

153-Styling the NoteList video

- lets start by deleting all of the notes that we have in our app, so just click delete on each note
- go to NoteListEmptyItem.js to start modifying it, we are going to be deleting the div and leave the paragraph and also add a class

```
import React from 'react';

const NoteListEmptyItem = ()=>{

  return (

    <p className="empty-item">Create a note to get started</p>

  );

};

export default NoteListEmptyItem;
```

- now go to _item.scss to create that class selector

```
// Item list

.item-list{

  background-color: white;

  border: 1px solid $grey;

  overflow-y: scroll;

  width: 100%;

}


// Item

.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;
}

//Empty Item
.empty-item {
}

```

- **now lets start styling**

```

//Empty Item
.empty-item {
  font-style: italic;
  padding: $space;
  text-align: center;
}

```

- **now lets remove the text that says NoteList and that is sitting in NoteList.js**

```

export const NoteList = (props)=>{
  return (
    <div className="item-list">
      <NoteListHeader/>
      {props.notes.length === 0 ? <NoteListEmptyItem /> : undefined }
      { props.notes.map((note)=>{

```

```

        return <NoteListItem key={note._id} note={note}/>
      )}}

      NoteList { props.notes.length }

    </div>

  );
};

```

- so it should look like this

```

export const NoteList = (props)=>{
  return (
    <div className="item-list">
      <NoteListHeader/>
      {props.notes.length === 0 ? <NoteListEmptyItem /> : undefined }
      { props.notes.map((note)=>{
        return <NoteListItem key={note._id} note={note}/>
      })}
    </div>
  );
};

```

- now we are going to style the NoteListHeader, so in _item.scss lets create another selector

```

// Item list
.item-list{
  background-color: white;
  border: 1px solid $grey;
  overflow-y: scroll;
  width: 100%;
}

.item-list__header {

}

```

- now we can go an apply this new class to NoteListHeader.js, and also the button class to the button

```
import React from 'react';

import { Meteor } from 'meteor/meteor';

import { createContainer } from 'meteor/react-meteor-data';

import { Session } from 'meteor/session';

import PropTypes from 'prop-types';

export const NoteListHeader = (props)=>{

  return (

    <div className="item-list__header">

      <button className="button" onClick = {()=>={

        props.meteorCall('notes.insert',(err, res)=>{

          if(res){

            props.Session.set('selectedNoteId', res)

          }

        }}>Create Note</button>

    </div>

  );

};
```

- go back to _item.scss to style this new class

```
// Item list

.item-list{

  background-color: white;

  border: 1px solid $grey;

  overflow-y: scroll;

  width: 100%;

}
```

```
.item-list__header {  
    display: flex;  
    flex-direction: column;  
    padding: $space;  
}
```

- now delete .item__message and .item__status-message and everything inside .item, so _item.scss should look like this

```
// Item list  
  
.item-list{  
    background-color: white;  
    border: 1px solid $grey;  
    overflow-y: scroll;  
    width: 100%;  
}  
  
.item-list__header {  
    display: flex;  
    flex-direction: column;  
    padding: $space;  
}  
  
// Item  
  
.item {  
  
}  
  
//Empty Item  
  
.empty-item {  
    font-style: italic;  
    padding: $space;  
    text-align: center;
```

```
}
```

- also in `_variables.scss` lets delete the unused variables in the item section

```
//header

$header-bg: $brand-primary;
$header-color: white;
$header-height: 6rem;


//item


//Page Content

$page-content-sidebar-width: 30rem;
$page-content-main-width: calc(#{ $site-max-width } - #{ $page-content-sidebar-width });
$page-content_height: calc(100vh - #{ $header-height });
```

- go back to `_item.scss` to start styling

```
// Item list

.item-list{

    background-color: white;

    border: 1px solid $grey;

    overflow-y: scroll;

    width: 100%;

}


.item-list__header {

    display: flex;

    flex-direction: column;

    padding: $space;

}


// Item
```

```

.item {

  cursor: pointer;

}

//Empty Item

.empty-item {

  font-style: italic;

  padding: $space;

  text-align: center;

}

```

- now lets implement this class in NoteListItem.js

```

export const NoteListItem = (props) => {

  return (

    <div className="item" onClick={()=>{

      props.Session.set('selectedNoteId', props.note._id);

    }}>

      <h5>{ props.note.title || 'Untitled note' }</h5>

      { props.note.selected ? 'selected': undefined }

      <p>{ moment(props.note.updatedAt).format('M/DD/YY')}</p>

    </div>

  );

};

