

Case Assignment

Senior Backend Engineer

aidn

Introduction

As part of the recruitment process for the position of **Senior Backend Engineer**, we ask you to complete a case assignment to better understand your professional expertise and thought-processes in areas that we see as important to succeed and be happy at Aidn.

The case is designed such that there is no one Right Answer [™]. Rather, the point of the exercise is to learn more about how you approach problems and communicate your ideas. Feel free to make any assumptions necessary to complete the assignment—just make sure you tell us about them!

Please be prepared to give a walkthrough of your code and a brief demo of the working solution. Your audience will be comprised of engineers, so you may use as much (or as little) technical language as you wish. We look forward to hearing you draw parallels to your experience and listen to how you use your knowledge, experience, and intuition.

The conversation will be conducted in either **English or Norwegian**, depending on the preferences of the participants.

We have reserved **60** minutes for the interview, where **45** minutes are allocated for the presentation and discussion of your technical solution and **15** minutes are allocated for follow-up questions from both sides.

Case Description

Nursing home patients are at considerable risk for rapid deterioration. To identify important changes in condition as soon as possible, workers perform physical assessments at regular intervals to determine when extra monitoring is required to keep a patient stable.

During an assessment, several measurements are feed into an algorithm that produces a single numeric value called a NEWS score. Any score above a certain threshold results in escalated monitoring. This simple heuristic is one of the most important clinical activities happening on any ward where long-term care is provided.

Your assignment is to create an API endpoint—using the programming language of your choice— that accepts a defined set of physical measurements and returns the appropriate NEWS score for the values given. The endpoint should also verify that the input data falls within reasonable limits to trap any errors due to mistyping.

You may coordinate the transfer of input data to your endpoint in any way you prefer (e.g., Postman, CLI, web form, etc). The focus of the assignment will be the endpoint and the underlying calculation service, not the UI layer.

Please see the following pages for more details.

Specification

In this simplified example, all input **measurements** are provided as a type and a value. There are three types of measurements that must be provided to calculate a score—body temperature (**TEMP**), heart rate (**HR**), and respiratory rate (**RR**). Input measurements should specify the type as a capitalised enumeration and the value as an integer.

For each measurement, your program should calculate an **individual score** by evaluating the input value against a defined set of ranges. All starting values are exclusive; all ending values are inclusive. Values outside of the defined ranges are invalid.

The three individual scores are then summed to produce a final **NEWS score**.

TEMP Range	Score
31..35	3
35..36	1
36..38	0
38..39	1
39..42	2

HR Range	Score
25..40	3
40..50	1
50..90	0
90..110	1
110..130	2
130..220	3

RR Range	Score
3..8	3
8..11	1
11..20	0
20..24	2
24..60	3

Example

Given the following input values:

```
{  
  measurements: [  
    { type: "TEMP", value: 37 },  
    { type: "HR", value: 60 },  
    { type: "RR", value: 5 },  
  ]  
}
```

The endpoint should return:

```
{ score: 3 }
```