

HENRY

A bright yellow beam of light originates from the left edge of the frame and points towards the letter 'R' in the word 'HENRY'. The beam is wider on the left and tapers as it moves to the right. The word 'HENRY' is rendered in a bold, black, sans-serif font.

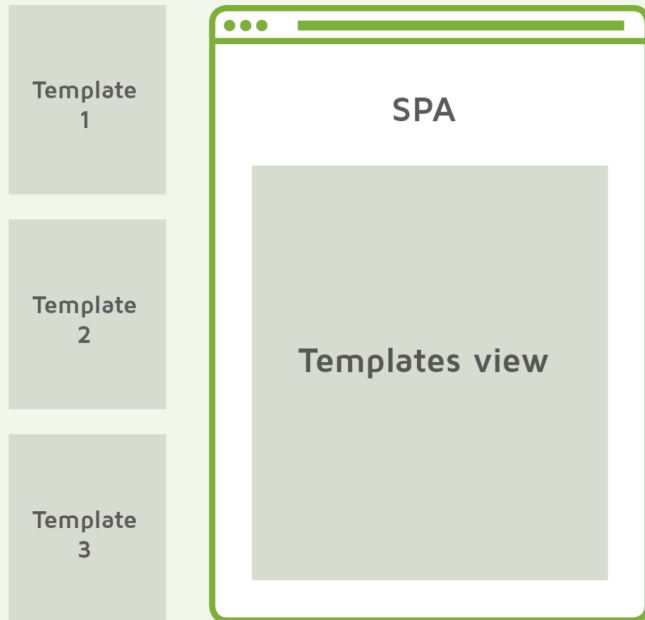


ROUTING



SPA

Single Page Application



No page refresh on request

Traditional Web Application



Whole page refresh on request



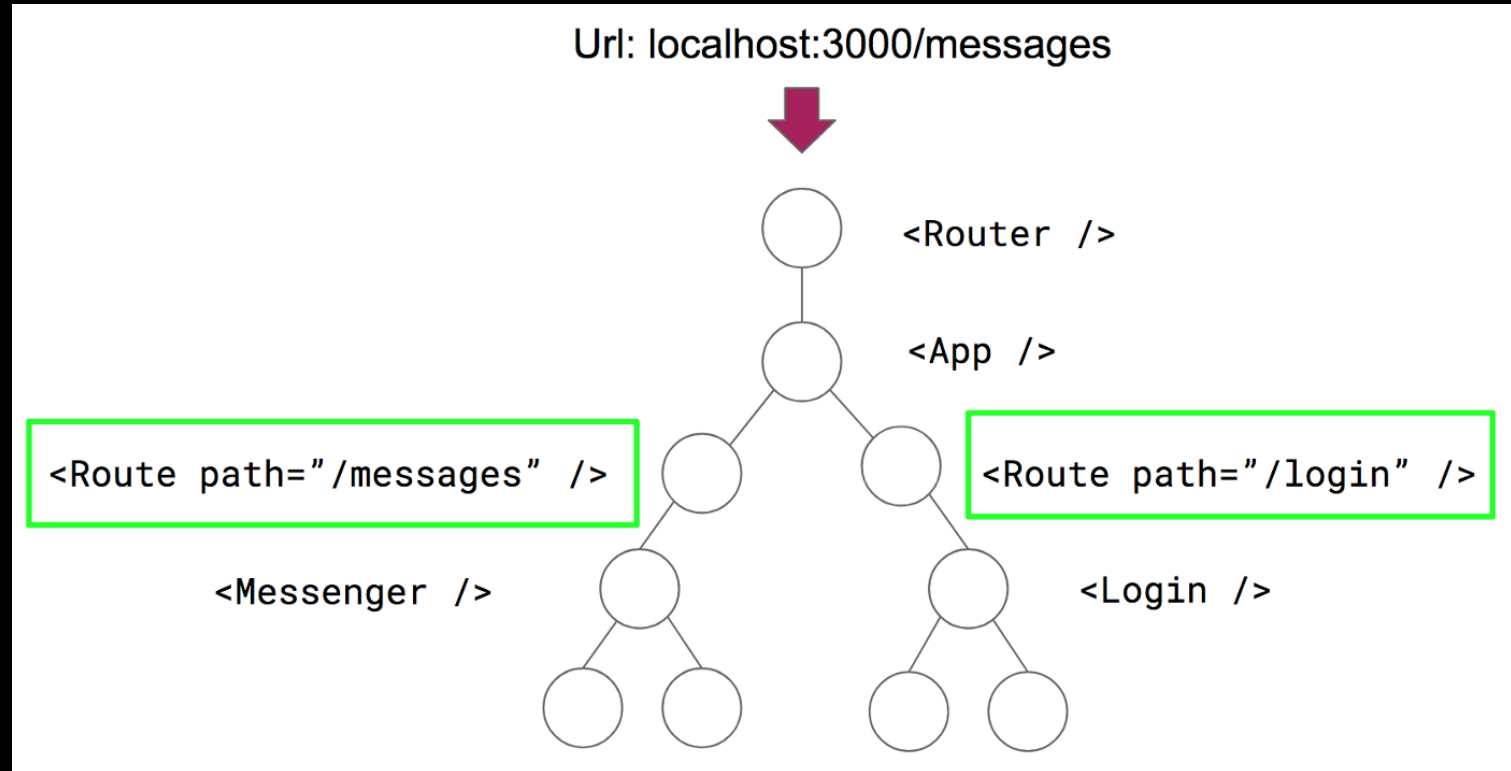
SPA



Librería para definir de forma declarativa la vistas que queremos renderizar dependiendo la URL

