

22-short-Ink-app [video](#)

Stateless Functional Components

- we are now going to learn an alternative way to make components
- lets create an example in client/main.js
- right here the function we just defined acts like a render() method

```
const MyComponent= () =>{  
  
}
```

so we just use the return, and also imported React

```
import { Meteor } from 'meteor/meteor';  
import ReactDOM from 'react-dom';  
import { Tracker } from 'meteor/tracker';  
  
import { routes, onAuthChange } from '../imports/routes/routes';  
import '../imports/startup/simple-schema-configuration.js';  
  
Tracker.autorun(()=>{  
  const isAuthenticated = !!Meteor.userId();  
  onAuthChange(isAuthenticated);  
});  
  
//stateless functional component  
import React from 'react';  
const MyComponent= () =>{  
  return(  
    <div>  
      <h1>MyComponent is here!</h1>  
    </div>  
  );  
}  
  
Meteor.startup(()=>{  
  ReactDOM.render(<MyComponent />, document.getElementById('app'));  
});
```

- since this is a stateless component we don't have access to state, but we do have access to props
- so let's pass a string to MyComponent and we would pass it in the function as a props argument

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

import { routes, onAuthChange } from '../imports/routes/routes';
import '../imports/startup/simple-schema-configuration.js';

Tracker.autorun(()=>{
  const isAuthenticated = !!Meteor.userId();
  onAuthChange(isAuthenticated);
});

//stateless functional component
import React from 'react';
const MyComponent= (props) =>{

  return(
    <div>
      <h1>MyComponent is here! {props.name}</h1>
    </div>
  );
}

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

- this would be called a presentational component which means that it's only job to display something very basic to the screen on the other hand
- LinkList component is a container component, it is more complex because it has a state
- so we would use these presentational components because they are much faster than es6 class components - it is also a lot easier to read and maintain
- let's change back the main/client.js how it was before

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

```
import { routes, onAuthChange } from '../imports/routes/routes';
import '../imports/startup/simple-schema-configuration.js';

Tracker.autorun(()=>{
  const isAuthenticated = !!Meteor.userId();
  onAuthChange(isAuthenticated);
});

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

- so now lets change the Link component so it becomes a presentational component
- and this is how Link.js should look like now

```
import React from 'react';
import { Accounts } from 'meteor/accounts-base'
import { Meteor } from 'meteor/meteor';
import { Links } from '../api/links';
import LinksList from './LinksList';
import PrivateHeader from './PrivateHeader';
import AddLink from './AddLink';

export default () => {
  return (
    <div>
      <PrivateHeader title="Your Links"/>
      <LinksList/>;
      <AddLink/>
    </div>
  );
}
```

- we are now going to change PrivateHeader Component
- we are going to define a const variable because we are going to use propTypes
- and also we changed the function on the button to a shorthand arrow function
- and at the bottom we export it our component

```

import React from 'react';
import { Accounts } from 'meteor/accounts-base';

const PrivateHeader = (props)=>{
  return (
    <div>
      <h1>{props.title}</h1>
      <button onClick={() => Accounts.logout()}>Logout</button>
    </div>
  );
}

PrivateHeader.propTypes = {
  title: React.PropTypes.string.isRequired
};

export default PrivateHeader;

```

- now we are going to change the NotFound.js Component to a presentational component

```

import React from 'react';

export default ()=>{
  return <p>NotFound Component here</p>;
}

```

-