31-short-lnk-app [video](#)

Creating Modals with React-Modal

- we can find more info about modals at https://github.com/reactjs/react-modal
- lets install it in terminal

```
joses-MacBook-Pro:short-lnk mendoza$ meteor npm install react-modal@1.6.5 --save
```

- **now** go over the AddLink.js
- lets first re arrange our components Link.js
- this is what the Link.js should look like

```jsx
import React from 'react';


import  LinksList  from './LinksList';

import PrivateHeader from './PrivateHeader';

import AddLink from './AddLink';

import LinksListFilters from './LinksListFilters'


export default () => {

    return (

        <div>

            <PrivateHeader title="Your Links"/>

            <LinksListFilters/>

            <AddLink/>

            <LinksList/>

        </div>

    );

}
```

- now lets switch back to AddLink.js
- import React modal

```jsx
import React from 'react';

import Modal from 'react-/modal';

import { Meteor } from 'meteor/meteor';
```

```
export default class AddLink extends React.Component{

    constructor(props){

        super(props);

        this.state = {

            url: ''

        }

    }
```

- now we can use it inside of our render() method
- we are going to render the form inside of modal

```
    render(){

        return(

            <div>

                <Modal>

                    <p>Add Link</p>

                    <form onSubmit={this.onSubmit.bind(this)}>

                        <input type="text" placeholder="URL" value={this.state.url} onChange={th
is.onChange.bind(this)}/>

                        <button>Add Link</button>

                    </form>

                </Modal>

            </div>

        );

    }

}
```

- Modal does require a few properties: isOpen - a boolean value, and contactLabel - a string, description of the modal

```
    render(){

        return(

            <div>

                <Modal isOpen={true}  contentLabel="Add link">

                    <p>Add Link</p>

                    <form onSubmit={this.onSubmit.bind(this)}>

                        <input type="text" placeholder="URL" value={this.state.url} onChange={th
is.onChange.bind(this)}/>
```

```
                    <button>Add Link</button>

                </form>

            </Modal>

        </div>

    );

  }

}
```

- lets create an isOpen state up above

```
import React from 'react';

import Modal from 'react-modal';

import { Meteor } from 'meteor/meteor';


export default class AddLink extends React.Component{

    constructor(props){

        super(props);

        this.state = {

            url: '',

            isOpen: false

        }

    }
```

- and now we can pass that property down below

```
    render(){

        return(

            <div>

                <Modal isOpen={this.state.isOpen}  contentLabel="Add link">

                    <p>Add Link</p>

                    <form onSubmit={this.onSubmit.bind(this)}>

                        <input type="text" placeholder="URL" value={this.state.url} onChange={th
is.onChange.bind(this)}/>

                        <button>Add Link</button>

                    </form>

                </Modal>

            </div>

        );
```

```
        }

    }
```

- now lets create a button so we can open this modal
- inside the button we will give it a onClick event and an arrow function that will turn isOpen property to true

```
render(){

    return(

        <div>

            <button onClick={()=> this.setState({isOpen: true})}>+ Add Link</button>

            <Modal isOpen={this.state.isOpen}  contentLabel="Add link">

                <p>Add Link</p>

                <form onSubmit={this.onSubmit.bind(this)}>

                    <input type="text" placeholder="URL" value={this.state.url} onChange={th
is.onChange.bind(this)}/>

                        <button>Add Link</button>

                </form>

            </Modal>

        </div>

    );

    }

}
```

- we later are going to add a button that closes the button
- and we are also going to clear the text field once you hit cancel

```
    render(){

        return(

            <div>

                <button onClick={()=> this.setState({isOpen: true})}>+ Add Link</button>

                <Modal isOpen={this.state.isOpen}  contentLabel="Add link">

                    <p>Add Link</p>

                    <form onSubmit={this.onSubmit.bind(this)}>

                        <input type="text" placeholder="URL" value={this.state.url} onChange={th
is.onChange.bind(this)}/>

                            <button>Add Link</button>

                    </form>
```

```
                <button onClick={()=>{
                    this.setState({isOpen:false, url: ''});
                }}>Cancel</button>
            </Modal>
        </div>
    );
    }
}
```

- now we want to close the modal once the users enters a valid url, and you do that at the onSubmit() method

```
import React from 'react';
import Modal from 'react-modal';
import { Meteor } from 'meteor/meteor';


export default class AddLink extends React.Component{
    constructor(props){
        super(props);
        this.state = {
            url: '',
            isOpen: false
        }
    }
    onSubmit(e){
        const { url } = this.state
        e.preventDefault();


        if(url){
            Meteor.call('links.insert', url, (err, res)=>{
                if(!err){
                    this.setState({url:'', isOpen: false});
                }
            });
        }
    }
```

-