

39-short-link video

Styling Items and Empty Items

- lets go to LinksList.js and under renderLinksListItems() we are going to check if there are no links in the array

```
renderLinksListItems(){

  if(this.state.links.length === 0){

    return (

      <div>

        <p>No Links Found</p>

      </div>

    );

  }

  return this.state.links.map((link)=>{

    const shortUrl = Meteor.absoluteUrl(link._id);

    return <LinksListItem key={link._id} shortUrl = {shortUrl} {...link}/ >

  });

}
```

- lets create a file under styles/components called _item.scss and import this file in _main.css

```
@import './variables';

@import './base';

@import './components/boxed-view';

@import './components/button';

@import './components/header';

@import './components/page-content';

@import './components/checkbox';

@import './components/item';
```

- lets** go to _variables.scss and create a new section
- notice that we also create a new color that was used on other sections

```
//color

$brand-primary: #5b4681;
$grey: #dddddd;
$light-grey: #f9f9f9;


//spacing

$space: 1.4rem;
$site-max-width: 50rem;


// Font-size

$base-font-size: $space;


//form inputs

$input-border: $grey;
$input-spacing: 1rem;


// Boxed view

$boxed-view-overlay-bg: $grey;
$boxed-view-bg: white;


//button

$button-bg: $brand-primary;
$button-color: white;
$button-pill-color: $brand-primary;
$button-secondary-bg: $light-grey;
$button-secondary-color: #777777;
$button-secondary-border: $grey;


//header

$header-bg: $brand-primary;
$header-color: white;


//item

$item-bg: $light-grey;
$item-border: darken($light-grey, 8%);
```

- now lets use these variables over in the _item.scss

```
.item {  
  background: $item-bg;  
  border: 1px solid $item-border;  
}
```

- **now** go back to LinkList.js to implement these new classes

```
renderLinksListItems(){  
  
  if(this.state.links.length === 0){  
    return (  
      <div className="item">  
        <p>No Links Found</p>  
      </div>  
    );  
  }  
}
```

- and we are going to be implementing this class in LinksListItem.js as well

```
render(){  
  return(  
    <div className="item">  
      <p>{this.props.url}</p>  
      <p>{this.props.shortUrl}</p>  
      <p>{this.props.visible.toString()}</p>  
      {this.renderStats()}  
      <a className="button button--pill button--link" href={this.props.shortUrl} target= "  
_blank">  
        Visit  
      </a>  
      <button className="button button--pill" ref="copy" data-clipboard-text={this.pro  
ps.shortUrl}>  
        {this.state.justCopied ? 'Copied' : 'Copy'}  
      </button>  
      <button className="button button--pill" onClick={()=>{  
        Meteor.call('links.setVisibility', this.props._id, !this.props.visible);
```

```

        >{this.props.visible ? 'Hide' : 'Unhide'}</button>
      </div>
    )
  }
}

```

- lets go back to our_item.scss partial

```

.item {
  background: $item-bg;
  border: 1px solid $item-border;
  margin-bottom: $space;
  padding: $space;
}

```

- lets go back to LinkListItem.js to change some things around
- right under render i have removed the 3rd paragraph because it's no longer needed, and the first paragraph got switched over to h2 tags

```

render(){
  return(
    <div className="item">
      <h2>{this.props.url}</h2>
      <p>{this.props.shortUrl}</p>
      {this.renderStats()}
      <a className="button button--pill button--link" href={this.props.shortUrl} target=
t= "_blank">
        Visit
      </a>
    </div>
  )
}

```

- head over to _base.scss lets modify h2

```

h2 {
  font-size: $large-font-size;
  margin-bottom: $space;
}

a {

```

```

    color: black;
}

input[type=text], input[type=email], input[type=password] {
    border: 1px solid $input-border;
    font-size: $base-font-size;
    margin-bottom: $space;
    padding: $input-spacing;
}

