React Application is a set of Components

1.We started with the App Component , in which we had the Friends Component <Apps>🡪 <Friends>

HTML

Communication between Parent to Child ie. App to Friends and Viceversa

Communication =- Parsing data from Parent to Child

<App> 🡪 Friends Data Transfer

WE will use data set state ={ cmpheading : “Values”}

For collections

friendsdata1: [ "John", "Sumitha", "Kumaran" ]

state= {Compheading : "Values",

friendsdata1: [ "John", "Sumitha", "Kumaran" ]

}

render(){

return <div> <h1> <Friends name={this.state.friendsdata1}/> Welcome{this.state.Compheading}</h1></div>

}

}

In The Child class ie, Friends like .this.props.propertyname

console.log(this.props.name)

This .props.<<array>>.map 🡪 Function to be used as a unordered list.

Example.js

//Component Mmber is a named Export

import React, { Component } from "react";

class Friends extends Component {

//Shd override the Render Method

//Definingthe state component

state = { Compheading: "Values" }

render() {

// console.log(this.props.name)

return <div>

<h1> Welcome to React friend Component {this.state.Compheading}</h1>

<ul>{

this.props.name.map((e,i) => {

return <li key={i}>{e}</li>

}

)

}

</ul>

</div>

}

}

// We need to Use it in index.js --> export defalut

export default Friends;

APP.js [Parent – sharing values to chile.]

//Component Mmber is a named Export

import React,{Component} from "react";

import Friends from "./Example";

class App extends Component{

//Shd override the Render Method

//Definingthe state component

state= {Compheading : "Values",

friendsdata1: [ "John", "Sumitha", "Kumaran" ]

}

render(){

return <div> <h1> <Friends name={this.state.friendsdata1}/> Welcome{this.state.Compheading}</h1></div>

}

}

// We need to Use it in index.js --> export defalut

export default App;

Assign the Condition to an output variable.

//Component Mmber is a named Export

import React, { Component } from "react";

class Friends extends Component {

//Shd override the Render Method

//Definingthe state component

state = { Compheading: "Values" }

render() {

// console.log(this.props.name)

let output = null

if (this.props.name !==undefined) {

output=this.props.name.map( (e,i) => {

return <li key={i}>{e}</li>

} )

}

else{

output= []

}

return <div>

<h1> Welcome to React friend Component {this.state.Compheading}</h1>

<ul>{

output

}

</ul>

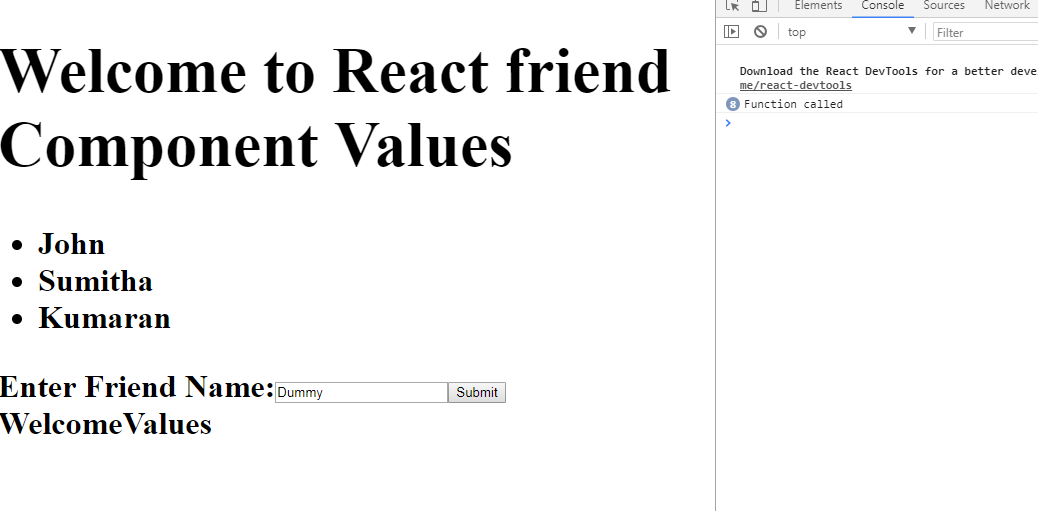
</div>

}

}

// We need to Use it in index.js --> export defalut

export default Friends;



