

24-short-link-app video

Copying Link URL to clipboard

- `document.execCommand` will give you the copy to clipboard feature and all browsers have it
- to test if you have run following command on the console

```
document.execCommand('copy')  
  
true
```

- if you get a true that means you have this feature
- now lets copy what we type in an input
- to do that lets first use `querySelector()`
- which will return the `<input>`

```
document.querySelector('input')  
  
<input type="text" placeholder="URL">
```

- we then run the following command to select () the value of the text field
- which gives you undefined

```
document.querySelector('input').select()  
  
undefined
```

- we then run `execCommand('copy')` to copy to the clip board

```
document.execCommand('copy')  
  
true
```

- and now if you paste you should get the value back
- but we are going to be using a 3rd party library that has all of these features
- you can find this feature at <https://www.npmjs.com/package/clipboard>
- so lets install it in the terminal

```
meteor npm install --save clipboard@1.5.16
```

- now lets go to `LinksListItem` and import it

```
import React from 'react';  
import Clipboard from 'clipboard';
```

- the clipboard is not specific to react, it can work with others like angular
- we are going to be using the life cycle function componentDidMount
 - we then are going to create a button that hold the value Copy with an attribute of ref and then another attribute that it is specific to the clipboard which is data-clipboard-text, and for now the value will be static
 -

```
import React from 'react';
import Clipboard from 'clipboard';

export default class LinkListItem extends React.Component{
  componentDidMount(){

  }
  render(){
    return(
      <div>
        <p>{this.props.url}</p>
        <p>{this.props.shortUrl}</p>
        <button ref="copy" data-clipboard-text="from my react app">Copy</button>
      </div>
    )
  }
}
```

- and now are going to initialize the library up above.
- first we are going to create a new instance of clipboard
- and we are going to pass the ref value to it

```
import React from 'react';
import Clipboard from 'clipboard';

export default class LinkListItem extends React.Component{
  componentDidMount(){
    new Clipboard(this.refs.copy);
  }
}
```

- and now we are set up and ready to check this in our browser
- if you click the button copy and pasting is in the console in chrome tools, you should be able to see

from my react app

- we are going to be exploring more with the clipboard library- we are going to be using the success event and error event
- go back to you LinksListItem.js and we are going to store what it's in new Component to a variable
- we then are going to use the on() method - it takes two argument - 1) takes a string of the event you want to listen to which in our case is the success event, 2) and the other argument is a function of what you want to do with that event. for now lets just do an alert to test it
- and we also we want to do the error event
- and we also want to change the the static value of the data-clipboard-text to the shortUrl

```
import React from 'react';
import Clipboard from 'clipboard';

export default class LinksListItem extends React.Component{
  componentDidMount(){
    const clipboard = new Clipboard(this.refs.copy);
    clipboard.on('success', ()=>{
      alert('it worked!');
    }).on('error', ()=>{
      alert('please copy the link manually');
    });
  }
  render(){
    return(
      <div>
        <p>{this.props.url}</p>
        <p>{this.props.shortUrl}</p>
        <button ref="copy" data-clipboard-text={this.props.shortUrl}>Copy</button>
      </div>
    )
  }
}

LinksListItem.propTypes = {
  _id: React.PropTypes.string.isRequired,
  url: React.PropTypes.string.isRequired,
  userId: React.PropTypes.string.isRequired,
```

```
    shortUrl: React.PropTypes.string.isRequired,
  }
}
```

- over in the browser every time you hit the copy button you should get an alert box, and if you paste in the chrome tools console you should be able to see the url address
- we next want to do is clean it up, we will be using componentWillUnmount to do this - this fires before our component removes from the screen
- first we need to turn our clipboard variable to become a global variable so we attached the this.clipboard to it
- we then are going to use the destroy method to delete everything from the clipboard after being used so it doesn't slow our browser

```
import React from 'react';
import Clipboard from 'clipboard';

export default class LinksListItem extends React.Component{
  componentDidMount(){
    this.clipboard = new Clipboard(this.refs.copy);
    this.clipboard.on('success', ()=>{
      alert('it worked!');
    }).on('error', ()=>{
      alert('please copy the link manually');
    });
  }
  componentWillUnmount(){
    this.clipboard.destroy();
  }
  render(){
    return(
      <div>
        <p>{this.props.url}</p>
        <p>{this.props.shortUrl}</p>
        <button ref="copy" data-clipboard-text={this.props.shortUrl}>Copy</button>
      </div>
    )
  }
}
```

- we are going to be switching the button name of the text from copy to copied when the user hits the button

- first we are going to create state by using a constructor
- our default justCopied will false
- after that we want our boolean justCopied to switch to true when hit the button
- and want to setTimeout to one 1 second
- we then use the ternary operator to switch between copy and copied

```
import React from 'react';
import Clipboard from 'clipboard';

export default class LinksListItem extends React.Component{

  constructor(props){
    super(props);
    this.state = {
      justCopied : false
    }
  }

  componentDidMount(){
    this.clipboard = new Clipboard(this.refs.copy);
    this.clipboard.on('success', ()=>{
      this.setState({justCopied: true})

      setTimeout(()=>{
        this.setState({justCopied: false});
      },1000);
    }).on('error', ()=>{
      alert('please copy the link manually');
    });
  }

  componentWillUnmount(){
    this.clipboard.destroy();
  }

  render(){
    return(
      <div>
        <p>{this.props.url}</p>
        <p>{this.props.shortUrl}</p>
        <button ref="copy" data-clipboard-text={this.props.shortUrl}>
```

```
        {this.state.justCopied ? 'Copied' : 'Copy'}
      </button>
    </div>
  )
}
```

```
LinksListItem.propTypes = {
  _id: React.PropTypes.string.isRequired,
  url: React.PropTypes.string.isRequired,
  userId: React.PropTypes.string.isRequired,
  shortUrl: React.PropTypes.string.isRequired,
}
```

-