

ReactJS

...

xyz

The Single Immutable State Tree (El Árbol de Estado Único Inmutable)

Un JSON inmutable con el estado de los datos de la app.



Describing State Changes with Actions (Describiendo Cambios de Estado con Acciones)

- State immutable.
- Cualquier cambio al estado es necesario hacerlo despachando una acción.
- Action: JSON describiendo la acción.
- State: mínima representación de la data en la app.
- Action: mínima representación del cambio en los datos.
- Action: debe tener una propiedad "type", de resto cualquier estructura.

The Reducer Function (La Función Reducer)

La aproximación de Redux: La UI o la capa de vista es más predecible cuando es descrita como una función pura del state de la app.

Redux: Los states en la aplicación son descritos como una función pura que recibe un state previo y el action despachado, y retorna el siguiente state de la app.

Cualquier app Redux cuenta con una función que hace esto: el reducer.

Mutar el state de la app.



Pure and Impure functions (Funciones Puras e Impuras)

Puras:

- Función que retorna un valor dependiendo sólo de los argumentos.
- No tienen efectos laterales observables como llamados a red o bases de datos.
- Con el mismo conjunto de argumentos se obtiene el mismo valor.
- Son predecibles.
- No modifican el valor pasado, no sobrescriben, sino que devuelven un nuevo objeto.

Impuras:

- Podrían tener efectos laterales como llamados a red o bases de datos.
- Podrían sobrescribir los valores pasados.



Nomenclature/Conventions (Nomenclatura/Convenciones)

- Presentational Components (Componentes de Presentación)
- Container Components (Componentes Contenedor)
 - Los *containers* delegan la presentación visual a la función `render()` de React.

ES6 syntax (sintaxis ES6)

concat()

() => {} (function)

slice()

() => () (function)

spread operator (...)

const {} = Object (destructuring)

Object.assign()

let

object literal shorthand notation

const

reduce()



ES6 syntax (sintaxis ES6)

```
import React from 'react';

class Hello extends React.Component {

  render() {

    return <h1>Hello</h1>

  }

}
```

// Mejor forma de definir elementos.
Más escalabilidad para componentes
complejos.

```
var React = require('react');

var Hello = React.createClass({displayName:
  'Hello',

  render: function() {

    return React.createElement("h1", null,
    "Hello ");

  }

});

// Componente React Hello sin ES6 y JSX.
```



References (Referencias)

- <https://egghead.io/series/getting-started-with-redux> .
- <https://github.com/lukehoban/es6features> .
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference> .

GRACIAS

...

