horizontal line

Halan

APIs and Documentation

Mobile User APIs :

**Bold is required**

|  |  |  |  |
| --- | --- | --- | --- |
| POST | /user/register | **firstName**  **lastName**  **password**  **confirmPassword**  email  **phoneNumber**  birthday  gender |  |
| POST | /user/login | **phoneNumber**  **password** |  |
| GET | /user/fblogin |  | a phoneNumber should be taken before submitting the login .. in case the user wasn’t registered before .. the login will check first that the phoneNumber is submitted on this first time login |
| POST | /user/updatePassword | oldPassword  newPassword  confirmNewPassword |  |
| POST | /user/forgotPassword | **phoneNumber** | 1st step .. request password change |
| POST | /user/resetPassword | newPassword  confirmNewPassword | 2nd step .. reset the password after requesting the reset in the 1st step |
| POST | /user/update | oldPassword  newPassword  confirmNewPassword  picture (file)  firstName  email  phoneNumber  birthday  gender | update profile including password |

Mobile Driver APIs :

|  |  |  |  |
| --- | --- | --- | --- |
| POST | /driver/register | **phoneNumber**  **password**  **confirmPassword**  email  **birthday**  **firstName**  **lastName** | request to be a driver |
| POST | /driver/login | **phoneNumber**  **password** |  |
| POST | /driver/updatepassword | **newPassword**  **confirmNewPassword**  **oldPassword** | update only password |

Models:

Mongo : (outdated)

User:

*For all kinds of users*

|  |  |
| --- | --- |
| firstName |  |
| lastName |  |
| phoneNumber | unique and identifies user |
| password |  |
| email | optional for users |
| picture | required for drivers |
| registrationDate |  |
| birthday |  |
| gender |  |
|  |  |

*For user mode*

|  |  |
| --- | --- |
| ownReferralCode | code user have to invite people with |
| referredByCode | code user used and was invited by |
| credit |  |
| storedLocations | list of locations user saved |
| allTripsUser | trips user had |
| userStatus | active .. banned .. pending( if we had confirmation first) |

Driver

|  |  |
| --- | --- |
| firstName |  |
| lastName |  |
| password |  |
| email |  |
| picture |  |
| registrationDate |  |
| birthday |  |
| gender |  |
| phoneNumber |  |
| credit |  |
| allTripsDriver | trips driver had |
|  |  |
| locationsVisited | every point the driver is at when they are online..  it has a point and date and type .. the type is whether off, onfree, startTrip, or endTrip  we calculate online time from onfree to startTrip/off .. or from startTrip to endtrip..  when endtrip point is added,new point of type onfree with the current location should be added immediately  we use this to get the number of online time driver is online each day |
|  |  |
| vehicle |  |
| driverLicense |  |
| nationalIdNo |  |
| driverPapers |  |
| address |  |
| driverStatus | pending , rejected, active , suspended  any user is pending by default |

*For operator mode*

|  |  |
| --- | --- |
| isOperator |  |
| operationsTitle |  |
| operatorStatus | active .. suspended |

Trip Request:

|  |  |
| --- | --- |
| user |  |
| from |  |
| to |  |
| date |  |
| driversAround | Drivers recieved the trip request .. they have a state .. whether the driver accepted , rejected or timeoout (didn’t have time to accept ) |
| acceptedDriver |  |

Trip:

|  |  |
| --- | --- |
| user |  |
| driver |  |
| from |  |
| to |  |
| tripRequest |  |
| estimatedCost |  |
| estimatedArrivalTime |  |
| tripStartTime |  |
| waitingStartTime |  |
| cancelTime |  |
| inTripStartTime |  |
| endTime |  |
| cancelledBy |  |
| routePoints | list of locations the trip went through .. each location point has a pointType .. whether it was to the user’s location or in trip |
| commentDriver |  |
| commentUser |  |
| ratingUser |  |
| ratingDriver |  |
| promoCodeUsed |  |
| tripCost |  |
| tripTime |  |

Code:

|  |  |
| --- | --- |
| createdBy |  |
| codeType | whether it’s referral code, promo code , advertisement code .. |
| codeValue | Ahmed45678 |
| description |  |
| startDate |  |
| expiryDate |  |
| discountPercent |  |
| discountAmount |  |
| userOwner | if the card type is referral code |

