note:

1. <…> means a redis db.
2. 如中的 ”Rider” 表明该endpoint允许的访问身份为rider。

* **Rider发送一个rider request**

url: /ridingRequests/:email

POST

body:{

startLocation: {},

endLocation: {},

gcm\_token:

}

1. Check if this rider’s email already exists in < ridingRequest >, if yes, goto 1.1, else go to 2.

1.1 Check the rider’s statuy6s, if == WATING, go to 2, else send back err.

2. Add startLocation into <riderStartLocation>;

3. Retrieve the rider’s info from mongodb;

4. Add the rider’s info, startLocation, endLocation into <ridingRequest>. The “status” is WATING, the “driver\_email” is null.

* **Rider取消request**

…..

* **Driver获取附近的riders**

url: /geo/riders

GET

query:{

latitude:

longitude:

radius:

}

Send back nearby riders in <riderStartLocation>;

* **Driver accept一个riding request**

url: /ridingRequests/:email

PUT

body:{

driver\_email:

currentLatitude:

currentLongitude:

}

1. Check the “status” in < ridingRequest >, if != “WATING”, send back err, if yes, continue 2;

2. Delete this rider from <riderStartLocation>;

3. Update the “driver\_email” and “status”(to ACCEPTED) in <ridingRequest>;

4. Add this driver’s geo in <driverCurrentLocation>; (行程结束后要删除他

5. Send gcm message (including the driver’s info) to rider to indicate that a driver has accepted his request.

Exception:

1. The ridingRequest has already been accecpted by another driver. This can be checked out by step 1;

* **Rider获取接送他的driver的坐标**

url :/geo/drivers/:email

GET

1. Check this rider’s “driver\_email” in <ridingRequest>, if it’s equals to :email, continue, otherwise send back err. (此步为检查该rider是否有获取该driver的权限，看看能不能从session里取到该rider的emai

2. Send back the driver’s geo;

* **Driver抵达pick up地点 || Driver抵达终点**

url: /ridingRequests/:email/status

PUT

body:{

status: status

}

1. Check if this driver’s email is equals to the rider’s “driver\_email” in <ridingRequest>, if yes, continue, otherwise send back err.

2. If the body.status == ‘RIDING’ go to 2.1. if the body.status == ‘DONE’ go to 2.2. if neither of two, send back err.

2.1. Check the rider’s “status” in <ridingRequest>, if it == ‘ACCEPTED’, update it to ‘RIDING’ and continue, otherwise send back err.

2.2. Check the rider’s “status” in <ridingRequest>, if it == ‘RIDING’, update it to ‘DONE’ and continue, otherwise send back err.

3. Send a gcm message to rider to indicate that the riding begins.

* **Rider付钱、评分**

url: /ridingRecords

POST

body:{

email:

price:

rating:

}

Add the \*record into mongodb.

Note: The structrure of record: {

rider\_email:

driver\_email:

start\_location:{ }

end\_location:{ }

price:

rating

}

url: /ridingRequest/:email

DELETE

Delete the riding request in <ridingRequest>.