Properties validation is a useful way to force the correct usage of the components. This will help during development to avoid future bugs and problems, once the app becomes larger. It also makes the code more readable, since we can see how each component should be used.

Validating Props

In this example, we are creating **App** component with all the **props** that we need. **App.propTypes** is used for props validation. If some of the props aren't using the correct type that we assigned, we will get a console warning. After we specify validation patterns, we will set **App.defaultProps**.

App.jsx

import PropTypes from 'prop-types';

import React from 'react';

import ReactDOM from 'react-dom';

class App extends React.Component {

render() {

return (

<div>

<h1> Hello, {this.props.name} </h1>

<h3>Array: {this.props.propArray}</h3>

<h3>Bool: {this.props.propBool ? "True..." : "False..."}</h3>

<h3>Func: {this.props.propFunc(3)}</h3>

<h3>Number: {this.props.propNumber}</h3>

<h3>String: {this.props.propString}</h3>

</div>

);

}

}

App.propTypes = {

name: PropTypes.string,

propArray: PropTypes.array.isRequired,

propBool: PropTypes.bool.isRequired,

propFunc: PropTypes.func,

propNumber: PropTypes.number,

propString: PropTypes.string,

};

App.defaultProps = {

name: 'Tutorialspoint.com',

propArray: [1, 2, 3, 4, 5],

propBool: true,

propFunc: function(e) {

return e

},

propNumber: 1,

propString: "String value..."

}

export default App;

main.js

import React from 'react';

import PropTypes from 'prop-types';

import ReactDOM from 'react-dom';

import App from './App.jsx';

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

webpack.config.js

var config = {

entry: './main.js',

output: {

path:'/',

filename: 'index.js',

},

devServer: {

inline: true,

port: 8080

},

externals: {

'react': 'React'

},

module: {

loaders: [

{

test: /\.jsx?$/,

exclude: /node\_modules/,

loader: 'babel-loader',

query: {

presets: ['es2015', 'react']

}

}

]

}

}

module.exports = config;

Since all **props** are valid, we will get the following result.

As can be noticed, we have use **isRequired** when validating **propArray** and **propBool**. This will give us an error, if one of those two don't exist. If we delete **propArray:** **[1,2,3,4,5]** from the **App.defaultProps** object, the console will log a warning.

React Props Validation Error

If we set the value of **propArray: 1**, React will warn us that the propType validation has failed, since we need an array and we got a number.

React Props Validation Error 2