

# 14-short-link-app - video

## Advanced Publications

- we have to associate the links the user created
- first we are going to modify the Link.js component
- currently the Link component call Links.insert
- but we also want the id, this is by no mean secure but for now we are going to do it like this

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

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

export default class Link extends React.Component{
  onLogout(){
    Accounts.logout();
  }
  onSubmit(e){
    const url = this.refs.url.value.trim();
    e.preventDefault();

    if(url){
      Links.insert({url, userId: Meteor.userId()});
      this.refs.url.value = '';
    }
  }
  render(){
    return (
      <div>
        <h1>Your Links</h1>
        <button onClick={this.onLogout.bind(this)}>Logout</button>
        <LinksList/>
        <p>Add Link</p>
        <form onSubmit={this.onSubmit.bind(this)}>
```

```

        <input type="text" ref="url" placeholder="URL"/>

        <button>Add Link</button>

    </form>

</div>

);

}

}

```

now are using Meteor in Link.js so lets import it up above

```

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

export default class Link extends React.Component{
  onLogout(){
    Accounts.logout();
  }
}

```

- lets check if this works, we ant to know if miniMongo does show userId
- go to the browser and enter 2 so it shows up
- and as we can they have the userId on miniMongo

```

_id: "WjuFR4GNbaEFsbJod"
url: "2"
1: 4er7Bx8vsYgZ8zrSB {} 3 keys
_id: "4er7Bx8vsYgZ8zrSB"
url: "2"
userId: "dWcqFGiaur6gwfpej"
2: 28RhX2iQ2rbLWqGy5 {} 3 keys
_id: "28RhX2iQ2rbLWqGy5"
url: "2"
userId: "dWcqFGiaur6gwfpej"
3: gMan9BXh7gy8yA6Q9 {} 3 keys
url: "2"

```

```
userId: "dWcqFGiaur6gwfpej"
```

```
_id: "gMan9BXh7gy8yA6Q9"
```

- now in order to fetch the userId over in the links.js we are going to need to switch from es6 to es5 function
- we need to do this switch because we need access to the **this** binding and that is something is not available inside of arrow functions
- the way Meteor allows us to access the userId in a publish function is this.userId
- but we first are going to remove all the links
- head over to the terminal and start mongo and remove the links

```
meteor:PRIMARY> db.links.remove({})
```

- now head on over to links.js and fix the issue by

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

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

if (Meteor.isServer){
  Meteor.publish('links', function(){
    return Links.find({userId : this.userId});
  })
}
```

- and once you log there will be no collections on miniMongo
- but as soon as you login you will see all of your links
-