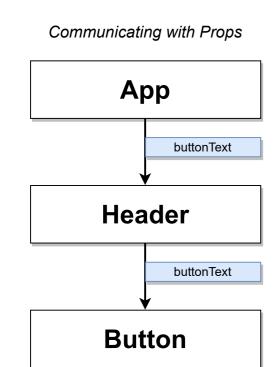
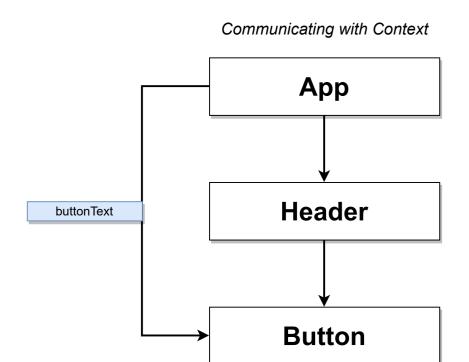
Props System

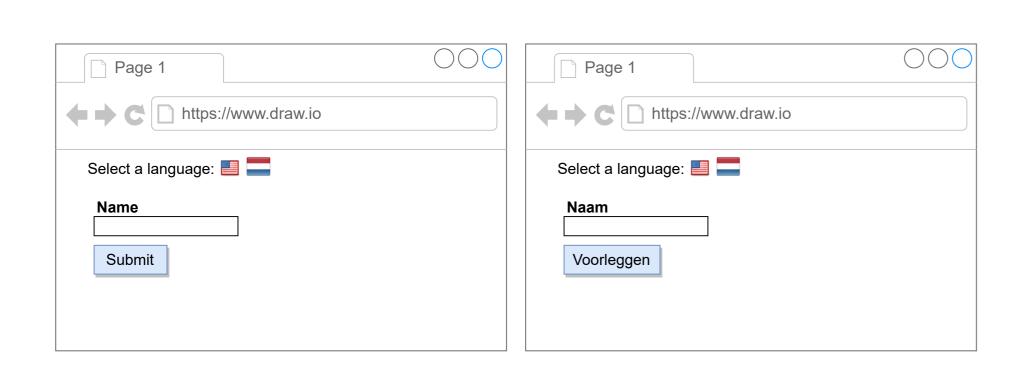
**Context System** 

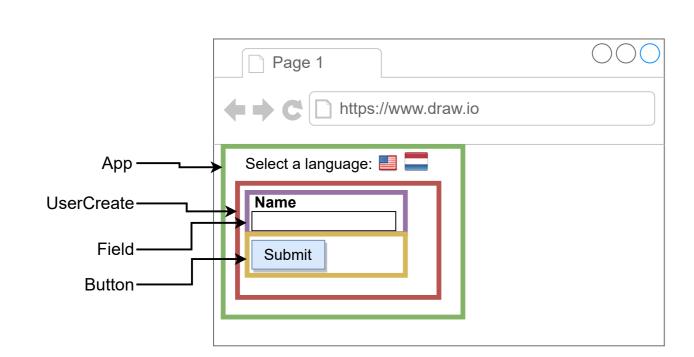
Gets data from a parent component to a *direct* child component

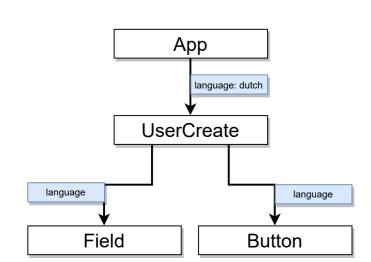
Gets data from a parent component to any nested child component

