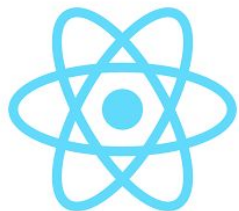


Sharing React Components

2017/02

Renato Matos



Agenda



1. Introdução ao uso dos componentes
2. Setup ambientes
3. Criando o componente Estados
4. Usando o componente Estados
5. Adicionando eventos ao componente
6. Criando o componente Cidades
7. Usando os componentes Estados e Cidades
8. Minificando a aplicação

Para que componentizar?



Vantagens da componentização

- Reutilização - Back e Front
- Padronização
- Simplificação na Manutenção
- Redução dos arquivos
- Especialização de funcionalidades
- Facilidade na Distribuição
- Equipes especializadas - CORE e CUSTOM (Front e Back)
- Baixo impacto para distribuição

Onde podemos componentizar



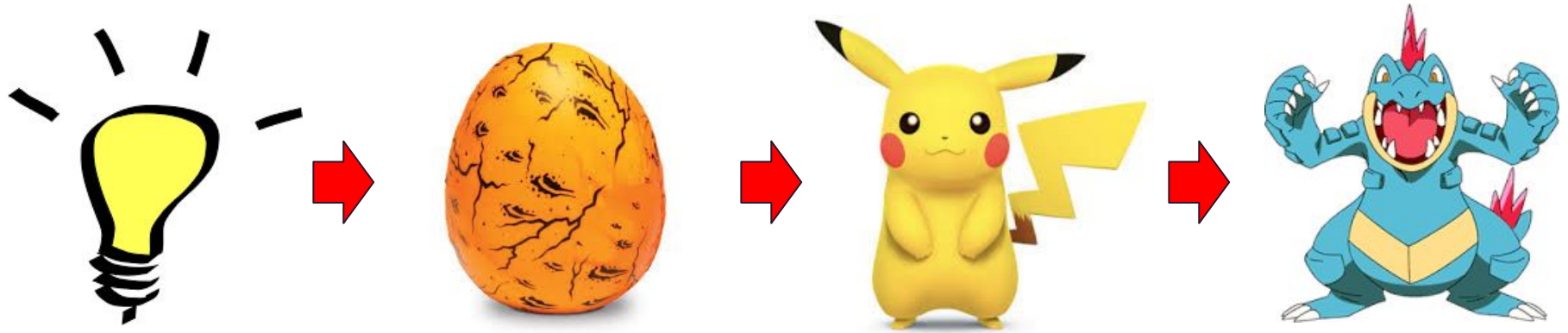
Como podemos componentizar



COM+



Sistemas Frankenstein



A problem has been detected and windows has been shut down to prevent damage to your computer.

The problem seems to be caused by the following file: kbdhid.sys

MANUALLY_INITIATED_CRASH

If this is the first time you've seen this stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any Windows updates you might need.

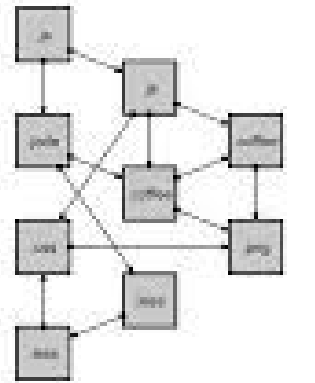
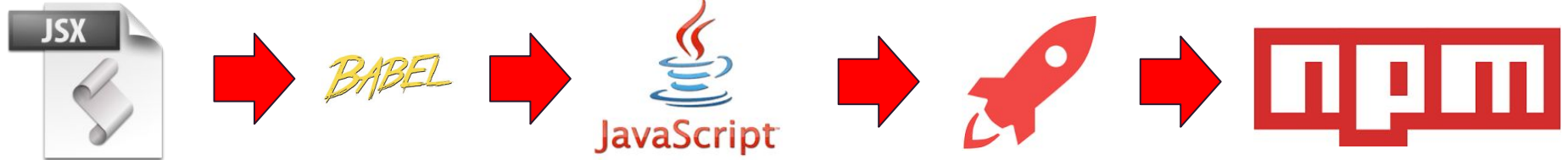
If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use safe mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

Technical Information:

*** STOP: 0x000000e2 (0x00000000, 0x00000000, 0x00000000, 0x00000000)

*** kbdhid.sys - Address 0x94efd1aa base at 0x94efb000 DateStamp 0x4a5bc705

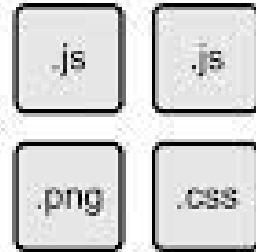
Web Components



modules
with dependencies



webpack
MODULE BUNDLER



static
assets

Dúvidas

Por que React?