//Component Mmber is a named Export

import React, { Component } from "react";

class Friends extends Component {

//Shd override the Render Method

//Definingthe state component

state = { Compheading: "Values" }

render() {

// console.log(this.props.name)

let output = null

if (this.props.name !==undefined) {

output=this.props.name.map( (e,i) => {

return <li key={i}>{e}</li>

} )

}

else{

output= []

}

return <div>

<h1> Welcome to React friend Component {this.state.Compheading}</h1>

<ul>{

output

}

</ul>

</div>

}

}

// We need to Use it in index.js --> export defalut

export default Friends;

export class AddFriend extends Component {

state ={ newFriend : "Dummy"}

handleOnChange = () =>{

console.log("Function called")

}

render(){

return(

<div> Enter Friend Name:

<input type = "text"

value={this.state.newFriend}

onChange={this.handleOnChange}

>

</input>

<button type = "submit"> Submit</button>

</div>

)

}

}

App.js

//Component Mmber is a named Export

import React,{Component} from "react";

import Friends,{AddFriend} from "./Example";

class App extends Component{

//Shd override the Render Method

//Definingthe state component

state= {Compheading : "Values",

friendsdata1: [ "John", "Sumitha", "Kumaran" ]

}

render(){

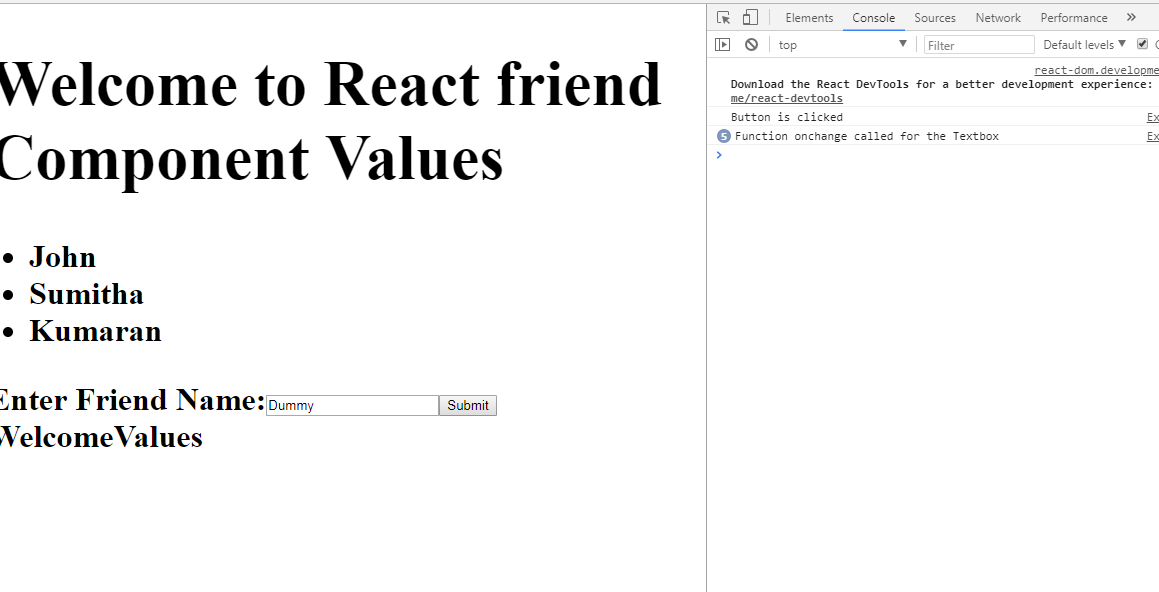
return <div> <h1> <Friends name={this.state.friendsdata1}/> <AddFriend/> Welcome{this.state.Compheading}</h1></div>

}

}

// We need to Use it in index.js --> export defalut

export default App;



Eample.js

//Component Mmber is a named Export

import React, { Component } from "react";

class Friends extends Component {

//Shd override the Render Method

//Definingthe state component

state = { Compheading: "Values" }

render() {

// console.log(this.props.name)

let output = null

if (this.props.name !==undefined) {

output=this.props.name.map( (e,i) => {

return <li key={i}>{e}</li>

} )

}

else{

output= []

}

return <div>

<h1> Welcome to React friend Component {this.state.Compheading}</h1>

<ul>{

output

}

</ul>

</div>

}

}

// We need to Use it in index.js --> export defalut

export default Friends;

export class AddFriend extends Component {

state ={ newFriend : "Dummy"}

handleOnChange = () =>{

console.log("Function onchange called for the Textbox")

}

buttonClick = () => {

console.log("Button is clicked")

