

MANAGING STATE IN MODERN APPLICATIONS





Introduction

Defining state is seemingly very complex, and arbitrary, but a common definition of this can be that "state management makes the state of your app tangible in the form of a data structure that you can read from and write to". It is imperative to manage state effectively, especially now that we enjoy full, rich, interactive web applications that are much more complex than web applications in the past.

Self-taught developer and Senior UI Architect at IHS Markit, Dillon Erhardt, joined our Trusted Tech Talks online event "State of Frontend" to discuss how to manage state in frontend modern applications, and how decisions can impact the overall software project delivery cycle.

Different Types of State

Dillon split States into three types:

UI State

• form, data fetching, layout

Navigation State

where you are in the application - lives in the URL bar

Business State

 Most complex part and makes up the majority of the state. Most of the issues therefore happen here, and this generally lives on the backend



Challenges & Solutions

Common Issues & Available Solutions

Problems arise as we move towards that single page application model. Devs on frontend have to also think about the state of the application. When developers are creating web applications, the frontend needs to display the editor and content, have authorisation and log in. But what often happens, is that the developers will implement changes and updates in the frontend, rather than syncing from the backend, which causes a fracture between the backend and the frontend.

State management solutions include software such as Redux which has a lot of implementations across different frameworks. Redux is a great tool, but it's also too flexible and often leaves too much up to the developer. Having all the states centralised and in one place can cause developers to get confused by having the UI and Navigation state all in the same place. A common outcome of this is that the Frontend shouldn't be the main source of truth. The Frontend shouldn't be the main source of truth with the backend catching up. The backend should be the source of truth.

Getting this process wrong has a wider impact than you'd think. Developing a State Model which has a frontend that holds all the state, and a backend that out of sync, causes a number of issues outside the development itself. e.g. story splitting, architect decisions, team structure, and testing. These stages and actions become arbitrary and blurred as vertical chunking becomes impossible.

Tooling is Key

Leaders need to make their teams' life easier by picking tools that are suitable for their organisation, and there is no one size fits all. Redux can be a great option is people know how to use it effectively. In order to do this, it's important to hire employees who are experienced in managing state and can use their experience to influence and support the wider team.



Contact Us

Build high performing teams through exceptional staffing solutions and business advice with Maxwell Bond, the recruitment partner of choice across the UK and Germany.



Jordan Matthews Frontend Software Consultant jordan.matthews@maxwellbond.co.uk 0161 359 3260