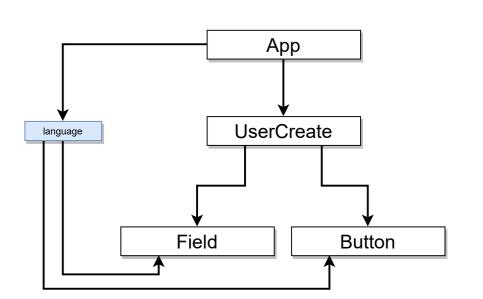


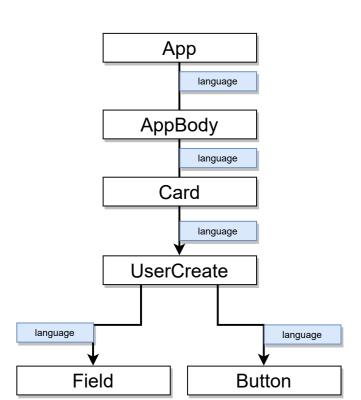


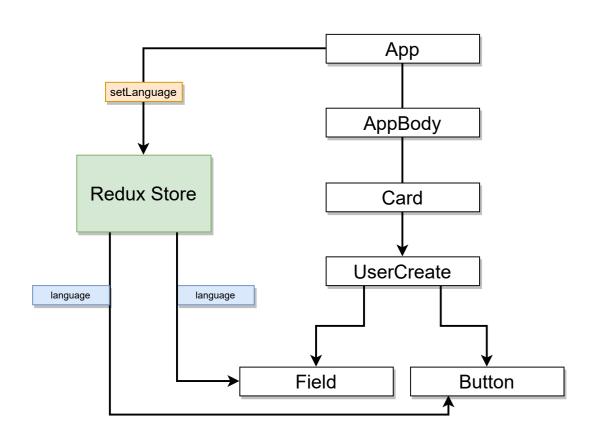


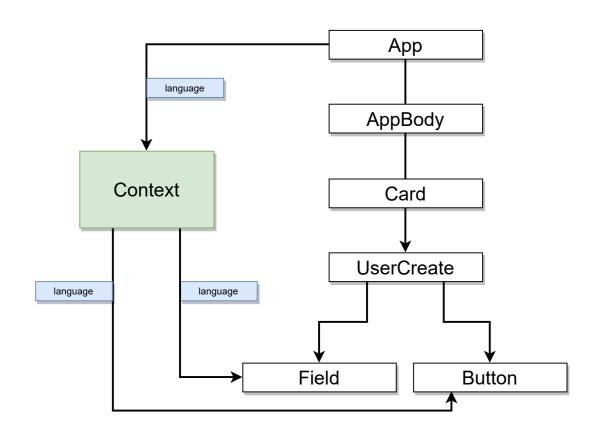












Redux

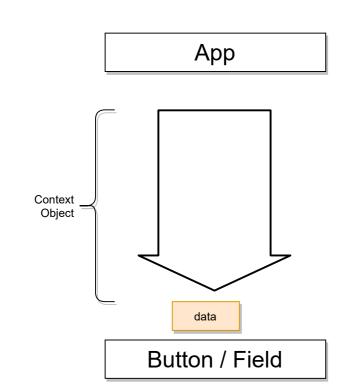
Context

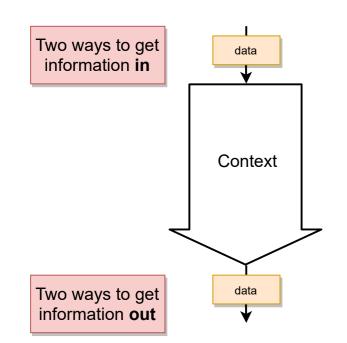
Distributes data to various components

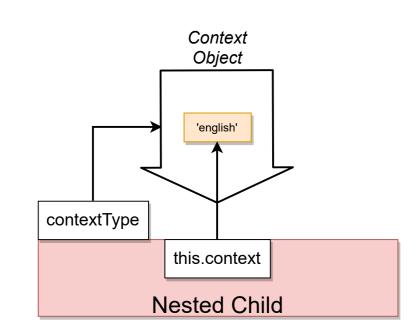
Distributes data to various components

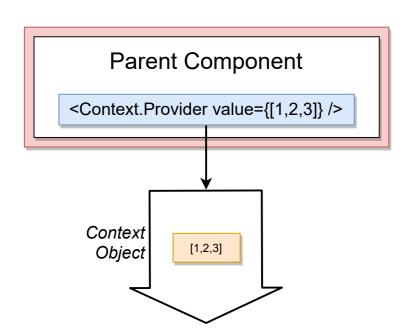
Centralizes data in a store

KIND OF maintain data









Application loads up in the browser

We create a context object with a default value of 'english'

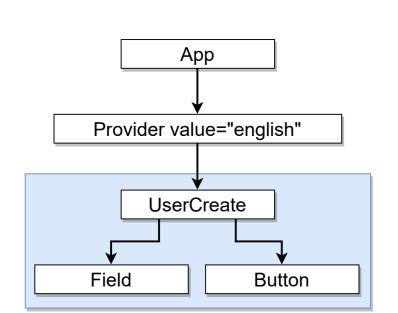
App component gets rendered, creates a Provider that wraps UserCreate

Provider updates the value of the context object to 'this.state.language'

