: {...}

}

-> server nhận tọa độ driver, emit cho customer trong city sự kiện ‘driverLocationUpdate’ :

{

socketId: String,

location: {...}

}

------------------------------------------------------------------------

Server 1p 1 lần gửi tọa độ các driver trong city cho customer: emit ‘driversLocation’, dữ liệu:

{

drivers: [{

socketId: String,

location: {

lat: Number,

long: Number

}

}]

}

-> customer get về cập nhật map

---------------------------------------------------------------------

Customer bấm tìm xe: emit 'findDriver' với data:

{

city: String,

productInfo: {..}

}

-> server thông báo cho các driver đang online trong khu vưc: emit 'customerFindDriver' với data:

{

customerSocketId: String,

productInfo: {..}

}

(driver nhận được dữ liệu thì đưa lên bản đồ ..)

-> server thông báo cho các driver đang offline trong khu vực: gửi message firebase:

{

collapse\_key: customerFindDriver,

data: {

customerSocketId: String,

productInfo: {..}

}

}

-----------------------------------------------------------------------

Customer dừng tìm kiếm, thông báo lên server: emit 'stopFinding'

-> server nhận thông báo, báo lại cho các driver trong khu vực: emit 'customerStopFinding'

-> driver ...

-----------------------------------------------------------------------

Driver bấm chọn customer: emit 'driverGetCustomer', data:

{

customerSocketId: String

}

-> server thông báo cho các driver khác trong cùng city là customer đã được chọn: emit 'customerBooked', data:

{

customerSocketId: String

}

-> server thông báo cho các customer trong city là driver đã bận: emit ‘driverBooked’ :

{

driverSocketId: String

}

-> server tạo phòng riêng giữa customer và driver

-> thông báo cho customer và driver: emit sự kiện 'roomCreated' vào room vừa tạo, data:

{

driverSocketId: String,

customerSocketId: String

}

--------------------------------------------------------------------------

Customer nhận sự kiện 'roomCreated', làm gì thì làm ..

--------------------------------------------------------------------------

Driver nhận sự kiện 'roomCreated', dừng gửi sự kiện ‘driverLocationUpdate’, xử lí (..), bắt đầu gửi tọa độ: emit 'bookedDriverLocation', data:

{

location: {

lat: Number,

long: Number

}

}

(?s 1 lần)

-> server nhận tọa độ, gửi lại cho customer: emit 'bookedDriverLocation', data:

{..} (như trên)

----------------------------------------------------------------------------

Driver gửi thông báo kết thúc chuyến xe: emit 'finish'

-> server nhận thông báo, báo cho customer: emit 'finish'

------------------------------------------------------------------------------------

Sau khi finish, server emit cho driver ‘welcomeBack’ :

-> driver nhận sự kiện: emit lại ‘driverHello’ như trên và bắt sự kiện ‘helloDriver’

-> driver tiếp tục gửi tọa độ lên cùng sự kiện ‘driverLocationUpdate’

------------------------------------------------------------------------------------