UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Programador Web Avanzado

Clase N° 13: React JS

Profesor: Ing. Leandro Rodolfo Gil Carrano
Email: leangilutn@gmail.com

UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

React Js

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

¿Que es?

ReactJS es una librería Javascript de código abierto, desarrollada por facebook. Esta librería es utilizada en algunas de sus plataformas, por ejemplo Instagram. Esta librería, a pesar de tener detalles algo polémicos, ofrece grandes beneficios en performance, modularidad y promueve un flujo muy claro de datos y eventos, facilitando la planeación y desarrollo de apps complejas.

ReactJS es una librería enfocada en la visualización. Si estamos iniciando un proyecto podemos basarnos en la arquitectura Flux, pero si ya tenemos un proyecto usando un Framework MVC como AngularJS podemos dejar AngularJS como Controlador y que ReactJS se encargue de las vistas. Esto tiene sentido pues ReactJS tiene un performance superior al momento para manipular el DOM, y esto tiene un gran impacto cuando se trata con listas largas que cambian constantemente en nuestra visualización.

¿Cómo funciona?

React.js está construido en torno a hacer funciones, que toman las actualizaciones de estado de la página y que se traduzcan en una representación virtual de la página resultante. Siempre que React es informado de un cambio de estado, vuelve a ejecutar esas funciones para determinar una nueva representación virtual de la página, a continuación, se traduce automáticamente ese resultado en los cambios del DOM necesarios para reflejar la nueva presentación de la página.

¿Cómo funciona?

A primera vista, esto suena como que fuera más lento que el enfoque JavaScript habitual de actualización de cada elemento, según sea necesario. Detrás de escena, sin embargo, React.js hace justamente eso: tiene un algoritmo muy eficiente para determinar las diferencias entre la representación virtual de la página actual y la nueva. A partir de esas diferencias, hace el conjunto mínimo de cambios necesarios en el DOM.

Pues utiliza un concepto llamado el DOM virtual que hace selectivamente sub-árboles de los nodos sobre la base de cambios de estado, desarrollando esto, con la menor cantidad de manipulación DOM posible, con el fin de mantener los componentes actualizados, estructurando sus datos.

Características

- **JSX**: Es una extensión sintáctica de javascript, si bien su uso no es obligatorio con react js es recomendable
- **Components**: En react js debemos pensar todo como un componente, de esta manera podremos lograr tener aplicaciones escalables y de fácil mantenimiento.
- **Flujo de datos unidireccional**: A diferencia de los frameworks modernos, el flujo de datos en react js es unidireccional. Esto lo hace más fácil para debugear y mantener
- **Licencia**: Esta licenciado bajo “the Facebook Inc. Documentation is licensed under CC BY 4.0.”

UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Virtual DOM

Virtual DOM

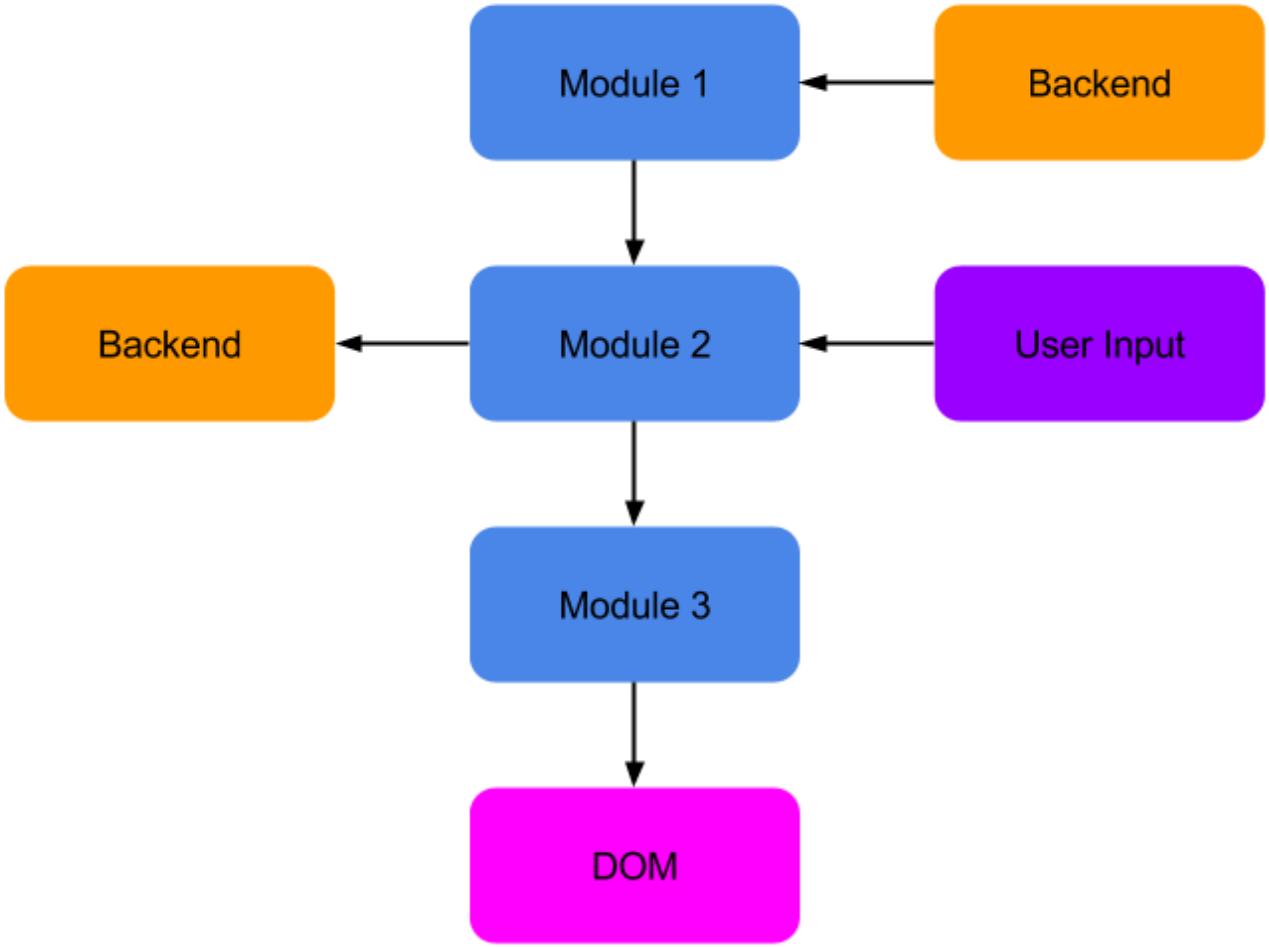
El secreto de ReactJS para tener un performance muy alto, es que implementa algo llamado Virtual DOM y en vez de renderizar todo el DOM en cada cambio, que es lo que normalmente se hace, este hace los cambios en una copia en memoria y después usa un algoritmo para comparar las propiedades de la copia en memoria con las de la versión del DOM y así aplicar cambios exclusivamente en las partes que varían. Esto puede sonar como mucho trabajo, pero en la práctica es mucho más eficiente que el método tradicional pues si tenemos una lista de dos mil elementos en la interfaz y ocurren diez cambios, es más eficiente aplicar diez cambios, ubicar los componentes que tuvieron un cambio en sus propiedades y renderizar estos diez elementos, que aplicar diez cambios y renderizar dos mil elementos.

Virtual DOM

Son más pasos a planear y programar, pero ofrece una mejor experiencia de usuario y una planeación muy lineal. Una característica importante de ReactJS es que promueve el flujo de datos en un solo sentido, en lugar del flujo bidireccional típico en Frameworks modernos, esto hace más fácil la planeación y detección de errores en aplicaciones complejas, en las que el flujo de información puede llegar a ser muy complejo, dando lugar a errores difíciles de ubicar.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Virtual DOM



Virtual DOM - ¿Cómo funciona?

Imagina que tienes un objeto que es un modelo en torno a una persona. Tienes todas las propiedades relevantes de una persona que podría tener, y refleja el estado actual de la persona. Esto es básicamente lo que React hace con el DOM.