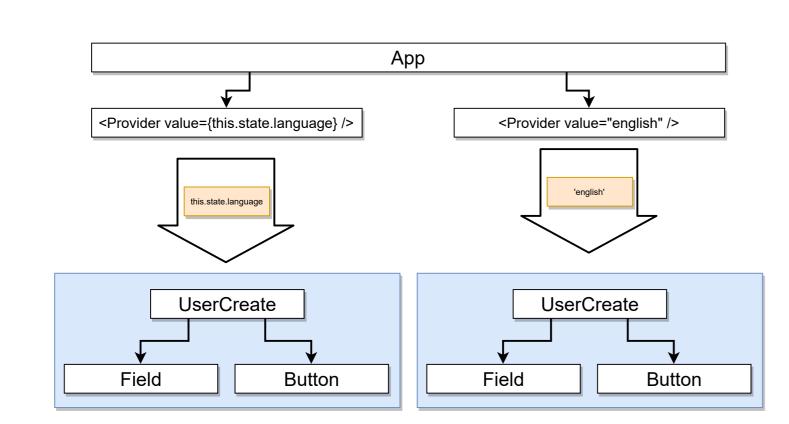
Button and Field reach into context object, see the value from

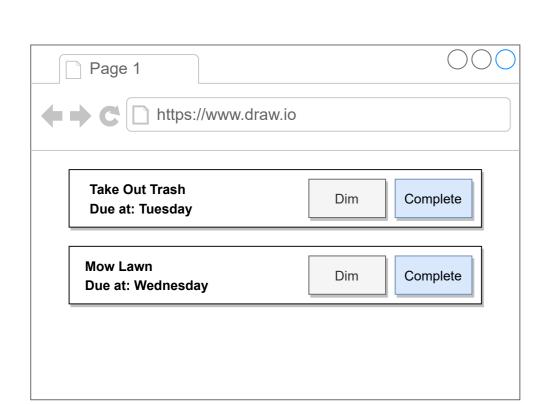
'this.state.language'

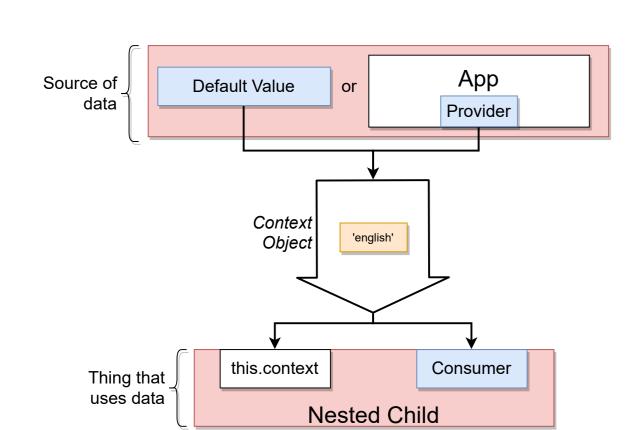
Button and Field render appropriate text to the screen

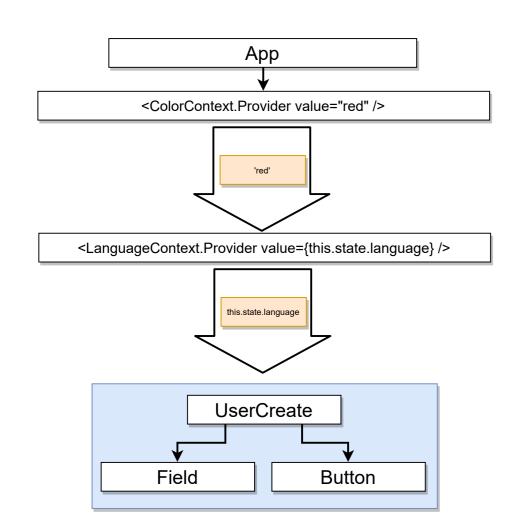


Each separate use of LanguageContext.Provider creates a new, separate 'pipe' of information!

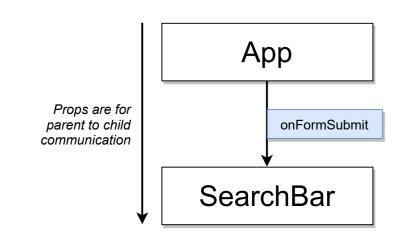








From the video app earlier in the course



Redux vs

Context

Distributes data to various components

Centralizes data in a store

Provides mechanism for changing data in the store

Distributes data to various

components

Context

