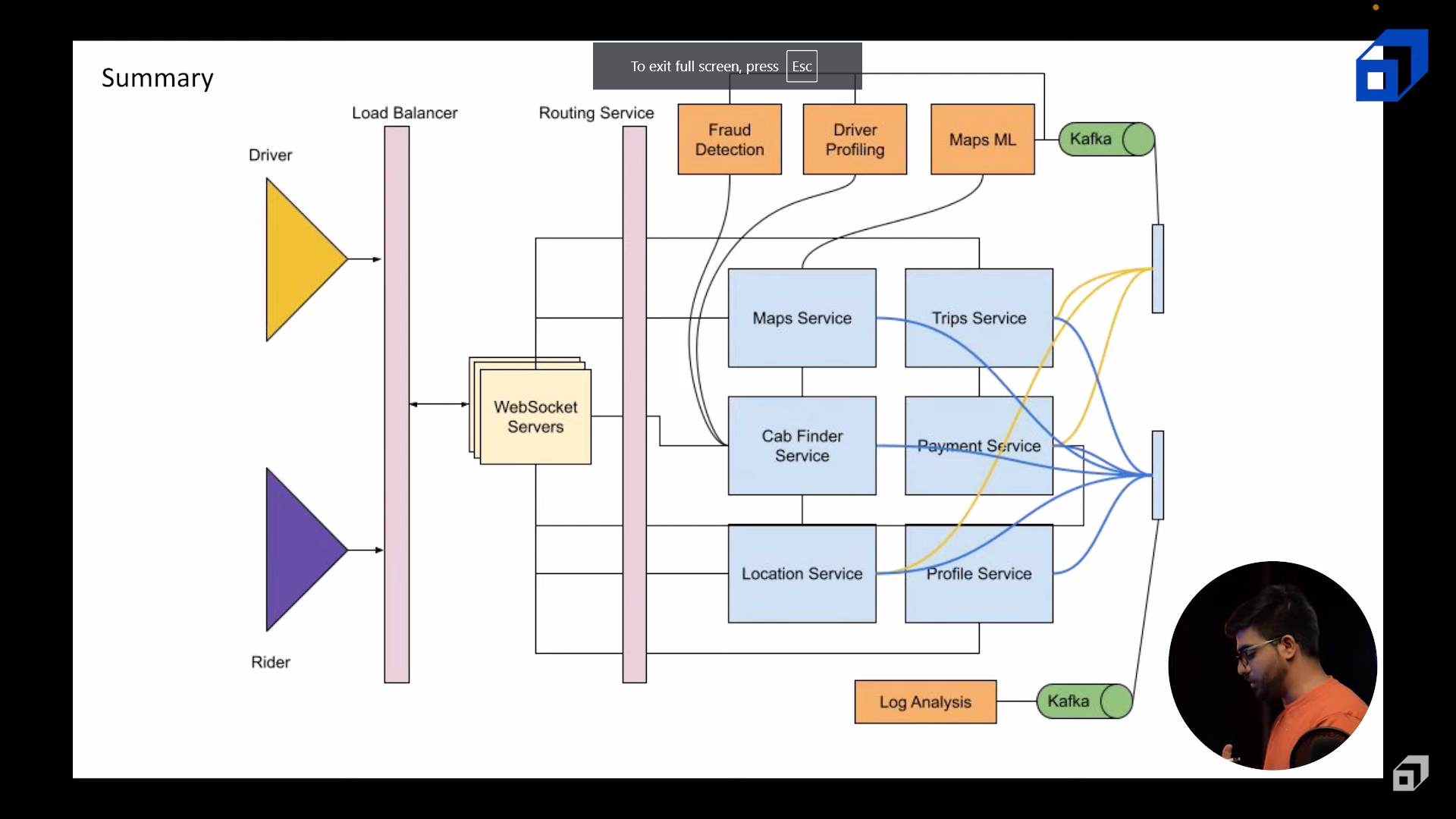
Cab Service System

# Microservice Design

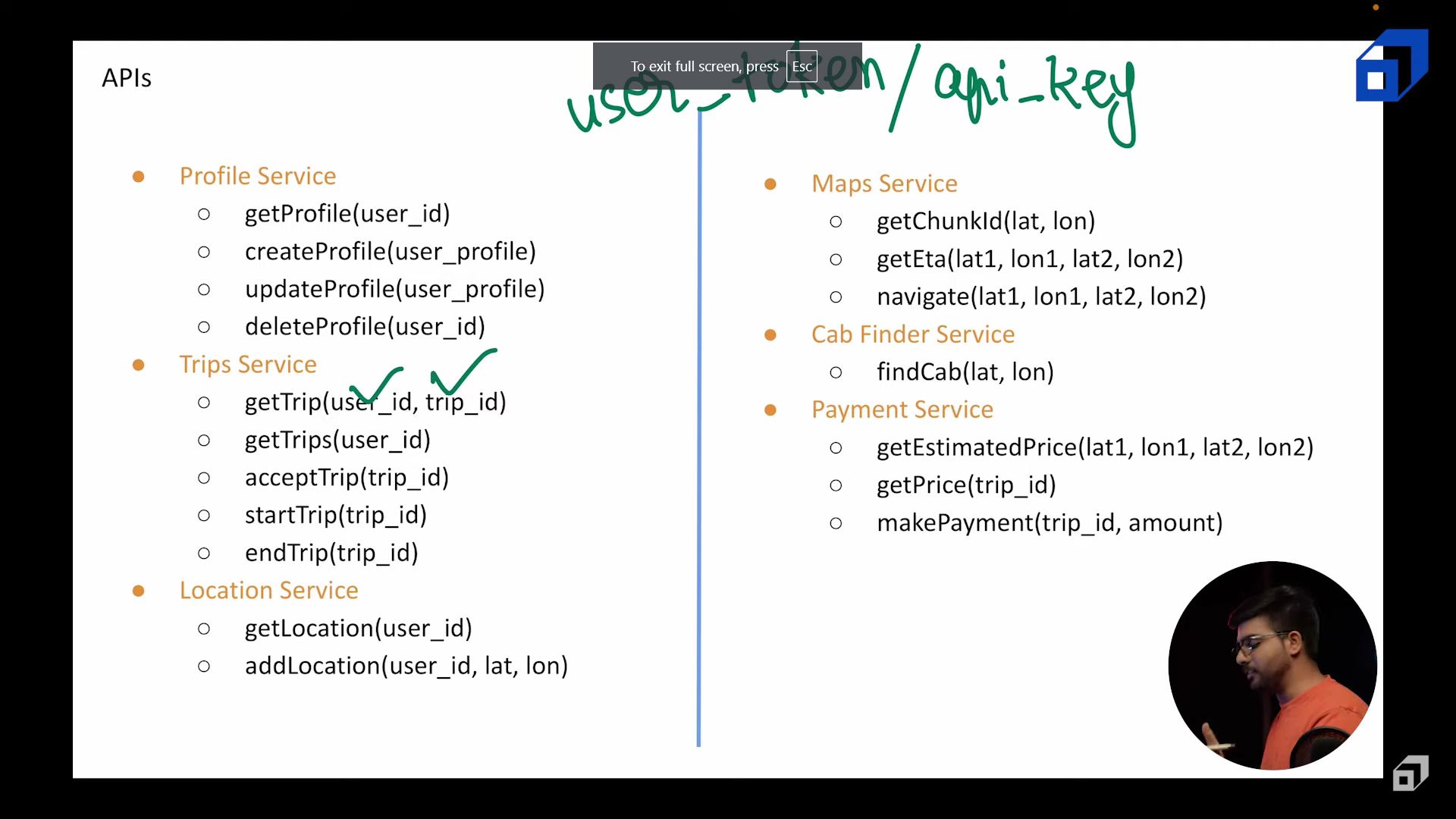
Diagram



# Services

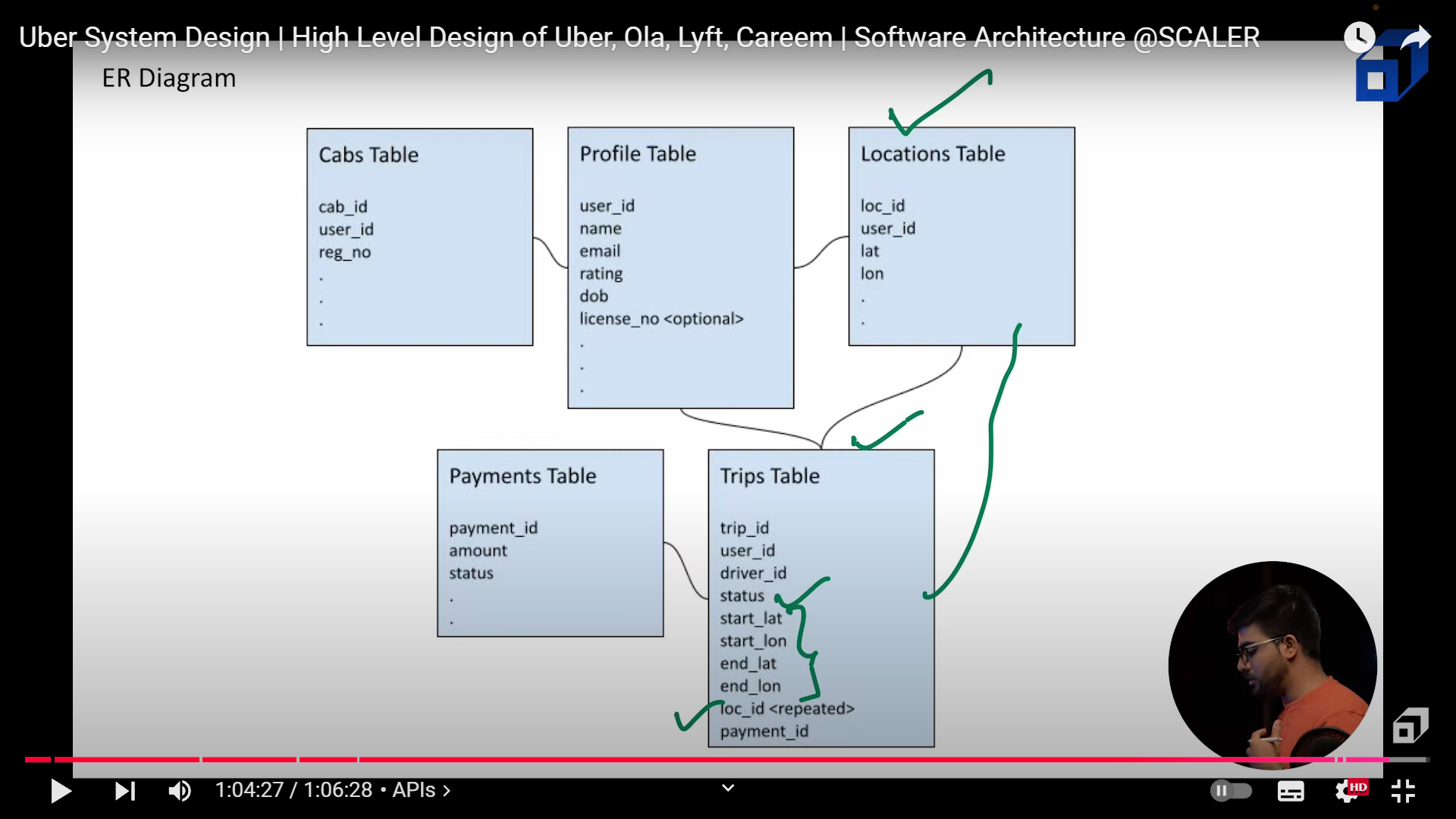
List of Services

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. 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
3. Cab-Finder service – cab table
   1. Cab\_id
   2. User\_id
   3. Reg\_no
4. Location-service- geospatial Coordinate
   1. Latitude
   2. Longitude
   3. Geohash
   4. userId
5. Payment-service
   1. paymentId
   2. amount
   3. status

|  |  |  |  |
| --- | --- | --- | --- |
| 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/\*\*

### 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/\*\*

### 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/\*\*

### 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;

}

}