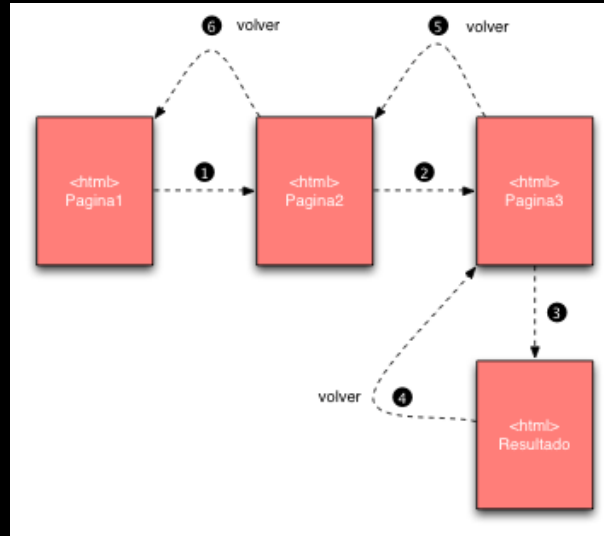


SPA - Routing



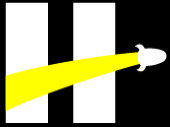


SPA - History API



El objeto DOM [window](#) proporciona acceso al historial del navegador a través del objeto [history](#) . Este da acceso a métodos y propiedades útiles que permiten avanzar y retroceder a través del historial del usuario, así como manipular el contenido del historial.

[Documentación API](#)



SPA - Routing

Para configurar nuestra aplicación primero tenemos que decidir que tipo de router vamos a usar:

`<BrowserRouter />`

`http://example.com/about`

Usa el API de history, no es legacy.

`<HashRouter />`

`http://example.com/#/about`

Legacy, compatible con todos.

```
1 // Instalar React-Router-DOM
2
3 npm i -S react-router-dom
4
5
6 // App.js
7
8
9 import { HashRouter, Route } from 'react-router-dom';
10
11
12 ReactDOM.render((
13   <HashRouter>
14     <Route path="/" component={App}/>
15   </HashRouter>
16 ), document.getElementById('app'))
```



SPA - Routing

```
1 import React from 'react';
2 import { render } from 'react-dom';
3 import { Route, Switch, NavLink, HashRouter as Router } from 'react-router-dom';
4
5 function Nav() {
6   return (
7     <div>
8       <ul>
9         <li><NavLink to="/" exact activeClassName= 'active'>Home</NavLink></li>
10        <li><NavLink to="/about" activeClassName= 'active'>About</NavLink></li>
11        <li><NavLink to="/ejemplo" activeClassName= 'active'>Ejemplo</NavLink></li>
12      </ul>
13    </div>
14  );
15 };
16
17 const Root = (
18   <Router>
19     <Route path="/" component={Nav} />
20     <Route exact path="/" component={Home} />
21     <Route exact path="/about" component={About} />
22     <Route exact path="/ejemplo" component={Ejemplo} />
23   </Router>
24 );
25
26 render(Root, document.querySelector('#app'));
```




< Demo />



<Route />

Props:

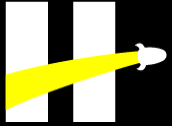
- **component:** Toma un componente y lo crea usando `React.createClass`
- **render:** Toma una función que retorna UI - Útil para hacer inline render y pasar Props.
- **children:** Similar a render, pero siempre renderiza.

path

El path es usado para identificar la porción de la url que debería coincidir con la ubicación actual.

Usa la librería **Path-to-RegExp** para convertir un string del path a una expresión regular.

Si el path y la ubicación actual coinciden se crea un objeto `match` el cual se pasa como prop del componente renderizado.

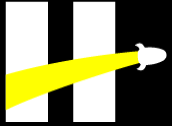


<Route />

{ match, location and history }

- **match.url:** Retorna la porción matcheada del url.
- **match.path:** El string del path de la ruta útil para agregar rutas anidadas.
- **match.isExact:** Es true si la ruta matchea exactamente.
- **match.params:** Objeto con los valores de los parametros de la ruta

```
1
2 <Route
3   path="/:id"
4   render={({match, location, history}) => (
5     <p>Ejemplo</p>
6   )}
7 />
```



< Demo />