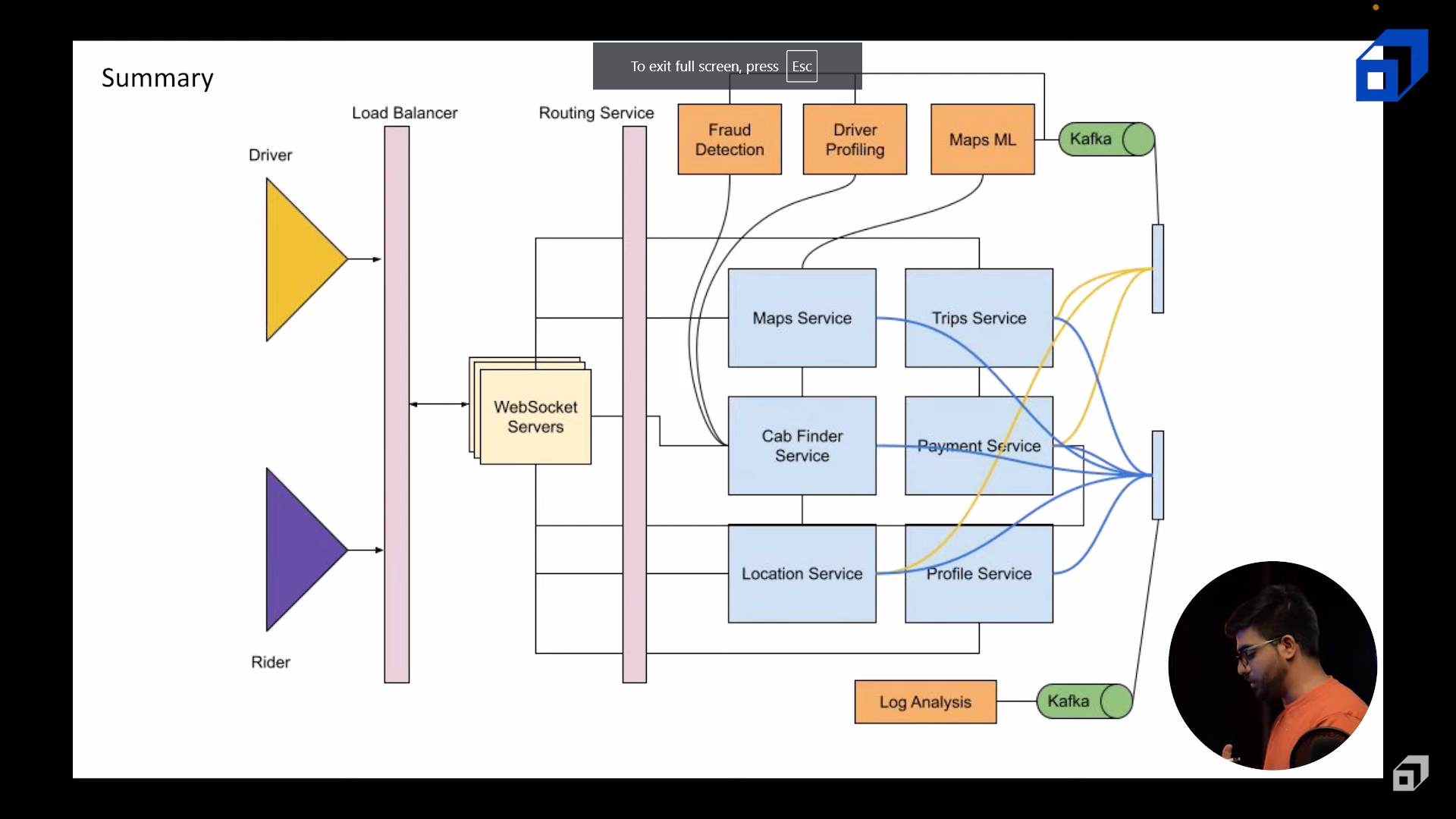
Cab Service System

# Microservice Design

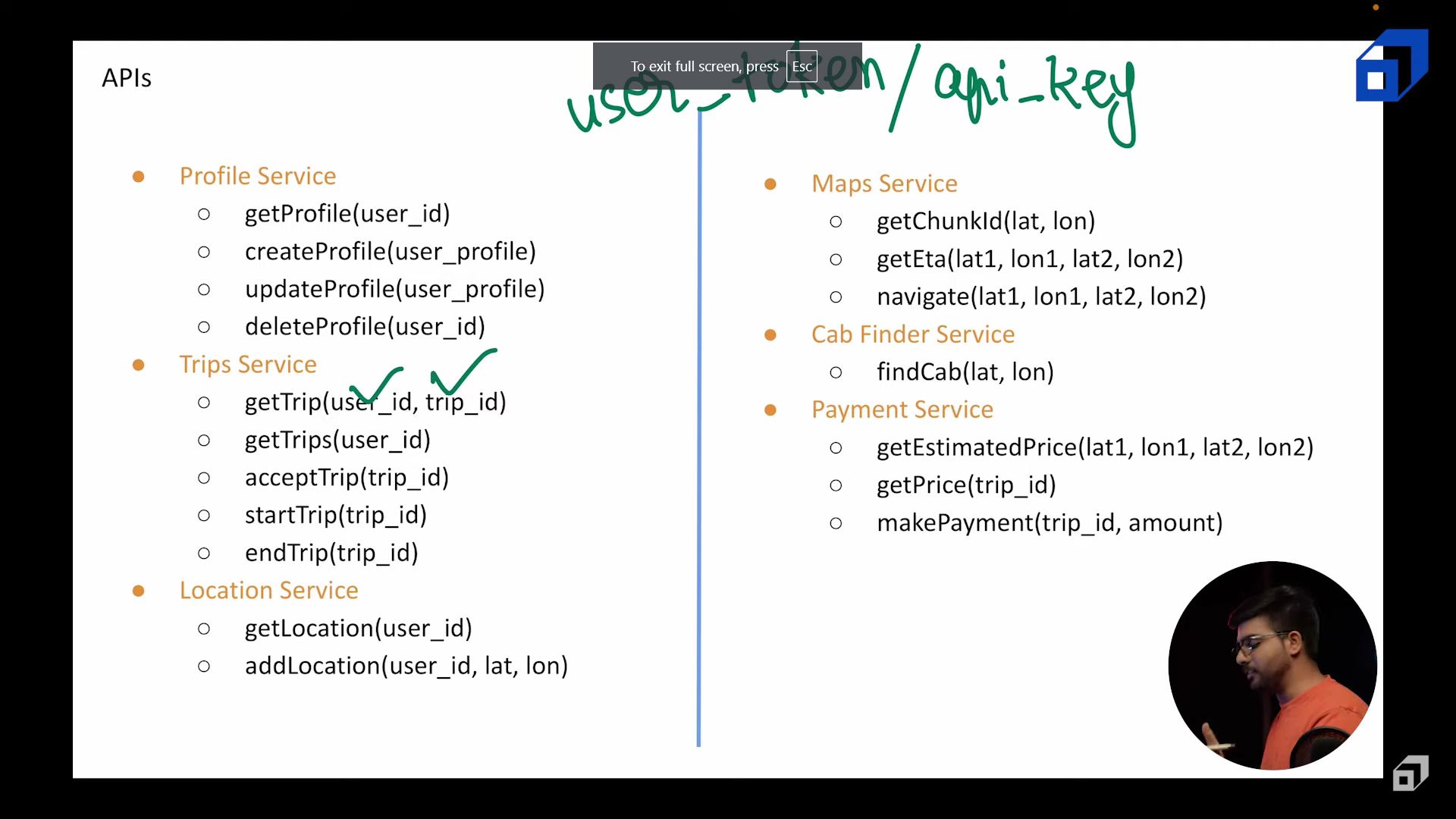
Diagram



# Services

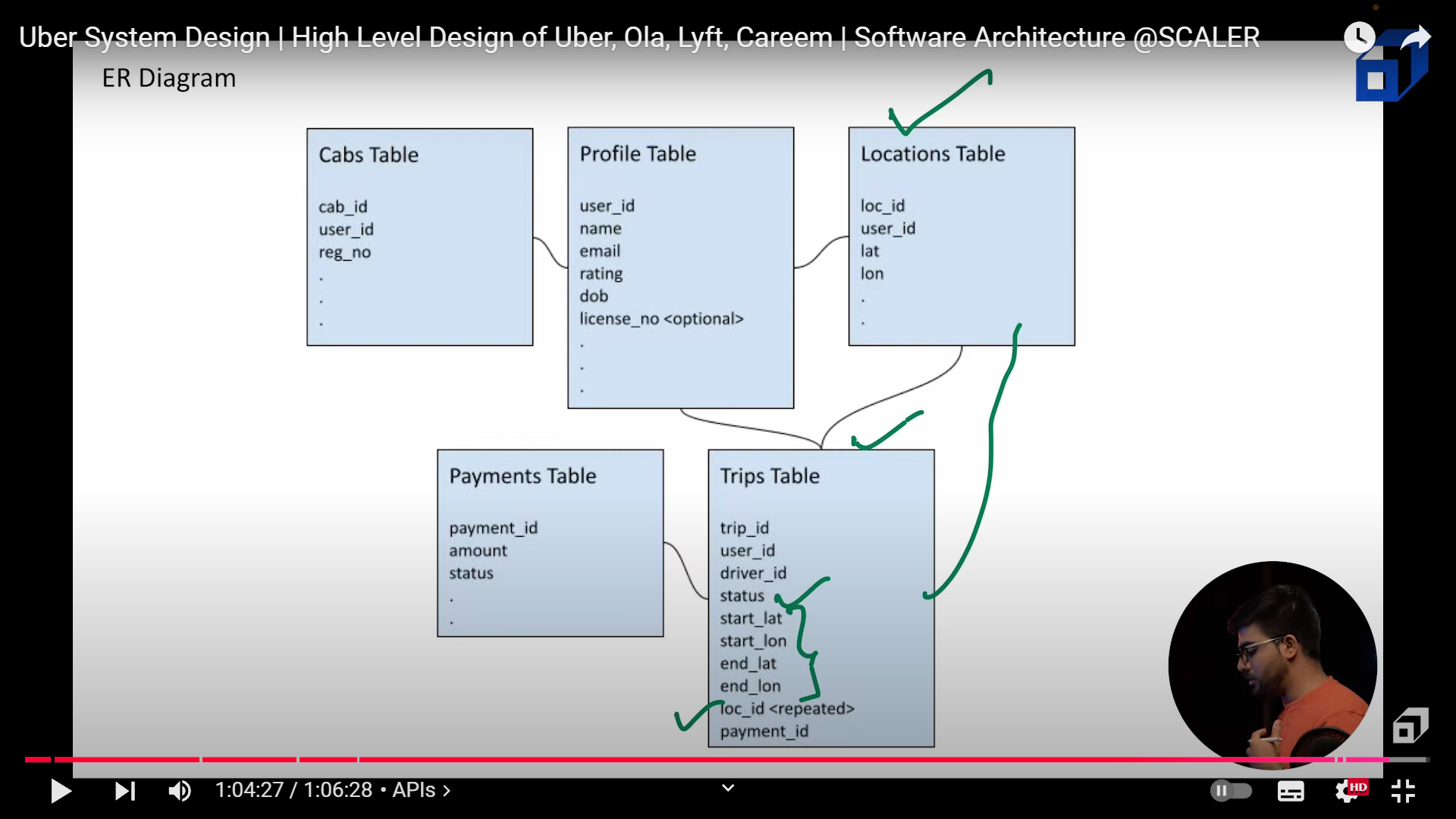
List of Services

1. User-service
2. Map-service
3. Trip-service
4. Cab-finder-service
5. Location-service
6. Payment-service
7. Map-service
   1. Navigate()
      1. localhost:8081/api/v1/map/navigate?sourceLat=49.932707&sourceLon=11.588051&destLat=50.3404&destLon=11.64705&vehicle=car&debug=true&key=cceb673a-91d5-4e84-a9b2-adff379ce38d&type=json&points\_encoded=false
8. Payment-service
   1. getEstimatedPrice()
      1. localhost:8082/api/v1/map/navigate?sourceLat=49.932707&sourceLon=11.588051&destLat=50.3404&destLon=11.64705&vehicle=car&debug=true&key=cceb673a-91d5-4e84-a9b2-adff379ce38d&type=json&points\_encoded=false