- Apesar de não ser um Framework MVC ele é focado na composição de componentes.
- Ele encoraja fortemente a criação de componentes de tal forma que você os reutilize em seu projeto;
- Você consegue renderizar as Views HTML mais facilmente e reaproveitar a interface visual usando JSX;
- Altamente performático para atualização do DOM usando o Virtual DOM;

Vídeo: [React and the Virtual DOM](#)



Setup do Ambiente - React

1. Download **NodeJS**
 - a. <https://nodejs.org/en/>
2. Sinopia
 - a. Setup - **sudo npm install sinopia -g**
 - b. Start - **sinopia**
 - c. Registry - **npm set registry <http://localhost:4873/>**
 - d. New User - **npm adduser --registry http://localhost:4873/**
3. React e suas dependências
 - a. **sudo npm install create-react-app -g**

Primeiro React Project

```
mkdir curso-react  
cd curso-react  
create-react-app hello-world  
cd hello-world  
npm start
```

Navegue para <http://localhost:3000/>



Setup do Ambiente - MongoDB

UBUNTU

1. Import the public key **sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 0C49F3730359A14518585931BC711F9BA15703C6**
2. Create a list file for mongoDB **echo "deb [arch=amd64,arm64] <http://repo.mongodb.org/apt/ubuntu> xenial/mongodb-org/3.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.4.list**
3. Reload local packages **sudo apt-get update**
4. Install mongo **sudo apt-get install -y mongodb-org**
5. Starting mongo **sudo service mongod start**

Obs: on Ubuntu 16 you can find the mongo config => "cat /etc/mongod.conf" check the log and instances => "/var/lib/mongodb" and "/var/log/mongodb"

Testando MongoDB

```
renato@asus: sudo service mongod status
```

```
renato@asus: mongo
```

```
> use prodesp
```

```
> db.estados.insert({"nome": "SP"});
```

```
> db.estados.insert({"nome": "RJ"});
```

```
> db.cidades.insert({"uf": "SP", "nome": "Sao Paulo"});
```

```
> db.cidades.insert({"uf": "SP", "nome": "Santo Andre"});
```

```
> db.cidades.insert({"uf": "RJ", "nome": "Rio de Janeiro"});
```

```
> show collections
```

mongo → ativa o cliente para o mongod

use "db" → cria um banco caso ele não exista

show dbs → lista os bancos

show collections → lista as collections dentro de um banco

db.collection.insert({ }) → insere um registro

db.collection.find() → lista os registros

Dúvidas

Criando um Componente React Estados

```
mkdir prodesp-estados
```

```
cd prodesp-estados
```

```
npm init
```

```
npm install babel-core babel-loader babel-preset-es2015 babel-preset-react --save-dev
```

```
npm install babel-preset-react --save-dev
```

```
npm install react react-dom --save
```

```
npm install bootstrap --save-dev
```

```
npm install jquery --save-dev
```

```
sudo npm install webpack -g
```

```
mkdir dist
```

```
mkdir src
```

```
cd src
```

```
touch estado.jsx
```

```
touch index.jsx
```

```
touch ../webpack.config.js
```

Arquivo estado.jsx

```
import React, { PropTypes } from 'react';

const propTypes = {
  datasource: PropTypes.array
}

const defaultProps = {
  datasource: []
}

class EstadoItem extends React.Component {

  constructor(props) {
    super(props);
  }

  render(){
    return (<li className="list-group-item">{this.props.nome}</li>);
  };
}
```

```
class Estado extends React.Component {
  constructor(props) {
    super(props);
  }

  render(){
    const items = this.props.datasource.map( (item, i) =>
      <EstadoItem key={i} nome={item.nome} />
    );

    return (
      <ul className="list-group">
        {items}
      </ul>
    );
  };
}
```

```
Estado.propTypes = propTypes;
Estado.defaultProps = defaultProps;
```

```
export default Estado;
```

