

# 01-Short-Link-App

## Setting Up React-Router

- lets create a new meteor react app
- go to your terminal and type the following

```
meteor create short-lnk --release 1.4.2.3
```

- navigate to that new directory
- install meteor
- and then just type meteor

```
cd short-lnk  
  
meteor npm install  
  
meteor
```

- now open up that folder in your editor navigate to main.html and delete a few things to make it look like this

```
<head>  
  <title>Short Lnk</title>  
</head>  
  
<body>  
  <div id="app"></div>  
</body>
```

- go to client main.js, in there you will find a lot of blaze code, go ahead and delete everything
- lets go ahead and install React from the terminal

```
meteor npm install react@15.4.1 react-dom@15.4.1 react-router@3.0.0 --save
```

- lets go client main.js and start importing what we need
- first import the name export { Meteor }
- then the default export React and ReactDOM
- we then going to create SignUp class that extends from React
- then we will use the render() method
- for now it will only return static html
- in order to render our component we have to wait for Meteor to start up by calling Meteor.startup

- that takes a call back
  - in Meteor.startup we will be using ReactDOM.render which takes two arguments-the jsx to render and the location to render it to, in our case we'll be using an instance of signup <SignUp> and the second argument is document.getElementById('app');

```
import { Meteor } from 'meteor/meteor';
import React from 'react';
import ReactDOM from 'react-dom';

class Signup extends React.Component{
  render() {
    return <p>sign up component here</p>;
  }
}

Meteor.startup(=>{
  ReactDOM.render(<Signup/>, document.getElementById('app'));
});
```

- we are now going to start breaking the app into different section
- first lets create a new folder in the root folder called Import, inside there another folder called ui, and then create a file in ui called Signup.js
- cut and paste the Signup class from main.js to Signup.js and import React
- export our class
- and import the class in Main.js
- This is what Signup.js should look like

```
import React from 'react';

export default class Signup extends React.Component{
  render(){
    return <p>sign up component here</p>;
  }
}
```

- and this is how client/Main.js should look like now

```
import { Meteor } from 'meteor/meteor';
import React from 'react';
import ReactDOM from 'react-dom';
```

```
import Signup from './imports/ui/Signup';
```

```
Meteor.startup(()=>{  
  ReactDOM.render(<Signup/>, document.getElementById('app'));  
});
```

- we then are going to create another component in the ui folder called Link.js
- and we do the something by defining the class and exporting it and importing React

```
import React from 'react';  
  
export default class Link extends React.Component{  
  render(){  
    return <p>This paragraph is coming from the Link component</p>  
  }  
}
```

- we then import this component in client/main.js

```
import { Meteor } from 'meteor/meteor';  
import React from 'react';  
import ReactDOM from 'react-dom';  
  
import Signup from './imports/ui/Signup';  
import Link from './imports/ui/Link';
```

## Setting up Routers

- lets go to client/main.js and start importing some stuff for routing

```
import { Meteor } from 'meteor/meteor';  
import React from 'react';  
import ReactDOM from 'react-dom';  
import { Router, Route, browserHistory } from 'react-router';  
  
import Signup from './imports/ui/Signup';
```

```
import Link from './imports/ui/Link';
```

```
Meteor.startup(()=>{  
  ReactDOM.render(<Link />, document.getElementById('app'));  
});
```

- we then are going to create a const variable with some jsx in it
- first we are going to make instance of <Router> with a prop of history

```
import { Meteor } from 'meteor/meteor';  
import React from 'react';  
import ReactDOM from 'react-dom';  
import { Router, Route, browserHistory } from 'react-router';  
  
import Signup from './imports/ui/Signup';  
import Link from './imports/ui/Link';  
  
const routes = (  
  <Router history={browserHistory}>  
  
    </Router>  
);  
  
Meteor.startup(()=>{  
  ReactDOM.render(<Link />, document.getElementById('app'));  
});
```

- inside of Router all we need is instantiate Route and give it a path prop and a component prop
- the path will take a string and the component will be a js expression, it will be the component we want to render
- so the path will be /signup and the component will be {Signup}
- now all we have to do is initialize our routes variable is to pass it inside React.render as supposed to < Signup/>

```
import { Meteor } from 'meteor/meteor';  
import React from 'react';  
import ReactDOM from 'react-dom';  
import { Router, Route, browserHistory } from 'react-router';
```

```
import Signup from '../imports/ui/Signup';
import Link from '../imports/ui/Link';

const routes = (
  <Router history={browserHistory}>
    <Route path="/signup" component={Signup}/>
  </Router>
);

Meteor.startup(()=>{
  ReactDOM.render(routes, document.getElementById('app'));
});
```

- So now in the url if you type localhost:3000/signup you will see our Signup component.
- if you just type localhost:3000 you won't see anything, and you will get an error on the console
- now we are gonna be doing the something with Link component with a different path

```
import { Meteor } from 'meteor/meteor';
import React from 'react';
import ReactDOM from 'react-dom';
import { Router, Route, browserHistory } from 'react-router';

import Signup from '../imports/ui/Signup';
import Link from '../imports/ui/Link';

const routes = (
  <Router history={browserHistory}>
    <Route path="/signup" component={Signup}/>
    <Route path="/links" component={Link}/>
  </Router>
);

Meteor.startup(()=>{
  ReactDOM.render(routes, document.getElementById('app'));
});
```

- now we want to create a base route (landing page) and a 404 in case the user types other than /links, /, or /signup
- lets create the 404 file first, in the ui folder create a component called NotFound.js

```
import React from 'react';

export default class NotFound extends React.Component{

  render(){

    return <p>NotFound Component here</p>;

  }

}
```

- now lets go back to client main.js and create another route
- but for this one instead of using a regular path like we did on the other ones, we are going to use a \* pattern
- and this is going to match all the unmatched routes, so this is going to match with /signup and /links, but if they don't match that it will take me NotFound component

```
import { Meteor } from 'meteor/meteor';
import React from 'react';
import ReactDOM from 'react-dom';
import { Router, Route, browserHistory } from 'react-router';

import Signup from '../imports/ui/Signup';
import Link from '../imports/ui/Link';
import NotFound from '../imports/ui/NotFound';

const routes = (

  <Router history={browserHistory}>

    <Route path="/signup" component={Signup}/>

    <Route path="/links" component={Link}/>

    <Route path="*" component={NotFound}/>

  </Router>

);

Meteor.startup(()=>{

  ReactDOM.render(routes, document.getElementById('app'));

});
```

- now we are going to create the Login component, making sure you put it on top of the other routes, because that will be the first route the user will see

```

import { Meteor } from 'meteor/meteor';
import React from 'react';
import ReactDOM from 'react-dom';
import { Router, Route, browserHistory } from 'react-router';

import Signup from '../imports/ui/Signup';
import Link from '../imports/ui/Link';
import NotFound from '../imports/ui/NotFound';
import Login from '../imports/ui/Login'

const routes = (
  <Router history={browserHistory}>
    <Route path="/" component={Login}/>
    <Route path="/signup" component={Signup}/>
    <Route path="/links" component={Link}/>
    <Route path="*" component={NotFound}/>
  </Router>
);

Meteor.startup(()=>{
  ReactDOM.render(routes, document.getElementById('app'));
});

```

- and this is the Login component

```

import React from 'react';

export default class Login extends React.Component{
  render(){
    return <p>Login component here</p>;
  }
}

```

- to learn more about react router, google react-router, you should be able to research more on react routing