# Databases against the services

Databases list



1. User-table
   1. User\_id
   2. Name
   3. Email
   4. role
   5. Rating
   6. Date of birth
   7. License number
   8. Cab\_id
2. **public** **class** User {
4. @Id
5. @GeneratedValue(strategy = GenerationType.***IDENTITY***)
6. **private** Long id;
7. **private** String name;
8. **private** String email;
9. **private** String role;
10. **private** String emailId;
11. **private** String password;
12. **private** **int** Rating;
13. **private** String licenseNumber;
14. **private** Long cabId;
15. }
16. Trip-table
    1. tripId
    2. userId
    3. driver\_id
    4. status
    5. start\_lat
    6. start\_lon
    7. end\_lat
    8. end\_lon
    9. list of location followed
    10. paymentId
17. **public** **class** Trip {
18. // @Id
19. **private** String id;
20. **private** Long userId;
21. **private** Long driverId;
22. **private** Status status;
23. **private** Double start\_latitude;
24. **private** Double end\_latitude;
25. **private** Double start\_longitude;
26. **private** Double end\_longitude;
27. **private** List<String> route;
28. **private** Long paymentId;
30. **public** **enum** Status{
31. ***NOT\_STARTED***, ***IN\_PROGRESS***, ***ENDED***
32. }
33. }
34. Cab-Finder service – cab table
    1. Cab\_id
    2. User\_id
    3. Reg\_no
35. **public** **class** Cab {
36. @Id
37. @GeneratedValue(strategy = GenerationType.***IDENTITY***)
38. **private** Long id;
39. **private** Long userId;
40. **private** String registrationNo;
41. **private** String carModel;
42. **private** vehicleType carType;
43. **private** Status status;
45. **public** **enum** Status{
46. ***AVAILABLE***, ***IN\_A\_RIDE***, ***DISCONNECTED***;
47. }
49. **public** **enum** vehicleType{
50. ***AUTO***, ***CAR***, ***SEDAN***, ***XUV***, ***INTERCITY***;
51. }
52. }

**public** **class** CabDTO {

@Id

**private** String driverId;

**private** Double latitude;

**private** Double longitude;

// add status string var

}

**public** **class** RideRequest {

**private** String userId;

**private** **double** sourceLat;

**private** **double** sourceLon;

**private** **double** destLat;

**private** **double** destLon;

}

1. Location-service- geospatial Coordinate
   1. Latitude
   2. Longitude
   3. Geohash
   4. userId
2. **public** **class** GeoCoordinate {
3. @Id
4. **private** Long userId;
5. **private** Double latitude;
6. **private** Double longitude;
7. **private** String geohash;
8. **private** String role;
9. }

**public** **class** DriverDTO {

**private** Long userId;

**private** Double latitude;

**private** Double longitude;

}

1. Payment-service
   1. paymentId
   2. amount
   3. status
2. **public** **class** PaymentDetails {
3. @Id
4. @GeneratedValue(strategy = GenerationType.***IDENTITY***)
5. **private** Long id;
6. **private** Double amount;
7. **private** String status;
8. **private** Double commissionAmount;

11. }

|  |  |  |  |
| --- | --- | --- | --- |
| Sr No. | Service Name | Server Port |  |
| 1 | Map-service | 8081 |  |
| 2 | Payment-service | 8082 |  |
| 3 | Profile-service | 8083 |  |
| 4 | Api-gateway | 8080 |  |
| 5 | Service-registry | 8761 |  |
| 6 | Trip-service | 8084 |  |
| 7 | Cab finder-service | 8085 |  |
| 8 | Location-service | 8086 |  |

CabDTO

LocationDTO

## Endpoints:

### Map service

spring.cloud.gateway.mvc.routes[0].id=map-service

spring.cloud.gateway.mvc.routes[0].uri=http://localhost:8081

spring.cloud.gateway.mvc.routes[0].predicates[0]=Path=/api/v1/map/\*\*

@RequestMapping("/api/v1/map")

**public** **class** MapController {

@Autowired

MapService mapService;

@GetMapping("/location")

**public** String getLatLon(@RequestParam **double** lat, @RequestParam **double** lng) **throws** IOException {

**return** mapService.getChunkId(lat, lng);

}

@GetMapping("/navigate")

**public** String getRoute(@RequestParam **double** sourceLat,@RequestParam **double** sourceLon,

@RequestParam **double** destLat,

@RequestParam **double** destLon ) **throws** IOException{

**return** mapService.getRoute(sourceLat, sourceLon, destLat, destLon);

}

@GetMapping("/estimated-time-distance")

**public** String getEstimatedTimeAndDistance(@RequestParam **double** sourceLat,@RequestParam **double** sourceLon,

@RequestParam **double** destLat,

@RequestParam **double** destLon ) **throws** IOException {

**return** mapService.getEstimatedTimeAndDistance(sourceLat, sourceLon, destLat, destLon);

}

}

### Payment service

spring.cloud.gateway.mvc.routes[1].id=payment-service

spring.cloud.gateway.mvc.routes[1].uri=http://localhost:8082

spring.cloud.gateway.mvc.routes[1].predicates[0]=Path=/api/v1/payment/\*\*

@RestController

@RequestMapping("/api/v1/payment")

**public** **class** PaymentController {

@Autowired

**private** PaymentDetailsService paymentService;

@Autowired

**private** MapServiceClient mapServiceClient;

@GetMapping("/price/{trip\_id}")