```

- go back to _items.scss

```

// Item

.item {

  cursor: pointer;

  padding: $space;

}

```

- now lets style item when is hovered

```

.item-list__header {
    display: flex;
    flex-direction: column;
    padding: $space;
}

// Item

.item {
    cursor: pointer;
    padding: $space;
}

.item:hover {
    background-color: $light-grey;
    border-left: 4px solid $grey;
}

```

- now lets go to `_variables.scss` to create a new variable for the 4px because it will get used more than once

```

$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;

$header-height: 6rem;

//item

$item-border-width: 4px;

```



```
//Page Content

$page-content-sidebar-width: 30rem;

$page-content-main-width: calc(#{ $site-max-width } - #{ $page-content-sidebar-width });

$page-content_height: calc(100vh - #{ $header-height });
```

- now go back to _item.scss to use this new variable

```
.item:hover {

    background-color: $light-grey;

    border-left: $item-border-width solid $grey;

}
```

- there is an issue when you hover each item in the list, it moves a little bit, so we are going to fix that - go to _item.scss

```
// Item

.item {

    border-left: $item-border-width transparent solid;

    cursor: pointer;

    padding: $space;

}

.item:hover {

    background-color: $light-grey;

    border-left: $item-border-width solid $grey;

}
```

- and now I no longer get that shift when hovering the list
- lets make the note transitioning a little bit better - go to _items.scss

```
// Item

.item {

    border-left: $item-border-width transparent solid;
```

```

    cursor: pointer;

    padding: $space;

    transition: background-color .3s ease;
  }

```

- but now the border is not transitioning with ease so lets fix that

```

// Item
.item {

  border-left: $item-border-width transparent solid;

  cursor: pointer;

  padding: $space;

  transition: background-color .3s ease, border-left .3s ease;
}

```

- now we are going to style the individual items, and this is going to be based on whether props.selected is true or false, so we need to have a dynamic class for that, so in NoteListItem.js create a const variable with a ternary then we can add this variable to the className, if it is selected we are className variable item item--selected if it's not then className will have the value item - lets also delete the line below

```

import React from 'react';
import moment from 'moment';
import { Session } from 'meteor/session';
import { createContainer } from 'meteor/react-meteor-data';
import PropTypes from 'prop-types';

export const NoteListItem = (props) => {

  const className = props.note.selected ? 'item item--selected' : 'item';

  return (

    <div className={className} onClick={()=>{
      props.Session.set('selectedNoteId', props.note._id);
    }}>

      <h5>{ props.note.title || 'Untitled note' }</h5>

      { props.note.selected ? 'selected': undefined } <--DELETE THIS LINE

      <p>{ moment(props.note.updatedAt).format('M/DD/YY')}</p>

```

```

        </div>

    );
};

```

- Now lets go create the modifier selector `item--selected` in `_item.scss` - and we also gave the same styles to `.item--selected: hover`

```

.item: hover {

    background-color: $light-grey;

    border-left: $item-border-width solid $grey;

}

.item--selected, .item--selected: hover{

    background-color: $light-grey;

    border-left: $item-border-width solid $brand-primary;

}

```

- lets go to `NoteListItem.js` to implement more `classNames`

```

import React from 'react';

import moment from 'moment';

import { Session } from 'meteor/session';

import { createContainer } from 'meteor/react-meteor-data';

import PropTypes from 'prop-types';

export const NoteListItem = (props) => {

    const className = props.note.selected ? 'item item--selected' : 'item';

    return (

        <div className={className} onClick={()=>{

            props.Session.set('selectedNoteId', props.note._id);

        }}>

            <h5 className="item__title">{ props.note.title || 'Untitled note' }</h5>

```

```

        <p className = "item__subtitle">{ moment(props.note.updatedAt).format('M/DD/YY')}
    </p>

    </div>

    );
};

```

- now lets go back to _item.scss to add those two new classes, and we also want to style .item__title when .item--selected is selected

```

.item__title{
    font-weight: normal;
}

.item--selected .item__title {
    font-weight: bold;
    font-size: $base-font-size;
    margin-bottom: .4rem;
}

.item__subtitle {
    color: $dark-grey;
    margin: 0;
}

//Empty Item
.empty-item {
    font-style: italic;
    padding: $space;
    text-align: center;
}

```

- now its time to commit and push, so go to your terminal

```
joses-MacBook-Pro:notes mendoza$ git commit -a -m "Style note list"
```

```
joses-MacBook-Pro:notes mendoza$ git push
```

-