1. **The First main Graphql is Passenger Subscription , this graphql is called before the passenger screen is load and gets all stops , route and dropoff related data from the database.**

const PassengerSubscription = gql`

subscription PassengerSubscription($userId: String!) {

users\_by\_pk(id: $userId) {

id

company\_id

created\_at

email

name

company {

created\_at

id

is\_active

logo\_url

name

updated\_at

}

is\_driver

phone\_number

phone\_verified

picture

updated\_at

route\_stop\_users {

boarded

confirmed

created\_at

desired\_dropoff

desired\_pickup

desired\_pickup\_address

dropoff\_stop\_id

desired\_dropoff\_address

pickup\_stop\_id

rating\_comment

rating\_score

route\_id

updated\_at

user\_id

dropoff\_stop {

created\_at

done

id

is\_dropoff

location

route\_id

updated\_at

weight

eta

location\_address

}

pickup\_stop {

created\_at

done

id

is\_dropoff

location

route\_id

updated\_at

weight

eta

location\_address

}

route {

arrival\_time

car {

created\_at

id

license\_plate

manufacturer

model

seats

updated\_at

year

}

car\_id

created\_at

current\_location

departed

departure\_time

destination\_area\_id

destination\_area {

id

name

}

driver {

id

name

}

driver\_id

origin\_area {

id

name

}

id

origin\_area\_id

updated\_at

}

}

}

}

`;

1. **This graphql is called when passengers try to find the routes with pickup and dropoff location.**

const FindRoutesByPickupAndDropoffQuery: TypedDocumentNode<

FindRoutesByPickupAndDropoffQueryResponse,

FindRoutesByPickupAndDropoffQueryVariables

> = gql`

query FindPassengerRoute(

$desired\_pickup: geometry!

$desired\_dropoff: geometry!

) {

routes(

where: {

departed: { \_eq: false }

done: { \_eq: false }

origin\_area: {

geography: { \_cast: { geometry: { \_st\_contains: $desired\_pickup } } }

}

destination\_area: {

geography: { \_cast: { geometry: { \_st\_contains: $desired\_dropoff } } }

}

}

order\_by: { departure\_time: asc }

) {

id

departure\_time

arrival\_time

origin\_area {

name

}

destination\_area {

name

}}

}}`

1. **The below graphql is called with OnClick of RESERVE or BOOK RIDE button to reserve the route from passenger side.**

**const ReserveSeatMutation = gql`**

**mutation ReserveSeatMutation(**

**$route\_id: uuid!**

**$desired\_pickup: geography!**

**$desired\_pickup\_address: String!**

**$desired\_dropoff: geography!**

**$desired\_dropoff\_address: String!**

**) {**

**ReserveSeatAction(**

**desired\_dropoff: $desired\_dropoff**

**desired\_dropoff\_address: $desired\_dropoff\_address**

**desired\_pickup: $desired\_pickup**

**desired\_pickup\_address: $desired\_pickup\_address**

**route\_id: $route\_id**

**) {**

**route\_id**

**user\_id**

**}**

**}**

**`;**