**public** Double getPrice(@PathVariable Long trip\_id) {

System.***out***.println("get price called");

**return** paymentService.getPrice(trip\_id);

}

@GetMapping("/estimate-time")

**public** String getEstimatedPrice(@RequestParam Double sourceLat,Double sourceLon, Double destLat, Double destLon) **throws** IOException {

System.***out***.println("The est price endpoint called");

**return** mapServiceClient.getEstimatedPrice(sourceLat, sourceLon, destLat, destLon);

}

// for make payment endpoint

@PostMapping("/payment-gateway")

**public** String makePayment(@RequestBody String entity) {

//**TODO**: process POST request

**return** entity;

}

}

### Profile service

spring.cloud.gateway.mvc.routes[2].id=profile-service

spring.cloud.gateway.mvc.routes[2].uri=http://localhost:8083

spring.cloud.gateway.mvc.routes[2].predicates[0]=Path=/api/v1/user/\*\*

@RestController

@RequestMapping("/api/v1")

**public** **class** UserController {

@Autowired

UserService userService;

@GetMapping("/user/")

**public** List<User> getAllUsers() {

**return** userService.getAllUsers();

}

@GetMapping("/user/{id}")

**public** Optional<User> getUserById(@PathVariable Long id) {

**return** userService.getUserById(id);

}

@PostMapping("/user/")

**public** String createUser(@RequestBody User user) {

**return** userService.createUser(user);

}

@PutMapping("/user/")

**public** String updateUser(@RequestBody User user) {

**return** userService.updateUser(user);

}

@DeleteMapping("/user/{id}")

**public** String createUser(@PathVariable Long id) {

**return** userService.deleteUser(id);

}

}

### Trip service

spring.cloud.gateway.mvc.routes[3].id=trip-service

spring.cloud.gateway.mvc.routes[3].uri=http://localhost:8084

spring.cloud.gateway.mvc.routes[3].predicates[0]=Path=/api/v1/trip/\*\*

@RestController

@RequestMapping("/api/v1/trip")

**public** **class** TripController {

@Autowired

**private** TripService tripService;

@GetMapping("/")

**public** Trip getTrip(@RequestParam String tripId, @RequestParam Long userID) {

**return** tripService.getTrip(tripId, userID);

}

@GetMapping("/{userID}")

**public** List<Trip> getAllTrip(@PathVariable Long userID) {

**return** tripService.getAllTrips(userID);

}

@PostMapping("/")

**public** String acceptTrip(@RequestParam Long userId ,@RequestBody Trip trip) {

//**TODO**: process POST request

**return** tripService.acceptTrip(trip);

}

}

### Cab finder service

spring.cloud.gateway.mvc.routes[4].id=cab-finder-service

spring.cloud.gateway.mvc.routes[4].uri=http://localhost:8085

spring.cloud.gateway.mvc.routes[4].predicates[0]=Path=/api/v1/ride/\*\*

spring.cloud.gateway.mvc.routes[4].predicates[1]=Path=/api/v1/cab/\*\*

@RestController

@RequestMapping("/api/v1/ride")

**public** **class** RideController {

@Autowired

**private** CabService cabService;

@PostMapping("/book")

**public** String bookRide(@RequestBody RideRequest rideRequest) {

//**TODO**: process POST request

**return** cabService.bookRide(rideRequest);

}

}

### Location service

spring.cloud.gateway.mvc.routes[5].id=location-service

spring.cloud.gateway.mvc.routes[5].uri=http://localhost:8086

spring.cloud.gateway.mvc.routes[5].predicates[0]=Path=/api/v1/location/\*\*

@RestController

@RequestMapping("/api/v1/location")

**public** **class** LocationController {

@Autowired

**private** LocationService locationService;

@PostMapping("/save")

**public** ResponseEntity<Void> saveLocation(@RequestParam Long userId,

@RequestParam **double** latitude,

@RequestParam **double** longitude) {

locationService.saveLocation( userId,latitude, longitude);

**return** ResponseEntity.*ok*().build();

}

@GetMapping("/{userId}")

**public** ResponseEntity<GeoCoordinate> getLocation(@PathVariable String userId) {

GeoCoordinate location = locationService.getLocation(userId);

**return** ResponseEntity.*ok*(location);

}

@GetMapping("/nearby")

**public** String getNearbyLocations(@RequestParam **double** latitude,

@RequestParam **double** longitude,

@RequestParam **double** radius) {

String nearbyLocations = locationService.getNearByLocations(latitude, longitude, radius);

**return** nearbyLocations;

}

}