

A Developer Diary

{about:"code learn and share"}

[Home](#)[Data Science](#)[Java](#)[JavaScript](#)[jBPM](#)[Tools](#)[Tips](#)[About](#)

April 5, 2016 By [Abhisek Jana](#) — [Leave a Comment \(Edit\)](#)

How to create reusable charts with React and D3 Part1



In this series we will continue working on our D3 components that we have created in our past lesson and make them fully reusable, extensible and maintainable. This **How to create reusable charts with React and D3 Part1** would cover the basics and then we will dive into more complex scenarios.

Prerequisites:

1. You would need basic understanding of React and D3 to follow along with this course. I already have posted many tutorials on creating charts using D3. You can find them here (<https://www.adeveloperdiary.com/category/d3-js/>)
2. We will start from where we had left in our last tutorial (**How to How to integrate React and D3**). In case you have not read it, please go through it and find many useful details.
<https://www.adeveloperdiary.com/react-js/how-to-integrate-react-and-d3-the-right-way/>

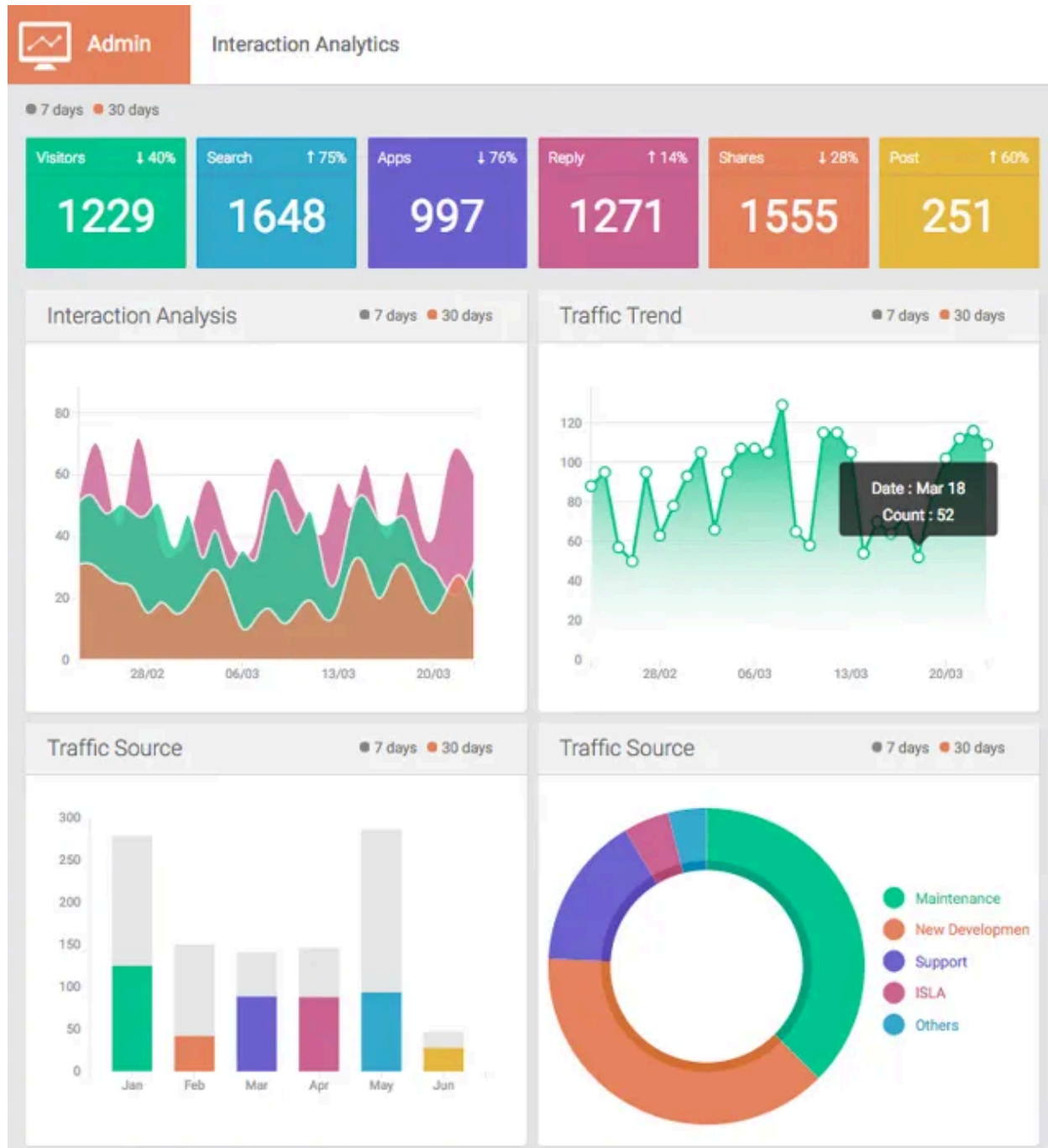
Objectives:

- We will extend out work and make the React components fully reusable, configurable , extensible and maintainable.

