

# Test task for Java developer "IN-house navigation system"

## Functional description

- There are X ( $X < 100$ ) base stations (BS) and Y ( $Y < 100$ ) mobile stations (MS).
- Base stations can detect the presence of mobile stations in a certain radius (detectionRadiusInMeters).
- When detected, MS id and timestamp are reported by BS to RestEndpoint1 (see below)
- One MS can be reported multiple times by multiple BSs.
- Data should be saved to relational database (in-memory db is fine for this assignment)
- MS position can be queried from RestEndpoint2 (see below).
- RestEndpoint2 should be mapped to `/location/{uuid}`, where uuid is MS id.
- RestEndpoint2 should correctly handle errors and situations where the information is not available.

## Technical guidelines

Major components should be covered with unit tests.

The system must be designed and implemented using following technologies / libraries:

- Spring Boot
- Java 8

## Examples and hints

**Base station** can be described with the following set of properties:

```
{
  "id": uuid,
  "name": string,
  "x": float,
  "y": float,
  "detectionRadiusInMeters": float
}
```

**Mobile station** can be described with the following set of properties:

```
{
  "id": uuid,
  "lastKnownX": float,
  "lastKnownY": float
}
```

**RestEndpoint1** message example:

```
{
  "base_station_id": uuid,
  "reports": [
    {"mobile_station_id": uuid, "distance": float, "timestamp": timestamp},
    {"mobile_station_id": uuid, "distance": float, "timestamp": timestamp},
  ]
}
```

**RestEndpoint2** response example:

```
{
  "mobileId": uuid,
  "x": float,
  "y": float,
  "error_radius": float,
  "error_code": integer,
  "error_description": string
}
```