}

render(){

return(

<div> Enter Friend Name:

<input type = "text"

value={this.state.newFriend}

onChange={this.handleOnChange}

>

</input>

<button type = "submit" onClick={this.buttonClick}> Submit</button>

</div>

)

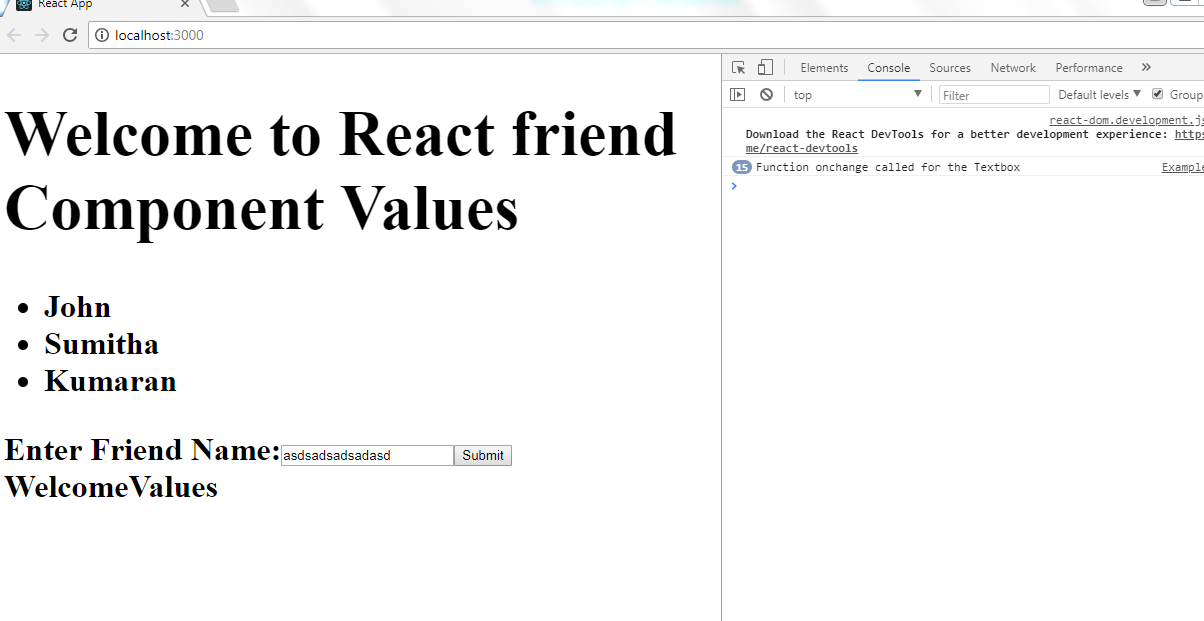
}

}

Event – Default Object In Javascript

Change the handleOnChange to Accept an Event as it store all the events in Javascript.

COC



**CODE SNIPPE**

handleOnChange = (event) =>{

console.log("Function onchange called for the Textbox")

this.setState ({newFriend:event.target.value})

//this.state.newFriend=event.target.value;

}

Full Code – Example.js

//Component Mmber is a named Export

import React, { Component } from "react";

class Friends extends Component {

//Shd override the Render Method

//Definingthe state component

state = { Compheading: "Values" }

render() {

// console.log(this.props.name)

let output = null

if (this.props.name !==undefined) {

output=this.props.name.map( (e,i) => {

return <li key={i}>{e}</li>

} )

}

else{

output= []

}

return <div>

<h1> Welcome to React friend Component {this.state.Compheading}</h1>

<ul>{

output

}

</ul>

</div>

}

}

// We need to Use it in index.js --> export defalut

export default Friends;

export class AddFriend extends Component {

state ={ newFriend : "Dummy"}

handleOnChange = (event) =>{

console.log("Function onchange called for the Textbox")

this.setState ({newFriend:event.target.value})

//this.state.newFriend=event.target.value;

}

buttonClick = () => {

console.log("Button is clicked")

}

render(){

return(

<div> Enter Friend Name:

<input type = "text"

value={this.state.newFriend}

onChange={this.handleOnChange}

>

</input>

<button type = "submit" onClick={this.buttonClick}> Submit</button>

</div>

)

}

}

# PASSING DATA FOM CHILD TO PARENT

1. Child to Parent comms is a combination is thru State and Props
2. Use Spread Operator
3. this.setState({friendsdata1: [...this.state.friendsdata1,friendname]})

