26-short-lnk-app video

Meteor Session Variables

- we are going to be user Meteor Session package, so we hide and show our links
- this only lives in the client side not the server side
- to find out more about this you can go to http://docs.meteor.com/api/session.html
- so lets go to the terminal to add this

```
meteor add session
```

- lets go to main.js to import it

```
import { Meteor } from 'meteor/meteor';
import ReactDom from 'react-dom';
import { Tracker } from 'meteor/tracker';
import { Session } from 'meteor/session';

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'));
});
```

- and now we can set and get values
- Session.set() takes two arguments 1)the key, 2) the value
- for example

```
import { Meteor } from 'meteor/meteor';
import ReactDom from 'react-dom';
import { Tracker } from 'meteor/tracker';
import { Session } from 'meteor/session';
import { routes, onAuthChange } from '../imports/routes/routes';
import '../imports/startup/simple-schema-configuration.js';

Tracker.autorun(()=>{
  const isAuthenticated = !!Meteor.userId();
  onAuthChange(isAuthenticated);
});
Session.set('name', 'Jose Mendoza')

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

- and also Session.get, it lets retreive that value if any
- so we'll store it in a variable and print it to the screen

```
import { Meteor } from 'meteor/meteor';
import ReactDom from 'react-dom';
import { Tracker } from 'meteor/tracker';
import { Session } from 'meteor/session';

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

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

Session.set('name', 'Jose Mendoza');
const name = Session.get('name');
console.log(name);

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

- over in the console chorme tools you should be able to see the name value

```
Jose Mendoza
```

- **let** setup a Tracker auto call up above and we are going to put the const name line and the console.log line

```
import { Meteor } from 'meteor/meteor';
import ReactDom from 'react-dom';
import { Tracker } from 'meteor/tracker';
import { Session } from 'meteor/session';

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

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

Tracker.autorun(()=>{
  const name = Session.get('name');
  console.log(name);
});

Session.set('name', 'Jose Mendoza');



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

- and now we should be able to see the name in the console
- but now lets try to set it to a different name but this time in the console

```
require('meteor/session').Session.set('name','Peter');
```

- and then we do Session.get , and we get Peter back

```
require('meteor/session').Session.get('name')
"Peter"
```

- now lets delete this example code now that we got some practice
- Your client/main.js should look like it did 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'));
});
```

- we are going to be using the Session package in LinksList.js
- so lets import it first

```
import React from 'react';
import { Meteor } from 'meteor/meteor';
import { Tracker } from 'meteor/tracker';
import { Session } from 'meteor/session';

import { Links } from '../api/links'
import LinksListItem from './LinksListItem';
```

- now over in the componentDidMount on Links.find we are going to be set visible to Session.get

```
import React from 'react';
import { Meteor } from 'meteor/meteor';
import { Tracker } from 'meteor/tracker';
import { Session } from 'meteor/session';


import { Links } from '../api/links'
import LinksListItem from './LinksListItem';


export default class LinksList extends React.Component{


    constructor(props){
        super(props);
        this.state = {
            links : []
        }
    }
    componentDidMount(){
        console.log('ComponentDidMount LinksList');
       this.linksTracker = Tracker.autorun(()=>{
           Meteor.subscribe('links');
            const links = Links.find({
                visible: Session.get('showVisible')
            }).fetch()
            this.setState({links});
          });
    }
```

- now over at client/main.js
- we need to set Session and also import Session

```
import { Meteor } from 'meteor/meteor';
import ReactDom from 'react-dom';
import { Tracker } from 'meteor/tracker';
import { Session } from 'meteor/session';


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


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

});


Meteor.startup(()=>{

  Session.set('showVisible', true)

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

});
```

- and now over in your browser should be able to see only the ones that are true
- as a challenge lets show the false ones over in the console in chrome tools

```
require('meteor/session').Session.set('showVisible', false);
```

- and by doing this you will only be able to see the ones that are false
-