Arquivo index.jsx

```
import React from 'react';  
import {render} from 'react-dom';  
import Estado from './estado.jsx';
```

// agora criamos uma classe chamada App que adicionará o componente Alert na página

```
class App extends React.Component {  
  render() {  
    let items = [{sigla:"SP", nome:"Sao Paulo"}, {sigla:"RJ", nome:"Rio de Janeiro"}];  
    return (  
      <div>  
        <Estado datasource={items} />  
      </div>  
    );  
  }  
}  
  
render(<App />, document.getElementById('app'));
```

Arquivo index.html

```
<!DOCTYPE html>
<html lang="en">
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Demonstração</title>
  <link href="node_modules/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<div id="app" />
<!-- bootstrap -->
<script type="text/javascript" src="node_modules/jquery/dist/jquery.min.js"></script>
<script type="text/javascript" src="node_modules/bootstrap/dist/js/bootstrap.min.js"></script>
<script type="text/javascript" src="dist/index.js"></script>
</body>
</html>
```

Arquivo package.json

```
{
  "name": "prodesp-estados",
  "version": "1.0.1",
  "description": "",
  "main": "src/estado.jsx",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "babel-core": "^6.22.1",
    "babel-loader": "^6.2.10",
    "babel-preset-es2015": "^6.22.0",
    "babel-preset-react": "^6.22.0",
    "bootstrap": "^3.3.7",
    "jquery": "^3.1.1"
  },
  "dependencies": {
    "react": "^15.4.2",
    "react-dom": "^15.4.2"
  }
}
```


Arquivo webpack.config.js

```
var path = require('path');

module.exports = {
  devtool: 'source-map',
  entry: {
    'index': ['./src/index.jsx'],
    'estado': ['./src/estado.jsx']
  },
  output: {
    path: path.resolve(__dirname, 'dist'),
    filename: "[name].js"
  },
  module: {
    loaders: [{
      test: /\.jsx?$/,
      loader: 'babel',
      exclude: /node_modules/,
      query: {
        presets: ['es2015', 'react']
      }
    }]
  }
};
```

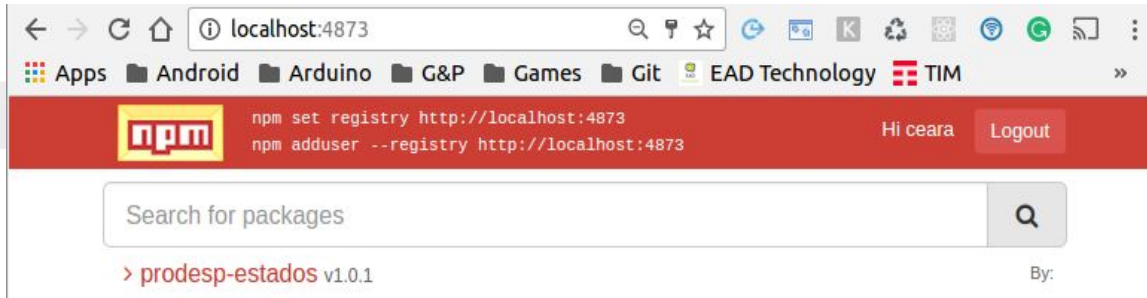
Compilando

```
cd prodesp-estados
```

```
webpack
```

Publicando

```
npm publish
```



Removendo

```
npm unpublish prodesp-estados --force
```

Dúvidas

Usando o Componente Estados

```
cd curso-react  
mkdir app  
npm init  
npm install webpack --save-dev  
npm install babel-core babel-loader babel-preset-es2015 babel-preset-react --save-dev  
npm install babel-preset-react --save-dev  
npm install react react-dom --save  
npm install bootstrap --save-dev  
npm install jquery --save-dev  
npm install prodesp-estados --save  
mkdir bin  
mkdir src  
touch webpack.config.js  
touch index.html  
cd src  
touch index.jsx
```