```

- lets now define a new variable for \$large-font-size in _variables.scss

```

//color
$brand-primary: #5b4681;
$grey: #dddddd;
$light-grey: #f9f9f9;

//spacing
$space: 1.4rem;
$site-max-width: 50rem;

// Font-size
$base-font-size: $space;
$large-font-size : 1.8rem;

//form inputs
$input-border: $grey;
$input-spacing: 1rem;

```

- lets go to _item.scss and create an element selector

```

.item {
    background: $item-bg;
    border: 1px solid $item-border;
    margin-bottom: $space;
    padding: $space;
}

```

```
.item__message {  
    color: $item-text-color;  
}
```

- lets go and define that new variable in `_variables.scss`

```
//color  
$brand-primary: #5b4681;  
$grey: #dddddd;  
$light-grey: #f9f9f9;  
$dark-grey: #777777;  
  
//spacing  
$space: 1.4rem;  
$site-max-width: 50rem;  
  
// Font-size  
$base-font-size: $space;  
$large-font-size : 1.8rem;  
  
//form inputs  
$input-border: $grey;  
$input-spacing: 1rem;  
  
// Boxed view  
$boxed-view-overlay-bg: $grey;  
$boxed-view-bg: white;  
  
//button  
$button-bg: $brand-primary;  
$button-color: white;  
$button-pill-color: $brand-primary;  
$button-secondary-bg: $light-grey;  
$button-secondary-color: $dark-grey;  
$button-secondary-border: $grey;  
  
//header
```

```

$header-bg: $brand-primary;

$header-color: white;

//item

$item-bg: $light-grey;

$item-border: darken($light-grey, 8%);

$item-text-color: $dark-grey;

```

- lets go apply the .item__message class over in LinkListItem.js

```

renderStats(){
  const visitMessage = this.props.visitedCount === 1 ? 'visit' : 'visits';
  let visitedMessage = null;
  const momentNow = moment(this.props.lastVisitedAt);

  if(typeof this.props.lastVisitedAt === 'number'){
    visitedMessage = `(visited ${momentNow.fromNow()})`;
  }

  return <p className="item__message">{this.props.visitedCount} {visitMessage} {visitedMess
age}</p>
}

render(){
  return(
    <div className="item">
      <h2>{this.props.url}</h2>
      <p className="item__message">{this.props.shortUrl}</p>
      {this.renderStats()}
      <a className="button button--pill button--link" href={this.props.shortUrl} target=
t= "_blank">
        Visit
      </a>
      <button className="button button--pill" ref="copy" data-clipboard-text={this.pro
ps.shortUrl}>
        {this.state.justCopied ? 'Copied' : 'Copy'}

```

```

        </button>

        <button className="button button--pill" onClick={()=>{
            Meteor.call('links.setVisibility', this.props._id, !this.props.visible);
        }}>{this.props.visible ? 'Hide' : 'Unhide'}</button>
    </div>

    )
}
}

```

- now lets create a new selector on _item.scss

```

.item {
    background: $item-bg;
    border: 1px solid $item-border;
    margin-bottom: $space;
    padding: $space;
}

.item__message {
    color: $item-text-color;
}

.item__status-message {
    color: $item-text-color;
    font-style: italic;
    text-align: center;
    margin: 0;
}

```

- lets now apply this class to LinksList.js

```

renderLinksListItems(){

    if(this.state.links.length === 0){
        return (
            <div className="item">
                <p className="item__status-message">No Links Found</p>
            </div>

```



```
    );  
  }
```

- lets now fix our pill buttons, head over to _button.scss

```
.button--link {  
  display: inline-block;  
  text-decoration: none;  
}  
  
.button--pill {  
  background-color: transparent;  
  border: 1px solid $button-pill-color;  
  color: $button-pill-color;  
  margin: 0 $space 0 0;  
  padding: .3rem .8rem;  
}
```

- finally we fixed the margin spacing between the pill button

```
.button--pill {  
  background-color: transparent;  
  border: 1px solid $button-pill-color;  
  color: $button-pill-color;  
  margin: 0 ($space/2) 0 0;  
  padding: .3rem .8rem;  
}
```