# Angular - Test Task

## Technical questions

- What kind of OS do you use?
- Which version of Angular do you use?
- Do you have experience in writing unit test with Jest?

# Task:

Create a SPA application that receives large amounts of data with high frequency. The data should be received in the web-worker from a pseudo-socket and passed to the main thread. The main thread should render the incoming data in the amount of the last 10 elements.

# Requirements:

1. Make a pseudo-socket through a timer. Take into account the ability to change the value of the timer interval through the UI (input), in ms
2. The size of the data array coming from the pseudo-socket can be adjusted via the UI (input)
3. A single array element is an object that has the following fields:
    1. **id** - string field
    2. **int** - integer field
    3. **float** - float field (precision === 18)
    4. **color** - string field with color name (can be in any representation rgb, hex, string)
    5. **child** - field which is an object that has two fields - **id** and **color**
4. Only the last 10 elements will be displayed in the UI and may contain their own set of ids, which can be set in the UI. For example, 1000 items come from the thread. Of these, 10 elements are selected for display as specified in the task. If `additional_ids` are set, the IDs are overwritten for the first elements of these 10 elements.
5. In the main thread, you need to use not raw objects, but classes that are created based on the models described in step 3
6. In the UI, it is required to display data in the following representation:
    1. each element is a table row
    2. fields are columns
    3. child is a nested table

When rendering the color column, it is necessary to fill its background with the color specified in the field

# Technologies and libraries:

1. angular 15
2. web-worker
3. jest || jasmine+karma

# The main points when checking the test:

1. compliance with the specified requirements
2. use of specified technologies and libraries
3. performance (under different given conditions)
4. decomposition of code entities
5. writing unit-tests (jest is preferable)

# Will be a plus:

1. Using design patterns

| Timer, ms | Array size | Additional array ids |
|-----------|------------|----------------------|
| 300 | 1000 | 956, 998, 970 |

| id | int | float | color | child |
|----|-----|-------|-------|-------|
| 956 | 12768464 | 23.457633... | red | id: 974, color: green |
| 960 | 3767585 | 13.874644... | blue | id: 976, color: purple |
| 970 | 98373 | 0.493839... | yellow | id: 977, color: orange |

...