

131/132-Adding and Counting Notes [video](#)

** when importing Meteor make sure to add { Meteor }*

- we are going to create two new components - NoteList and NoteListHeader in the imports/ui folder
- NoteListHeader is going to be rendered inside on NoteList, so lets set NoteList.js first-lets start importing what we are going to need
- next we want to create a stateless functional component, and we are going to pass props to it

```
import React from 'react';

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

export const NoteList = (props)=>{

  return (

  );

};
```

- lets also set the containerized of NoteList- createContainer takes two arguments - a function and the component we want to render

```
import React from 'react';

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

export const NoteList = (props)=>{

  return (

    <div>


    </div>

  );

};

export default createContainer(()=>{

}, NoteList);
```

- from the container we want to fetch the notes- we need to do two things - 1) subscribe the subscription we setup in notes.js to the Meteor.publish
- so lets start by importing first, and then we can Meteor.subscribe in our code, passing in the name of our subscription which is 'notes', just like in notes.js

```
import React from 'react';

import Meteor from 'meteor/meteor';

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

export const NoteList = (props)=>{

  return (

    <div>


    </div>

  );
};

export default createContainer(()=>{

  Meteor.subscribe('notes');

}, NoteList);
```

- and now we can actually start fetching data off of our database, we are going to return an object, and we are going to setup a prop called notes, and the value is going to be our api, so lets import notes

```
import React from 'react';

import Meteor from 'meteor/meteor';

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

import { Notes } from '../api/notes'

export const NoteList = (props)=>{

  return (

    <div>


    </div>

  );
```

```

    );
  };

export default createContainer(()=>{
  Meteor.subscribe('notes');

  return {
    notes :
  }

}, NoteList);

```

- now we can access to Notes.find, this will return all notes this user has access to

```

export default createContainer(()=>{
  Meteor.subscribe('notes');

  return {
    notes : Notes.find().fetch()
  }

}, NoteList);

```

- now in our div up above lets dump the length of our notes prop

```

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

import { Notes } from '../api/notes'

export const NoteList = (props)=>{
  return (
    <div>

```

```

        NoteList { props.notes.length }

      </div>

    );
  };

  export default createContainer(()=>{
    Meteor.subscribe('notes');

    return {
      notes : Notes.find().fetch()
    }

  }, NoteList);

```

- lets set up propTypes as well, and it is going to be a required array

```

NoteList.propTypes = {
  notes : React.PropTypes.array.isRequired
}

export default createContainer(()=>{
  Meteor.subscribe('notes');

  return {
    notes : Notes.find().fetch()
  }

}, NoteList);

```

- now save it and lets put it into practice in our Dashboard.js - because this is where NoteList is going to get rendered,
- so in Dashboard.js lets import our Default export NoteList

```

import React from 'react';

```

```

import PrivateHeader from './PrivateHeader';

import NoteList from './NoteList';

export default () => {

  return (

    <div>

      <PrivateHeader title="Dashboard"/>

      <div className="page-content">

        Dashboard page content.

      </div>

    </div>

  );

}

```

- now lets make an instance of <NoteList/>

```

import React from 'react';

import PrivateHeader from './PrivateHeader';

import NoteList from './NoteList';

export default () => {

  return (

    <div>

      <PrivateHeader title="Dashboard"/>

      <div className="page-content">

        <NoteList/>

      </div>

    </div>

  );

}

```

- down below I had to comment out the Meteor.subscribe, because it was giving me an error of Meteor.subscribe is not a function, but when I commented out the app worked fine - *solution - when I imported Meteor up above I didn't use { Meteor }, I was using just Meteor*

```
import React from 'react';

import Meteor from 'meteor/meteor';

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

import { Notes } from '../api/notes'

export const NoteList = (props)=>{

  return (

    <div>

      NoteList { props.notes.length }

    </div>

  );

};

NoteList.propTypes = {

  notes : React.PropTypes.array.isRequired

};

export default createContainer(()=>{

  // Meteor.subscribe('notes');

  return {

    notes : Notes.find().fetch()

  };

}, NoteList);
```

- on the browser we should see NoteList 0
- no over at server/main.js lets import notes

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

import { WebApp } from 'meteor/webapp';
```

```
import moment from 'moment';

import '../imports/api/users';
import '../imports/api/notes';
import '../imports/startup/simple-schema-configuration.js';

Meteor.startup(() => {

});
```

- now lets go to NoteListHeader and import some stuff
- lets export our const, and this is just going to render a button to the screen, the button is going to call our meteorCall down below and that will trigger our notes.insert method that is located in notes.js

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

export const NoteListHeader = (props)=>{
  return (
    <div>
      <button onClick = {()}=>{
        props.meteorCall('notes.insert');
      }>Create Note</button>
    </div>
  );
};

export default createContainer(()=>{
  return {
    meteorCall : Meteor.call
  };
},NoteListHeader);
```

- now over at NoteList, lets render our NoteListHeader component

```
import React from 'react';

import { Meteor } from 'meteor/meteor';

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

import { Notes } from '../api/notes'

import NoteListHeader from './NoteListHeader';

export const NoteList = (props)=>{

  return (

    <div>

      NoteList { props.notes.length }

      <NoteListHeader/>

    </div>

  );

};
```

- now on the browser you should see a button and everytime is pushed the NoteList counter should increment
-