

# Hello!

Thomas Roch



# Routing

Past and future of client-side routing

*We've only seen one player. Is there any innovation going on? (...) Innovation already happened a few years before. And it happened in the Ember community*

Christopher Chedeaux, React Europe keynote 2015

Is routing a solved problem?

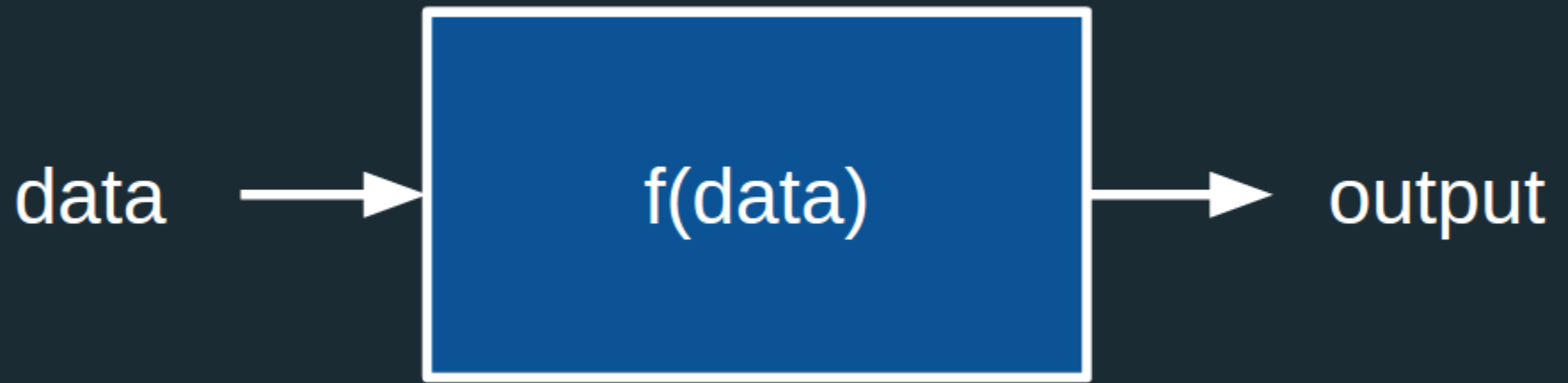
**LET ME THINK, HMM**

**NO, THAT'S NOT RIGHT**

memes.com

Today is all about components







Components are data-driven (state)

```
function Button({ children, onClick, disabled }) {  
  return (  
    <button type='button' onClick={ onClick } disabled=  
      { children }  
    </button>  
  );  
}
```

```
function LoggedUser({ user }) {  
  return user  
    ? <div>Hello { user.name }!</div>  
    : <a href='/login'>Log in</a>  
}
```

We compose components together  
in trees of components

But when it comes to routing...

```
function App({ route }) {  
  if (route === 'dashboard') {  
    return <Dashboard />;  
  }  
  
  if (route === 'profile') {  
    return <Profile />;  
  }  
}
```

Why do we treat routing differently?

We have been influenced by  
server-side routing



Django

```
router.register(r'users', UserViewSet)
```

Laravel

```
Route::get('/users', function () {  
    /* ... */  
});
```

Ruby on Rails

```
get '/users', to: 'users#list'
```

Play

```
GET /users controllers.Users.list()
```

Express

```
app.get('/users', function(req, res) {  
    res.send('Users list');  
});
```

request → response

route → action

# Stateless routing

We transposed the same mental model  
to stateful browsers

Angular 1.x

```
$stateProvider.state({  
  name: 'users',  
  url: '/users',  
  controller: UsersController,  
  templateUrl: '/users/list.html'  
});
```

React-router

```
<Route path="users" component={ Users }>
```

Page.js

```
page('/users', function() {  
  /* MUTATE THE DOM HERE */  
})
```

Routers became responsible for  
rendering components...

...transitioning between them...

...and what about code splitting?



Routers are made more complex by being tied to a component library or framework

Routing has been mystified

What about view / state separation?