Example .js

//Component Mmber is a named Export

import React, { Component } from "react";

class Friends extends Component {

//Shd override the Render Method

//Definingthe state component

state = { Compheading: "Values" }

render() {

// console.log(this.props.name)

let output = null

if (this.props.name !==undefined) {

output=this.props.name.map( (e,i) => {

return <li key={i}>{e}</li>

} )

}

else{

output= []

}

return <div>

<h1> Welcome to React friend Component {this.state.Compheading}</h1>

<ul>{

output

}

</ul>

</div>

}

}

// We need to Use it in index.js --> export defalut

export default Friends;

export class AddFriend extends Component {

state ={ newFriend : "Dummy"}

handleOnChange = (event) =>{

console.log("Function onchange called for the Textbox")

this.setState ({newFriend:event.target.value})

//this.state.newFriend=event.target.value;

}

buttonClick = () => {

// console.log("Button is clicked")

//Assign the Property to State

this.props.addNew(this.state.newFriend)

}

render(){

return(

<div> Enter Friend Name:

<input type = "text"

value={this.state.newFriend}

onChange={this.handleOnChange}

>

</input>

<button type = "submit" onClick={this.buttonClick}> Submit</button>

</div>

)

}

}

App.js

//Component Mmber is a named Export

import React,{Component} from "react";

import Friends,{AddFriend} from "./Example";

class App extends Component{

//Shd override the Render Method

//Definingthe state component

state= {Compheading : "Values",

friendsdata1: [ "John", "Sumitha", "Kumaran" ]

}

addNewFriend = (friendname) =>{

console.log("New Function addNewFriend" +friendname);

this.setState({friendsdata1: [...this.state.friendsdata1,friendname]})

}

render(){

return <div> <h1> <Friends name={this.state.friendsdata1}/>

<AddFriend addNew ={this.addNewFriend }/> Welcome{this.state.Compheading}</h1></div>

}

}

// We need to Use it in index.js --> export defalut

export default App;

PROP TYPES; DEFAULT Props

Npm install prop-types –S

Friends.defaultProps = {

name:["F1","F2"]

}

//Component Mmber is a named Export

import React, { Component } from "react";

import PropTypes from "prop-types"

class Friends extends Component {

//Shd override the Render Method

//Definingthe state component

state = { Compheading: "Values" }

render() {

// console.log(this.props.name)

let output = null

if (this.props.name !==undefined) {

output=this.props.name.map( (e,i) => {

return <li key={i}>{e}</li>

} )

}

else{

output= []

}

return <div>

<h1> Welcome to React friend Component {this.state.Compheading}</h1>

<ul>{

output

}

</ul>

</div>

}

}

// We need to Use it in index.js --> export defalut

export default Friends;

export class AddFriend extends Component {

state ={ newFriend : "Dummy"}

handleOnChange = (event) =>{

console.log("Function onchange called for the Textbox")

this.setState ({newFriend:event.target.value})

//this.state.newFriend=event.target.value;

}

buttonClick = () => {

// console.log("Button is clicked")

//Assign the Property to State

this.props.addNew(this.state.newFriend)

}

render(){

return(

<div> Enter Friend Name:

<input type = "text"

value={this.state.newFriend}

onChange={this.handleOnChange}

>

</input>

<button type = "submit" onClick={this.buttonClick}> Submit</button>

</div>

)

}

}

//Add peroperty types to Addfriend -

//The peroperty is addNew - add New is Function type & is marked as required

AddFriend.propTypes = {

addNew: PropTypes.func.isRequired

}

Friends.defaultProps = {

name:["F1","F2"]

}

//Component Mmber is a named Export

import React,{Component} from "react";

import Friends,{AddFriend} from "./Example";

class App extends Component{

//Shd override the Render Method

//Definingthe state component

state= {Compheading : "Values",

friendsdata1: [ "John", "Sumitha", "Kumaran" ]

}

addNewFriend = (friendname) =>{

console.log("New Function addNewFriend" +friendname);

this.setState({friendsdata1: [...this.state.friendsdata1,friendname]})

}

render(){

return <div>

<Friends/>

<Friends name={this.state.friendsdata1}/>

<AddFriend addNew ={this.addNewFriend }/>

{<h1> Welcome{this.state.Compheading}</h1>}

{/\*<h1> Welcome{this.state.Compheading}</h1>\*/}

</div>

}

}

// We need to Use it in index.js --> export defalut

export default App;