Arquivo index.jsx

```
import React from 'react';
import {render} from 'react-dom';
import Estado from 'prodesp-estados';

class App extends React.Component {
  render () {
    let items = [{sigla:"SP", nome:"Sao Paulo"}, {sigla:"RJ", nome:"Rio de Janeiro"}];
    return (
      <div>
        <Estado datasource={items} />
      </div>
    );
  }
}

render(<App />, document.getElementById('app'));
```

Arquivo index.html

```
<!DOCTYPE html>
<html lang="en">
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Demonstração</title>
  <link href="node_modules/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<div id="app" />
<!-- bootstrap -->
<script type="text/javascript" src="node_modules/jquery/dist/jquery.min.js"></script>
<script type="text/javascript" src="node_modules/bootstrap/dist/js/bootstrap.min.js"></script>
<script type="text/javascript" src="bin/app.bundle.js"></script>
</body>
</html>
```

Arquivo webpack.config.js

```
var path = require('path');

module.exports = {
  devtool: 'source-map',
  entry: './src/index.jsx',
  output: {
    path: path.resolve(__dirname, 'bin'),
    filename: "app.bundle.js"
  },
  module : {
    loaders: [{
      test: /\.jsx?$/,
      loader: 'babel-loader',
      query: {
        presets:['es2015', 'react']
      }
    }]
  }
};
```


Dúvidas

Trabalhando com Eventos



Add Eventos em EstadoItem

```
class EstadoItem extends React.Component {  
  
  constructor(props) {  
    super(props);  
    this.onItemClick = this.onItemClick.bind(this);  
  }  
  
  onItemClick(){  
    let json = {sigla: this.props.sigla, nome: this.props.nome};  
    this.props.onSelected(json);  
  }  
  
  render(){  
    return (<li className="list-group-item" onClick={this.onItemClick}><a href="#">{this.props.nome}</a></li>);  
  };  
}
```

Add Eventos em Estado

```
const propTypes = {
  datasource: PropTypes.array,
  onSelected: PropTypes.func
}

class Estado extends React.Component {

  constructor(props) {
    super(props);
  }

  render(){
    const items = this.props.datasource.map( (item, i) =>
      <EstadoItem key={i} nome={item.nome} sigla={item.sigla} onSelected={this.props.onSelected} />
    );

    return (
      <ul className="list-group">
        {items}
      </ul>
    );
  }
}
```

Testando em Index.jsx

```
class App extends React.Component {  
  
  constructor(props) {  
    super(props);  
    this.onSelected = this.onSelected.bind(this);  
  }  
  
  onSelected(data){  
    console.log(data);  
  }  
  
  render () {  
    let items = [{sigla:"SP", nome:"Sao Paulo"}, {sigla:"RJ", nome:"Rio de Janeiro"}];  
    return (  
      <div>  
        <Estado datasource={items} onSelected={this.onSelected} />  
      </div>  
    );  
  }  
}
```

Dúvidas

Componente Cidades

- Faça uma cópia do diretório prodesp-estados
- Nomeie a pasta como prodesp-cidades
- Altere os arquivos `src\estado.jsx` para `src\cidade.jsx`
- Altere o arquivo `package.json` o campo `nome` para `prodesp-cidades`
- Remova o conteúdo da pasta **DIST** pois a mesma será atualizada com os novos arquivos

Classe Cidade

```
class Cidade extends React.Component {
  constructor(props) {
    super(props);
  }
  render(){
    const items = this.props.datasource.map( (item, i) =>
      <CidadeItem key={i} nome={item.nome} area={item.area} populacao={item.populacao} capital={item.capital} />
    );
    return (
      <table className="table table-condensed table-striped">
        <caption>{this.props.title}</caption>
        <thead>
          <tr>
            <th>Nome</th><th>População</th><th>Area</th><th>Capital</th>
          </tr>
        </thead>
        <tbody>{items}</tbody>
      </table>
    );
  };
};

Cidade.propTypes = propTypes;
Cidade.defaultProps = defaultProps;

export default Cidade;
```

