

156-Mobile Menu: Part 1 video

- first lets create a new Session variable, go to main.js

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

import ReactDOM from 'react-dom';

import { Tracker } from 'meteor/tracker';

import { Session } from 'meteor/session';

import { browserHistory } from 'react-router';

import { routes, onAuthChange } from '../imports/routes/routes';

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

Tracker.autorun(()=>{

  const isAuthenticated = !!Meteor.userId();

  const currentPagePrivacy = Session.get('currentPagePrivacy');

  onAuthChange(isAuthenticated, currentPagePrivacy);

});

Tracker.autorun(()=>{

  const selectedNoteId = Session.get('selectedNoteId')

  if(selectedNoteId){

    browserHistory.replace(`/dashboard/${selectedNoteId}`)

  }

});

Meteor.startup(()=>{

  Session.set('selectedNoteId', undefined)

  Session.set('isNavOpen', false);

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

});
```

- next download the images from this course, we will need bars.svg, favicon.png, and x.svg, make sure to create a folder called public, and in there create another folder called images, that is where you will put these images
- now lets go to PrivateHeader.js to setup an image tag

```
import React from 'react';

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

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

import PropTypes from 'prop-types';

export const PrivateHeader = (props)={>{

  return (

    <div className="header">

      <div className="header__content">

        

        <h1 className="header__title">{props.title}</h1>

        <button className="button button--link-text " onClick={()=>

props.handleLogout()}>Logout</button>

      </div>

    </div>

  );

}
```

- now we are going to pass a prop to our createContainer called isNavOpen

```
export default createContainer(()=>{

  return {

    handleLogout: () => Accounts.logout(),

    isNavOpen : Session.get('isNavOpen')

  }

}, PrivateHeader);
```

- now import Session and setup a prop type

```
import React from 'react';

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

```

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

import PropTypes from 'prop-types';

import { Session } from 'meteor/session';

export const PrivateHeader = (props)=>{

  return (

    <div className="header">

      <div className="header__content">

        

        <h1 className="header__title">{props.title}</h1>

        <button className="button button--link-text " onClick={()=>
props.handleLogout()}>Logout</button>

      </div>

    </div>

  );
}

PrivateHeader.propTypes = {

  title: PropTypes.string.isRequired,

  handleLogout: PropTypes.func.isRequired,

  isNavOpen: PropTypes.bool.isRequired

};

export default createContainer(()=>{

  return {

    handleLogout: () => Accounts.logout(),

    isNavOpen : Session.get('isNavOpen')

  }

}, PrivateHeader);

```

- next we need to create a variable for "/images/bars.svg" because we need to dynamically change it, we are going to be using a ternary operator to change it to X or bars

```

import React from 'react';

```

```

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

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

import PropTypes from 'prop-types';

import { Session } from 'meteor/session';

export const PrivateHeader = (props)=>{

  const navImageSrc = props.isNavOpen ? '/images/x.svg' : '/images/bars.svg';

  return (

    <div className="header">

      <div className="header__content">

        

        <h1 className="header__title">{props.title}</h1>

        <button className="button button--link-text " onClick={()=>
props.handleLogout()}>Logout</button>

      </div>

    </div>

  );
}

```

- and now we can use that variable in the image tag src

```

import React from 'react';

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

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

import PropTypes from 'prop-types';

import { Session } from 'meteor/session';

export const PrivateHeader = (props)=>{

  const navImageSrc = props.isNavOpen ? '/images/x.svg' : '/images/bars.svg';

  return (

    <div className="header">

      <div className="header__content">

        <img src={navImageSrc}/>


```

```

        <h1 className="header__title">{props.title}</h1>

        <button className="button button--link-text " onClick={()=>
props.handleLogout()}>Logout</button>

    </div>

</div>

);
}

```

- now if you go to the dev tools console and type the following you should the bars turn to X

```
require('meteor/session').Session.set('isNavOpen', true)
```

- and we change it to false will turn the X to bars

```

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

export const PrivateHeader = (props) => {
    const navImageSrc = props.isNavOpen ? '/images/x.svg' : '/images/bars.svg';
    return (
        <div className="header">
            <div className="header__content">
                <img onClick={()=>props.handleNavToggle()} src={navImageSrc}/>
                <h1 className="header__title">{props.title}</h1>
                <button className="button button--link-text " onClick={()=>
props.handleLogout()}>Logout</button>
            </div>
        </div>
    );
}

```

```

PrivateHeader.propTypes = {
  title: PropTypes.string.isRequired,
  handleLogout: PropTypes.func.isRequired,
  isNavOpen: PropTypes.bool.isRequired,
  handleNavToggle : PropTypes.func.isRequired
};

export default createContainer(()=>{
  return {
    handleLogout: () => Accounts.logout(),
    handleNavToggle : () => Session.set('isNavOpen', !Session.get('isNavOpen')),
    isNavOpen : Session.get('isNavOpen')
  }
}, PrivateHeader);

```

- now we are going to adding a new Tracker.autorun call in main.js - this is going to watch that Session variable

```

import { Meteor } from 'meteor/meteor';
import ReactDOM from 'react-dom';
import { Tracker } from 'meteor/tracker';
import { Session } from 'meteor/session';
import { browserHistory } from 'react-router';

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

Tracker.autorun(()=>{
  const isAuthenticated = !!Meteor.userId();
  const currentPagePrivacy = Session.get('currentPagePrivacy');
  onAuthChange(isAuthenticated, currentPagePrivacy);
});

```

```
Tracker.autorun(()=>{  
  const selectedNoteId = Session.get('selectedNoteId')  
  if(selectedNoteId){  
    browserHistory.replace(`/dashboard/${selectedNoteId}`)  
  }  
});
```

```
Tracker.autorun(()=>{  
  const isNavOpen = Session.get('isNavOpen');  
})
```

- we are going to be using a js method called `classList.toggle`

```
Tracker.autorun(()=>{  
  const isNavOpen = Session.get('isNavOpen');  
  document.body.classList.toggle('is-nav-open', isNavOpen);  
})
```

-