Ahora piensa, si tomamos ese objeto y le hacemos algunos cambios. Se ha añadido un bigote, unos bíceps y otros cambios. En React, cuando aplicamos estos cambios, dos cosas ocurren:

En primer lugar, React ejecuta un algoritmo de “diffing”, que identifica lo que ha cambiado.

Virtual DOM - ¿Cómo funciona?

El segundo paso es la reconciliación, donde se actualiza el DOM con los resultados de diff.

La que hace React, ante estos cambios, en lugar de tomar a la persona real y reconstruirla desde cero, sólo cambiaría la cara y los brazos. Esto significa que si usted tenía el texto en una entrada y una actualización se llevó a cabo, siempre y cuando nodo padre de la entrada no estaba programado para la actualización, el texto se quedaría sin ser cambiado.

Virtual DOM - Ventajas

- Performance con DOM Virtual
- Flujo unidireccional de datos y eventos a través de una jerarquía de componentes modulares con punto único de entrada (programación reactiva)
- Binding unidireccional entre el view (vista) y modelo, se evita complejidad y explosión de eventos difíciles para debugging.
- Se usa realmente por sus creadores

Virtual DOM - Ventajas

- Se integra directamente con implementaciones del paradigma también reactivo de gestión de datos Flux (como Redux) que maneja los datos de la aplicación en una división también jerárquica de componentes contenedores vinculados al estado de los datos en el store, y que pasan estos datos y funciones gestores de esos datos a componentes puros que demuestran los datos en forma predecible sin efectos secundarios y ofrecen puntos de interacción al usuario.
- Es fácil incluir rendering (proyección del componente en el DOM virtual como nodos de elementos con estilo y comportamiento, o sea HTML + CSS + JavaScript) en el lado del servidor con funciones sencillas, para lograr las llamadas aplicaciones universales (antes se llamaban isomórficas).

CLI

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

React JS – CLI Instalación

Debemos ejecutar el comando **npm install -g create-react-app**

```
G:\sites\react>npm install -g create-react-app
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

React JS – Crear una aplicación

Debemos ejecutar **create-react-app my-app**

```
G:\sites\react>create-react-app my-app

Creating a new React app in G:\sites\react\my-app.
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

React JS – Ejecutar aplicación en el navegador

Para ejecutar una aplicación y poder acceder desde el navegador debemos ejecutar (dentro del directorio de la aplicación creada)
npm start

```
G:\sites\react\utn>npm start

> utn@0.1.0 start G:\sites\react\utn
> react-scripts start
Starting the development server...
Compiled successfully!

You can now view utn in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://192.168.0.11:3000/

Note that the development build is not optimized.
To create a production build, use npm run build.
```

UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

JSX

JSX - ¿Qué es?

JSX es una extensión de JavaScript creada por Facebook para el uso con su librería React. Sirve de preprocesador (como Sass o Stylus a CSS) y transforma el código a JavaScript.

Te puede parecer que estás mezclando código HTML dentro de tus ficheros JavaScript, pero nada más lejos de la realidad. A continuación te lo explico.

React al basar el desarrollo de apps en componentes, necesitamos crear elementos HTML que definan nuestro componente, por ejemplo , <p>, <img>.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

JSX – Ejemplo sin JSX

Crear un componente sin utilizar JSX sería algo como esto

```
var image = React.createElement('img', {
  src: 'react-icon.png',
  className: 'icon-image'
});

var container = React.createElement('div', {
  className: 'icon-container'
}, image);

var icon = React.createElement('Icon', {
className: 'avatarContainer'
}, container);

ReactDOM.render(
  icon,
  document.getElementById('app')
);
```

JSX – Ejemplo sin JSX

Obteniendo el siguiente resultado:

```
<div class='icon-container'>
  <img src='icon-react.png' class='icon-image' />
</div>
```

```
.icon-image {
  width: 100px
}
.icon-container {
  background-color: #222;
  width: 100px
}
```

JSX – Ejemplo con JSX

```
var Icon = (
  <div className='icon-container'>
    <img
      src='icon-react.png'
      className='icon-image'
    />
  </div>)

ReactDOM.render(Icon, document.getElementById('app'))
```