13-short-lnk-app - video

Creating Publications and Subscriptions

- first lets create a second user, open up a different browser and create a new user
- now create new links
- currently all users can see each others links
- the reason is because all are getting synced on miniMongo
- the first thing to do is explore the packages that come built in with Meteor
- some of these packages are nableing the insecure behaviour we see here
- lets go to the terminal to explore these packages

```
joses-MacBook-Pro:short-lnk mendoza$ meteor list
```

- **and this is the list**
- **the** autopublish is responsible for nableing this insecure behaviour
- you can view this package like any other package on atmospherejs.com

```
accounts-password      1.4.1  Password support for accounts
autopublish            1.0.7  (For prototyping only) Publish the entire database to all clients
blaze-html-templates   1.1.2  Compile HTML templates into reactive UI with Meteor Blaze
ecmascript             0.8.3  Compiler plugin that supports ES2015+ in all .js files
es5-shim               4.6.15  Shims and polyfills to improve ECMAScript 5 support
insecure               1.0.7  (For prototyping only) Allow all database writes from the client
jquery                 1.11.10  Manipulate the DOM using CSS selectors
meteor-base            1.0.4* Packages that every Meteor app needs
mobile-experience      1.0.5  Packages for a great mobile user experience
mongo                  1.2.2  Adaptor for using MongoDB and Minimongo over DDP
reactive-var           1.0.11  Reactive variable
shell-server           0.2.4  Server-side component of the `meteor shell` command.
standard-minifier-css  1.3.5  Standard css minifier used with Meteor apps by default.
standard-minifier-js   1.2.3* Standard javascript minifiers used with Meteor apps by default.
tracker                1.1.3  Dependency tracker to allow reactive callbacks
```

- so in order to remove this package lets go to the terminal and type...

```
joses-MacBook-Pro:short-lnk mendoza$ meteor remove autopublish
```

- this will break the app until we add publications and subscriptions to fix it
- now lets go to links.js
- first import Meteor
- and then we are going to be using the publish() method from Meteor

```
import { Mongo } from 'meteor/mongo';

import { Meteor } from 'meteor/meteor';


export const Links = new Mongo.Collection('links');


Meteor.publish()
```

- the Meteor.publish() method is only available on the server
- lets explore some Meteor booleans
- like Meteor.isServer and Meteor.isClient
- lets check them out over in our console by typing

```
require('meteor/meteor').Meteor;

{isProduction: false, isDevelopment: true, isClient: true, isServer: false, isCordova: false, …}
```

- and right here the ones that we are looking for are the ones that start with ***is***

```
flush

:

ƒ (options)

isAppTest

:

false

isClient

:

true

isCordova

:

false

isDevelopment

:

true

isPackageTest
```

```
:
false
isProduction
:
false
isServer
:
false
isTest
:
false
```

- go back to links.js and lets do conditional statement

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

export const Links = new Mongo.Collection('links');

if (Meteor.isServer){
    Meteor.publish()
}
```

- **now lets take a look what arguments publish requires**
- **it requires 2**
  - **A string name like 'links' for this app (links here does not refer to the links collection, you name it what ever you like)**
  - **and a function**
    - **this function determines what data a specific client should have access to**
    -

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

export const Links = new Mongo.Collection('links');

if (Meteor.isServer){
    Meteor.publish('links', ()=>{
    })
}

```

- as a test we are going to first enable that first behaviour we had, where everyone has access to all data
- and we do that by...

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

export const Links = new Mongo.Collection('links');

if (Meteor.isServer){
    Meteor.publish('links', ()=>{
        return Links.find()
    })
}

```

- we next need to figure out how to subscribe to that publication, in order to do that lets go to LinksList.js
- we are going to be using another Meteor method first lets import it

```
import React from 'react';
import { Meteor } from 'meteor/meteor';
import { Tracker } from 'meteor/tracker';
import { Links } from '../api/links'

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();
            const links = Links.find().fetch()
            this.setState({links});
          });
    }
```

- subscribe only takes one argument, and it's going to be the exact same name we passed in to publish which in our case is links

```
import React from 'react';
import { Meteor } from 'meteor/meteor';
import { Tracker } from 'meteor/tracker';
import { Links } from '../api/links'


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().fetch()
           this.setState({links});
         });
    }
```

- now should be able to see all of the links from different accounts
- now as an exercise, we created 1, 2,3,and 4 urls, if we want to just show 1 we would do the following
- go links.js and type

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

```
export const Links = new Mongo.Collection('links');


if (Meteor.isServer){
    Meteor.publish('links', ()=>{
        return Links.find({url:'1'});
    })
}
```

- this will only show you urls with the value of string 1
-