Server:

currentRideQueue: Queue<Ride>

laterQueue: Queue<Ride>

Our server finds the **next closest** and **available driver** for a given rider.

We will use this ordering to send current and later ride queues to drivers.

Server maintains logic for adding un-selected future rides to current ride queue.

If driver calls acceptRide(), this triggers a DELETE request. This will remove the Ride from the current or later queue.

Rider  
requestRideNow()

requestRideLater()

Rider  
requestRideNow()

requestRideLater()

Rider  
requestRideNow()

requestRideLater()

Send updated ride queues and notify driver of updates.

Driver

currentRideQueue: Queue<Ride>

laterQueue: Queue<Ride>

isOnline: Boolean

isAvailable: Boolean

viewRide()

acceptRide (param: queueToUpdate)  
rejectRide()

Driver polls API every 5-10 seconds for queue changes

acceptRide()  
removes ride from currentRide queue

NOTES:

**Book Future Ride:**

1. EX: Rider sends request for later ride 🡪 accept driver  
   Another rider sends request for right now 🡪 same driver receives “right now” request  
   Display a message: “Your next ride scheduled for 8pm. You have XX minutes left and your next ride is YY km away. Are you sure you have enough time to get there while driving safely?”
2. How do we deal with future rides -> and if driver is late / misses the pickup?
3. Driver UI – 2 different “modes” for accepting a ride.   
   We can accept a current ride or future ride. (TODO: adjust the diagram so driver can do both).

**Book Current Ride:**

1. How do we assign riders to drivers?  
   Possibilities:  
   - Driver at shortest distance from rider?  
   - Do all drivers get a request?

Our system finds the **next closest** and **available driver.**

**We can send multiple requests to more than 1 driver** (case of 1 or more drivers considerably close to rider).