Challenge 6

By: Mosh Hamedani

Filtering, Sorting and Pagination

1- Add a new page at **/vehicles** and display the list of vehicles in a table. The table should have 5 columns:

- Id
- Make
- Model
- Contact Name
- View (add a link to take the user to the vehicle form)

2- Add a new menu in the navigation bar called **Vehicles.** Change the route configuration so when the user navigates to **localhost:5000**, they're automatically redirected to **localhost:5000/vehicles**.

3- Add a section for filtering on top of this table. This section should include only one drop-down list for the vehicle makes. Upon selecting a make, the user should only see the vehicles with the given make. Filtering should happen on the server. But as an exercise, try implementing the filtering on the client first.

4- Add sorting to **Make, Model** and **Contact Name** columns. The **Id** and **View** columns should not be sortable. When the user clicks on any of these columns, a sort icon should appear and the data should be sorted in an ascending order. When the same column is clicked again, the sort order should be reversed. Use Font Awesome library for rendering the sort icons. Sorting should also happen on the server.

5- Add pagination below the vehicles table. Set the page size to 3 so you don't have to create a lot of vehicles. Render the pages based on the total number of vehicles. Add buttons to navigate forward and backward. If there are less vehicles than the page size, the pagination should disappear.

Resources

To implement filtering, sorting and pagination on the server, you need to use LINQ. Dynamically construct the query based on the given parameters. If you need to brush up your LINQ skills, watch the section called **Querying Data using LINQ** in my **Entity Framework** course:

Entity Framework in Depth: The Complete Guide