Driver Papers:

|  |  |
| --- | --- |
| driver |  |
| bloodTest ... |  |
| ... |  |

Vehicle:

|  |  |
| --- | --- |
| driver |  |
| vehicleType | toktok, moto.. |
| model | year |
| make | BMW |
| label | ABC|123 |
| vehicleLicence | {expirationDate and number} |
| vehicleOwner | an optional object with { name , phoneNumber, nationalIdNo } |

Redis

|  |  |
| --- | --- |
| userLocation-userID | location |
| driverLocation-driverID | location |
| userTripState-userID | User Trip State .. if a user in trip cannot request a trip  [inTrip,free] |
| driverTripState-driverID | Driver Trip State .. whether the driver is in trip or available to receive a trip request or in a trip request and cannot receive a trip request  [inTrip,free,locked] |
|  |  |
|  |  |
|  |  |
|  |  |

**Redis stores on heroku :**

|  |  |
| --- | --- |
| **Blue** | **TokenStore** |
| **Maroon** | **StateStore** |
| **Redis** | **Location** |

Dashboard APIs :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method | Route | Input | Description | Extra Notes |
| POST | /api/operator/adddriver | **phoneNumber**  email  **firstName**  **lastName**  **password**  picture : file  birthday : date  gender  address  nationalIdNo | add driver from the dashboard | If the phone number or email is duplicated u will recieve an error |
| POST | /api/operator/getdriverbyphone | **phoneNumber** | get driver |  |
| GET | /api/operator/getalldrivers |  |  |  |
| POST | /api/operator/updatedriver | **driverId**  phoneNumber  email  firstName  lastName  password  picture : file  birthday : date  gender  address  nationalIdNo |  |  |
|  |  |  |  |  |
| POST | /api/operator/changedriverstatus | **driverId**  **newStatus** |  |  |
| POST | /api/operator/deletedriver | **driverId** |  |  |
| POST |  |  |  |  |
| POST | /api/operator/adddriverpapers | **driver**  nationalId  address  drivingLicense  vehicleLicense vehicleOwnerShip criminalRecord  drugTest |  |  |
| POST | /api/operator/deletedriverpapers |  |  |  |
| Get | /api/operator/getdriverpapers |  |  |  |
| POST | /api/operator/adddrivervehicle | **driver** .. |  |  |
| POST | /api/operator/deletedrivervehicle | **driver** |  |  |
| POST | /api/operator/createPromoCode | allowedRedepmtionTimes  **codeType**  **codeValue**  description  **startDate**  **expiryDate**  status  **discountPercent**  **discountAmount**  maximumAmount |  | operator must be logged on and the token must be entered  dates are numbers.. getTime  one of discountPercent  discountAmount is required and both of them will return error message  status is the switch button true/false boolean .. |
| GET | /api/operator/getallpromocodes |  |  |  |

\*\*all request must have ‘application/json’ header

Code flow and Scenarios :

Driver:

When a user opens the driver app can either login as a driver or request to be a driver .. requestToBeDriver is called .. only *firstName , lastName, password, birthday* and *phoneNumber* are required at this stage and no extra info the user should enter when requesting to be a driver .. a user driver will be created with *status* pending if the *phoneNumber* is not used before for another user.. if it existed before it will check if the user was already a driver so the request cannot be done .. otherwise the request goes on but doesn’t take password from the user.

For an operator to add a driver after the recruiting process , addDriver is used.. it checks the *phoneNumber* entered ..if there is no user with this *phoneNumber* then a new User that *isDriver* is created .. if it exists for an existing User which *isDriver* for example a user who s.. the data of this user is updated .. if there is a user that is not *isDriver* .. the data of this user is updated and becomes *isDriver*.

For operators to update driver info , updateDriver is used. It check first that the user phone number is not used anywhere else.

Usually when a driver becomes suspended they don’t get deleted from the database .. only their *driverStatus* is changed to *suspended* .. but in case a driver was created by mistake or or for any other reason, deleteDriver is used to delete a driver permanently.

For operators to change driver’s status between *pending, active, rejected or suspended* , changeDriverStatus is used.

For operators to add papers to a driver .. addDriverPapers is used .. it checks if there is a driverPaper directory already exists then it adds the papers to it otherwise it add a new driverPaper directory then adds it to the User driver.

If the papers directory model takes for example *contract* and *bloodTest* only then the operator wanted to add a new paper type like birthdayCertificate that is not in the driverPaper schema .. it will be added normally.

For operators to delete one or more papers from the user’s papers directory .. deleteDriverPaper is used ..