

PragmaBrewery Code Challenge

Meet Shane. He works at PragmaBrewery, a boutique micro brewery based in a rural area of Australia and creator of 6 unique beer varieties. Shane is responsible for driving the large transport truck, delivering goods from the brewery to a number of pubs across metropolitan Sydney each week.

Each beer has its own specific refrigeration needs while being transported:

- | | |
|--------------------------------|-----------------------------------|
| - Beer 1 (Pilsner) - 4°C - 6°C | - Beer 4 (Stout) - 6°C - 8°C |
| - Beer 2 (IPA) - 5°C - 6°C | - Beer 5 (Wheat beer) - 3°C - 5°C |
| - Beer 3 (Lager) - 4°C - 7°C | - Beer 6 (Pale Ale) - 4°C - 6°C |

The refrigerated truck is loaded with multiple containers with beer bottles inside, each set to a specific temperature and each containing a thermometer sensor.

While driving, Shane is alerted if any of the containers fall outside of the temperature range. Unfortunately, this is common due to factors such as opening the doors to unload, the heat of the Sydney summer or sometimes due to forgetting the container doors ajar.

Instructions

Develop a solution that allows Shane to be aware of the current temperature of each container and notifies him when the temperatures are outside the correct range.

1. We would prefer that you complete this code challenge using JavaScript technology for at least part of the solution. If you have strong reasons to use another technology to better demonstrate your engineering skills, please have a chat about it with us first and add that explanation in your *readme* file.
2. Use your time to deliver a solution that showcases your coding skills and the level of quality you expect (but no need to gold plate it).
3. In a real-life scenario, you would ask questions to clarify any doubts but for this challenge, document the questions you would ask and provide your own answers.
4. The solution is not required to use a database server, if needed mock the data you will need internally in any of the application layers.
5. The solution must be implemented with an acceptable level of automated tests.
6. Once complete, send us a link where we can download the package from,
7. Make sure your package contains a *readme* file with any relevant information necessary to run your project, including:
 - What are the highlights of your logic/code writing style?
 - What could have been done in a better way? What would you do in version 2.0?
 - What were the questions you would ask and your own answers/assumptions?
 - Any other notes you feel relevant for the evaluation of your solution.

Thank you :)

Thanks for taking the time to complete the code challenge.
We look forward to receiving your solution!