

Automating Checking - Phase 2 - TodoMVC challenge part 2 write up

The plan we created was mostly made up of simple tests that would be automated across the different frameworks

Below is a list of each of the sections for this challenge

- automate now
- automate later
- dont automate

and

- end to end
- integration
- unit

Areas

Automate now

- **Add a new ToDo item**
 - Can't add an item with an empty value
 - Can add a value with a single character
 - Check that every single accented character and symbol is supported
 - Check that every single emoji character is supported
- **Modify a ToDo item** (by double-clicking)
- **If you modify a ToDo item and click Escape during edit, it should cancel the modification**
- **A ToDo item can be ticked-off** (it will appear with a line through it)

- A completed ToDo item can be unticked again
- Delete an incomplete ToDo item
 - Delete a completed ToDo item
- Status bar displays "0 items left" when there are no items left
 - Status bar displays "1 item left" when there is 1 item left
 - Status bar displays "2 items left" when there are 2 items left
 - Status bar displays "3 items left" when there are 3 items left
 - Status bar is hidden when there are no ToDo items in the system
- When there are any completed items, a "Clear completed" link appears in the status bar
 - When the "Clear completed" link is clicked, all completed items are deleted
 - Status bar is hidden when there are no ToDo items in the system
- Clicking the down arrow symbol next to the "What needs to be done?" box will toggle all items to Completed or Not Completed
- ToDo items from one variant do not "bleed over" into other applications (e.g if you create a ToDo item in the React version, it is *not* visible in the Vue.js version)

Reasoning

- Increase efficiency in testing all the frameworks
- short time and manually testing them all would be difficult
- Time saving relative
- maintainability
- relatively simple tests for experienced staff to run

automate later

- A ToDo item can be reordered by dragging it up or down in the list

Reasoning

- a key feature to maintain

- but - currently not working (is bugged)

dont automate

- ToDo items have a 248 Character limit
- clicking on the links on the pages take you where expected (front page only)

Reasoning

- removed the 248 limit
- link clicking we can just do manually - only need to test a few links on the front page

and

integration

end to end and unit

- ToDo items from one variant do not "bleed over" into other applications (e.g if you create a ToDo item in the React version, it is *not* visible in the Vue.js version)
- **Add a new ToDo item**
 - Can't add an item with an empty value
 - Can add a value with a single character
 - Check that every single accented character and symbol is supported
 - Check that every single emoji character is supported
- **Modify a ToDo item** (by double-clicking)
- **If you modify a ToDo item and click Escape during edit, it should cancel the modification**
- **A ToDo item can be ticked-off** (it will appear with a line through it)
- A completed ToDo item can be unticked again

- A ToDo item can be reordered by dragging it up or down in the list
- Delete an incomplete ToDo item
 - Delete a completed ToDo item
- Status bar displays "0 items left" when there are no items left
 - Status bar displays "1 item left" when there is 1 item left
 - Status bar displays "2 items left" when there are 2 items left
 - Status bar displays "3 items left" when there are 3 items left
 - Status bar is hidden when there are no ToDo items in the system
- When there are any completed items, a "Clear completed" link appears in the status bar
 - When the "Clear completed" link is clicked, all completed items are deleted
 - Status bar is hidden when there are no ToDo items in the system
- Clicking the down arrow symbol next to the "What needs to be done?" box will toggle all items to Completed or Not Completed

Reasoning

- Lots are in unit test as we felt the tests would be testing one factor - the program
 - by default this could be considered an end-to-end test as it tests the user side as well as all the code up until that point
-