- We will create a Analytics Dashboard in order to implement our concept.

Analytics Dashboard

Here is the screen print of the dashboard we will be creating. The 7 Days & 30 Days filter is active, you can click on them to reload the data (We are loading random dataset for this demo)



Click on the button below to see the demo:

[Demo](#)

HTML Scaffolding:

We will start with the HTML first before we convert everything to React components. You can find the HTML layout [here](#).

[Code](#)

We will be using bootstrap for the layout of the page. We have full sized header and a container. Here is the structure of the HTML, just to get enough idea about it. I have intentionally left many styles in the HTML itself so that you don't have to refer the `style.css` all the time.

React Components:

Our next task would be to convert the HTML into React Components. Add the `app.jsx` and here is the structure of the main Page component. The `MainRangeSelection` is the top filter then we have the `Cards` followed by `MainContainer` & `SubContainer`.

[illegible]

```

    }
  });


```

We have the date range filter in 5 different places, so it's a good to create a React Component for that. Let's call it Range. We will use svg element to create the small circles. Later we will enhance this component to use events.


```

var Range=React.createClass({
  render:function(){
    return(

```



7 days



30 days

```

    );
  }
});

```

Place the Range inside the `MainRangeSelection` component.

```

var MainRangeSelection=React.createClass({
  render:function(){
    return(

```

```
    );  
  }  
});
```

Next, we will add the Cards component. We will loop through the color array to generate the cards. We will customize it more later in this series.

```
var Cards=React.createClass({  
  render:function(){  
  
    var color=  
    ['#53c79f','#64b0cc','#7a6fca','#ca6f96','#e58c72','#e5c072'];  
  
    var cards=color.map(function(d,i){  
      var style={  
        'backgroundColor':d  
      };  
      return (  

```

70%

3,502

```
    );  
  });  
  
  return(  
  
    {cards}  
  
  );  
}  
});
```


We have 4 panels, so again we will create a **Panel** & **PanelHeader** component. The Panel component is very straightforward. We will wrap everything in a div and use **this.props.children** to place all the child elements inside. **this.props.children** is a very powerful component, we will see more use later.

The **PanelHeader** displays the title and Range component.

```
var Panel=React.createClass({
  render:function() {
    return (

      {this.props.children}

    );
  }
});
```

```
var PanelHeader=React.createClass({
  propTypes: {
    title:React.PropTypes.string
  },
  render: function () {
    return (

      {this.props.title}
```

```
{this.props.children}
```

```
    );  
  }  
});
```

The `MainContainer` is below. The `SubContainer` would be exactly same at this point of time. Notice how I have defined `Panel`, `PanelHeader` & `Range` component.

```
var MainContainer=React.createClass({  
  render:function(){  
    return(  

```

```
    );  
  }  
});
```

Code

Please find the code we have changed so far.

Code

Demo

Here is the demo, there is not much of difference from our earlier HTML demo.

Demo

Conclusion

We just converted the HTML to React Components in this part. In part2 we will start adding our charts using D3.

Related



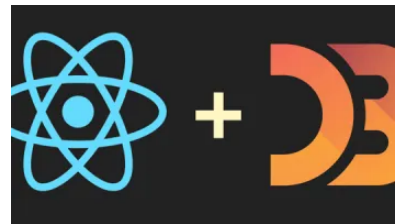
How to create reusable charts with React and D3 Part2

In "D3.js"



How to create reusable charts with React and D3 Part3

In "D3.js"



How to integrate React and D3 – The right way

In "D3.js"

Filed Under: [D3.js](#), [React JS](#) | Tagged With: [Charts](#), [d3](#), [d3.js](#), [Dashboard](#), [example](#), [integrate react and D3](#), [React](#), [React+d3](#), [step by step](#), [tutorial](#), [Visualization](#)

Subscribe to stay in loop

* indicates required

Email Address *

Subscribe

Leave a Reply

Logged in as Abhisek Jana. [Edit your profile](#). [Log out?](#) Required fields are marked *

Comment *

Post Comment

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

Copyright © 2024 A Developer Diary