Arquivo Index.jsx

```
import React from 'react';
import {render} from 'react-dom';
import Cidade from './cidade.jsx';

class App extends React.Component {
  constructor(props) {
    super(props);
  }
  render () {
    let items =
      [
        {nome:"Sao Paulo", area: 450234, populacao: 10000000, capital: true},
        {nome:"Santo Andre", area: 350234, populacao: 5000000, capital: false},
        {nome:"Sao Bernardo", area: 234321, populacao: 7000000, capital: false}
      ];
    return (
      <div className="bs-example" data-example-id="simple-table">
        <Cidade datasource={items} title="Relacao de Cidades do Estado de SP" />
      </div>
    );
  }
}

render(<App />, document.getElementById('app'));
```

Publicando o componente Cidades



Publicando o componente Cidades

- não esqueça de empacotar seu controle usando **webpack**
- publique o componente usando **npm publish**
- abra o projeto app e instale o novo componente usando npm
npm install prodesp-cidades --save
- vamos editar o projeto app

Index.jsx

```
import React from 'react';
import {render} from 'react-dom';
import Estado from 'prodesp-estados';
import Cidade from 'prodesp-cidades';

constructor(props){
  super(props);
  this.state = {
    estados: [{sigla:"SP", nome:"Sao Paulo"}, {sigla:"RJ", nome:"Rio de Janeiro"}],
    cidades: [],
    title: "Relação de Cidades"
  }
  this.onSelected = this.onSelected.bind(this);
}

render () {
  return (
    <div className="row">
      <div className="col-lg-6">
        <Estado datasource={this.state.estados} onSelected={this.onSelected} />
      </div>
      <div className="col-lg-6">
        <Cidade datasource={this.state.cidades} title={this.state.title} />
      </div>
    </div>
  );
}
```

Atualizando as Cidades usando onSelected

```
onSelected(data){
  console.log(data);

  var cidades = [];
  if (data.sigla === "SP"){
    cidades.push({nome:"Sao Paulo", area: 450234, populacao: 10000000, capital: true});
    cidades.push({nome:"Santo Andre", area: 350234, populacao: 5000000, capital: false});
    cidades.push({nome:"Sao Bernardo", area: 234321, populacao: 7000000, capital: false});
  } else {
    cidades.push({nome:"Angra dos Reis", area: 450234, populacao: 10000000, capital: false});
    cidades.push({nome:"Barra Mansa", area: 350234, populacao: 5000000, capital: false});
    cidades.push({nome:"Rio de Janeiro", area: 234321, populacao: 7000000, capital: true});
    cidades.push({nome:"Macaé", area: 234321, populacao: 7000000, capital: false});
    cidades.push({nome:"Marica", area: 234321, populacao: 7000000, capital: false});
  }

  this.setState({
    title: "Relação de Cidades do Estado " + data.nome,
    cidades: cidades
  });
}
```

Resultado Final

← → ↺ 🏠 ⓘ localhost:63342/app/index.html?_ijt=tm6o3ot7 🔍 ☆ 🔄 📄 K 🗑️ 🧩 📶 🟢 📶 ⋮			
📱 Apps 📁 Android 📁 Arduino 📁 G&P 📁 Games 📁 Git 📱 EAD Technology 📱 TIM 📁 Imagens »			
Sao Paulo			
Rio de Janeiro			
Relação de Cidades do Estado Rio de Janeiro			
Nome	População	Area	Capital
Angra dos Reis	450234	10000000	<input type="checkbox"/>
Barra Mansa	350234	5000000	<input type="checkbox"/>
Rio de Janeiro	234321	7000000	<input checked="" type="checkbox"/>
Macaé	234321	7000000	<input type="checkbox"/>
Marica	234321	7000000	<input type="checkbox"/>

Dúvidas

Minificando a aplicação



Webpack plugins

```
var webpack = require('webpack');  
var PROD = process.env.PROD_ENV || "0";  
module.exports = {  
  ....  
  plugins: (PROD === "1") ? [  
    new webpack.optimize.UglifyJsPlugin({  
      compress: { warnings: false }  
    })  
  ] : []  
}
```

Webpack plugins

Na bash ou sheel você digita:

```
export PROD_ENV=1  
echo $PROD_ENV  
webpack
```

Links úteis

Curso online sobre React e muito mais

<https://egghead.io/courses>

Repositório contendo os fontes do projeto

<https://github.com/renatomatos79/react-components>

E-mail para contato

renato.matos79@gmail.com

Skype

renato.matos.79

Fim