

140-Editor Input and TextArea

- go to Editor.js and lets add three tags to the return statement if there is a note

```
import React from 'react';

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

import { Session } from 'meteor/session';

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

import PropTypes from 'prop-types'

export class Editor extends React.Component{

  render(){

    if(this.props.note){

      return (

        <div>

          <input/>

          <textarea/>

          <button/>

        </div>

      );

    }

  }

}
```

- then we want to give the value of the text area to note.body and also a placeholder, and also an onChange handler

```
import React from 'react';

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

import { Session } from 'meteor/session';

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

import PropTypes from 'prop-types'

export class Editor extends React.Component{

  render(){

    if(this.props.note){

      return (


```

```

        <div>

            <input/>

            <textarea value={this.props.note.body} placeholder="Your note here" onChange=
{this.handleBodyChange.bind(this)}></textarea>

            <button></button>

        </div>

    );

```

- lets now create the handleBodyChange method above - this is an event callback function, we are going to pass the e, and we are going to be able to access the target which references the dom element where the event was fired from, in this case was the textarea

```

import React from 'react';
import { createContainer } from 'meteor/react-meteor-data';
import { Session } from 'meteor/session';
import { Notes } from '../api/notes';
import PropTypes from 'prop-types'

export class Editor extends React.Component{

    handleBodyChange(e){

    }

    render(){
        if(this.props.note){
            return (
                <div>

                    <input/>

                    <textarea value={this.props.note.body} placeholder="Your note here" onChange=
{this.handleBodyChange.bind(this)}></textarea>

                    <button></button>

                </div>

            );

```

- in this method we want to call the notes.update method that is located in notes.js, notes.update passes two arguments, _id and updates
- start by importing Meteor and down at the bottom we want to setup a call prop to setup Meteor.call

```
import React from 'react';

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

import { Session } from 'meteor/session';

import { Meteor } from 'meteor/meteor';

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

import PropTypes from 'prop-types'

export class Editor extends React.Component{

  handleBodyChange(e){

  }

  render(){
    if(this.props.note){
      return (
        <div>

          <input/>

          <textarea value={this.props.note.body} placeholder="Your note here" onChange=
{this.handleBodyChange.bind(this)}></textarea>

          <button></button>

        </div>

      );
    }else{
      return (
        <p>

          {this.props.selectedNoteId ? 'Note not found' : 'Pick or create a note to ge
t started'}

        </p>

      );
    }
  }
}
```

```

    }
  }
};

Editor.propTypes = {
  note : PropTypes.object,
  selectedNoteId : PropTypes.string,
}

export default createContainer(()=>{
  const selectedNoteId = Session.get('selectedNoteId');

  return {
    selectedNoteId,
    note: Notes.findOne(selectedNoteId),
    call: Meteor.call
  };
}, Editor)

```

- now we can use this call prop in our method above so we can use it to call our notes.update method from notes.js, and we are going to pass in the two arguments it needs, first is the string _id and the second one is our object that contains the body and all that, and the body will target the value of the textarea

```

import React from 'react';
import { createContainer } from 'meteor/react-meteor-data';
import { Session } from 'meteor/session';
import { Meteor } from 'meteor/meteor';
import { Notes } from '../api/notes';
import PropTypes from 'prop-types'

export class Editor extends React.Component{

  handleBodyChange(e){

    this.props.call('notes.update', this.props.note._id, {

```

```

        body: e.target.value

    })

}

```

- now in the browser you should be able to type in the textarea and when page refresh the content should be remain
- now we are going to handle the input, which is going to be handling the title of the note, and we are going to be using the same steps as the body

```

import React from 'react';

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

import { Session } from 'meteor/session';

import { Meteor } from 'meteor/meteor';

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

import PropTypes from 'prop-types'

export class Editor extends React.Component{

  handleBodyChange(e){

    this.props.call('notes.update', this.props.note._id, {

      body: e.target.value

    });

  }

  handleTitleChange(e){

    this.props.call('notes.update', this.props.note._id, {

      title: e.target.value

    })

  }

  render(){

    if(this.props.note){

```

```

    return (
      <div>
        <input value={this.props.note.title} placeholder="Untitled" onChange={this.handleTitleChange.bind(this)} />
        <textarea value={this.props.note.body} placeholder="Your note here" onChange={this.handleBodyChange.bind(this)} />
        <button />
      </div>
    );
  } else {

```

- and this should be able to update the title as soon as you start typing and it should not go away when you refresh
- lets commit and push our code in our terminal

```
joses-MacBook-Pro:notes mendoza$ git add .
```

```
joses-MacBook-Pro:notes mendoza$ git commit -m "Setup editor to change title/body"
```

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

-
-