

Front-end assessment: Item selector with folders

The goal of this assessment is to get an indication of your skill and knowledge, and to provide a discussion topic for the next interview round.

The expected deliverables for this assessment are:

- Github link to your code
- A clear readme with instructions on how to run the project. Make sure you also specify the right node version to use.

We expect this assessment to not take more than 3 hours of your time. If the task could not be completed within this timeframe this is not an issue. Please state so in the Readme so we can take this into account.

The assignment

For this assignment you're expected to create an interactive item selector with folder structure. Below is a screenshot that illustrates the deliverable. Further down below you will find a Figma design link with the exact design details.

The screenshot shows a web-based item selector interface. It features a list of items organized into a folder structure. The items are: Audio (expanded), Speakers (expanded), Active Speakers (expanded), Active Speakers Item 1, Passive Speakers (expanded), Passive Speakers Item 1, Speaker Item 1 (checked), Speaker Item 1 (checked), Audio Item 1, Rigging (checked), Truss (checked), Truss Item 1 (checked), and Truss Item 2 (checked). A mouse cursor is hovering over 'Active Speakers Item 1'. At the bottom, it displays 'Selected item IDs: 3, 5, 6' and a 'Clear selection' button.

☒ Audio

^

☒ Speakers

^

☐ Active Speakers

^

☐ Active Speakers Item 1

☐ Passive Speakers

^

☐ Passive Speakers Item 1

☒ Speaker Item 1

☒ Speaker Item 1

☐ Audio Item 1

☒ Rigging

^

☒ Truss

☒ Truss Item 1

☒ Truss Item 2

Selected item IDs: 3, 5, 6

Clear selection

Requirements

The requirements are as follows:

- The project is written in Typescript and uses proper types everywhere
- The program is modeled like you would in a large-scale application
- The project follows the design
- An external component library (such as MaterialUI) is **not** used
- The “response.json” file contains the back-end data. Retrieve the data as you would do with an external API according to Angular best practices.
 - Do not directly import it using Javascript imports
- The folders from the response are used to render a list. The folders contain a “parent” field which defines the three-dimensional structure
- Folders and item rows should be sorted alphabetically, with folder rows coming first
- All folders should be expanded initially
- Every row has a hover state
- A folder row should have a toggle button to show and hide its children
- An item row should have a checkbox to toggle between the selected state
 - Selecting *some* of the items in a folder row should show the indeterminate state on all parent folder rows
 - This can be seen in the **Speakers** and **Audio** folders in the Figma file
 - Select *all* items in a folder row should show the selected state on the direct parent folder
 - This can be seen in the **Truss** and **Rigging** folders in the Figma file
- Clicking on a folder row should select toggle all items inside of it
 - If the folder row was in the selected state all items should be deselected
 - If the folder row was unselected, or in the indeterminate, state all items should be selected
- The selected item IDs are outputted on the screen
 - No styling needed for this list
 - Should be outside of the selector component
- A “Clear selection” button is present which clears the selection
 - Should be outside of the selector component

On a day-to-day basis you will use Angular. Using this framework is not a requirement, but it is preferred. If you feel you can show your skills better using a different framework this is allowed.

Designs

Note: For the checkboxes the system’s default checkboxes are allowed.

[Link to design on Figma](#)

Data

See included response.json file