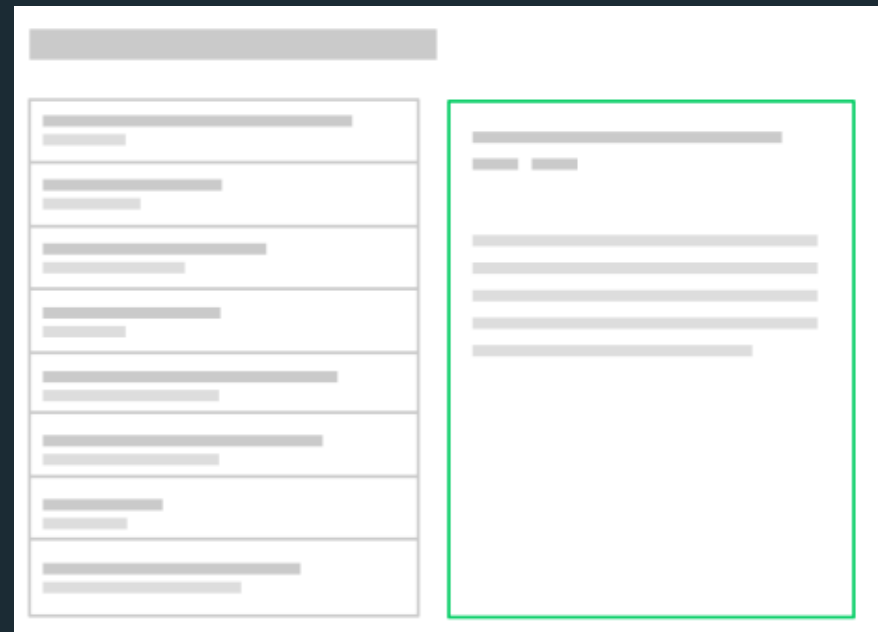
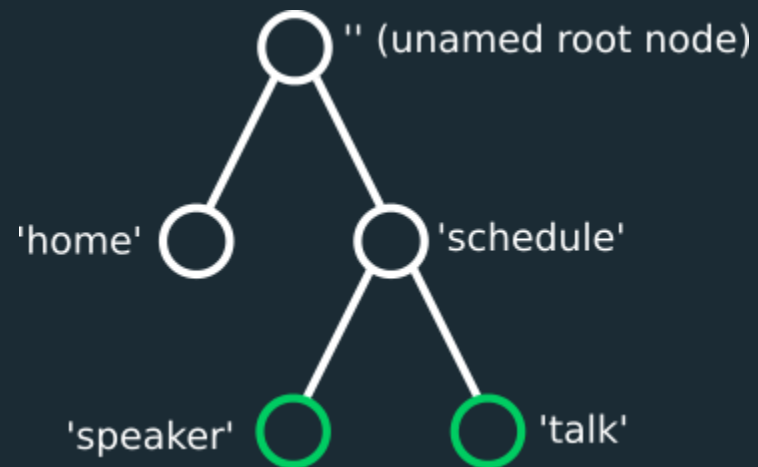
router5

[github.com/router5/router5](https://github.com/router5/router5)

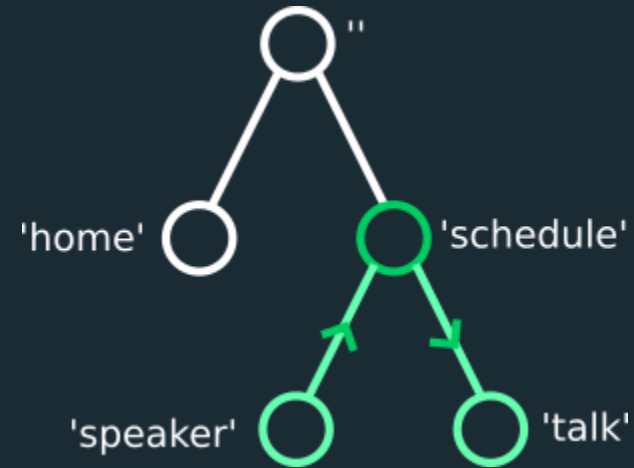
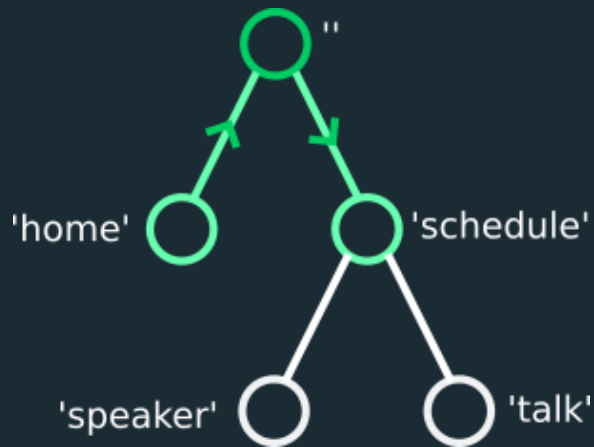
## router5 has advanced features

- Nested routes
- canActivate / canDeactivate
- Middleware functions
- Plugins

# Linking nodes to your UI



# Detecting the changing node



```
import React from 'react'
import { routeNode } from 'react-router5'

router.addNodeListener('', (state) => {
  /* ... */
});

const AppMainRouteNode = routeNode('')(AppMain)
```

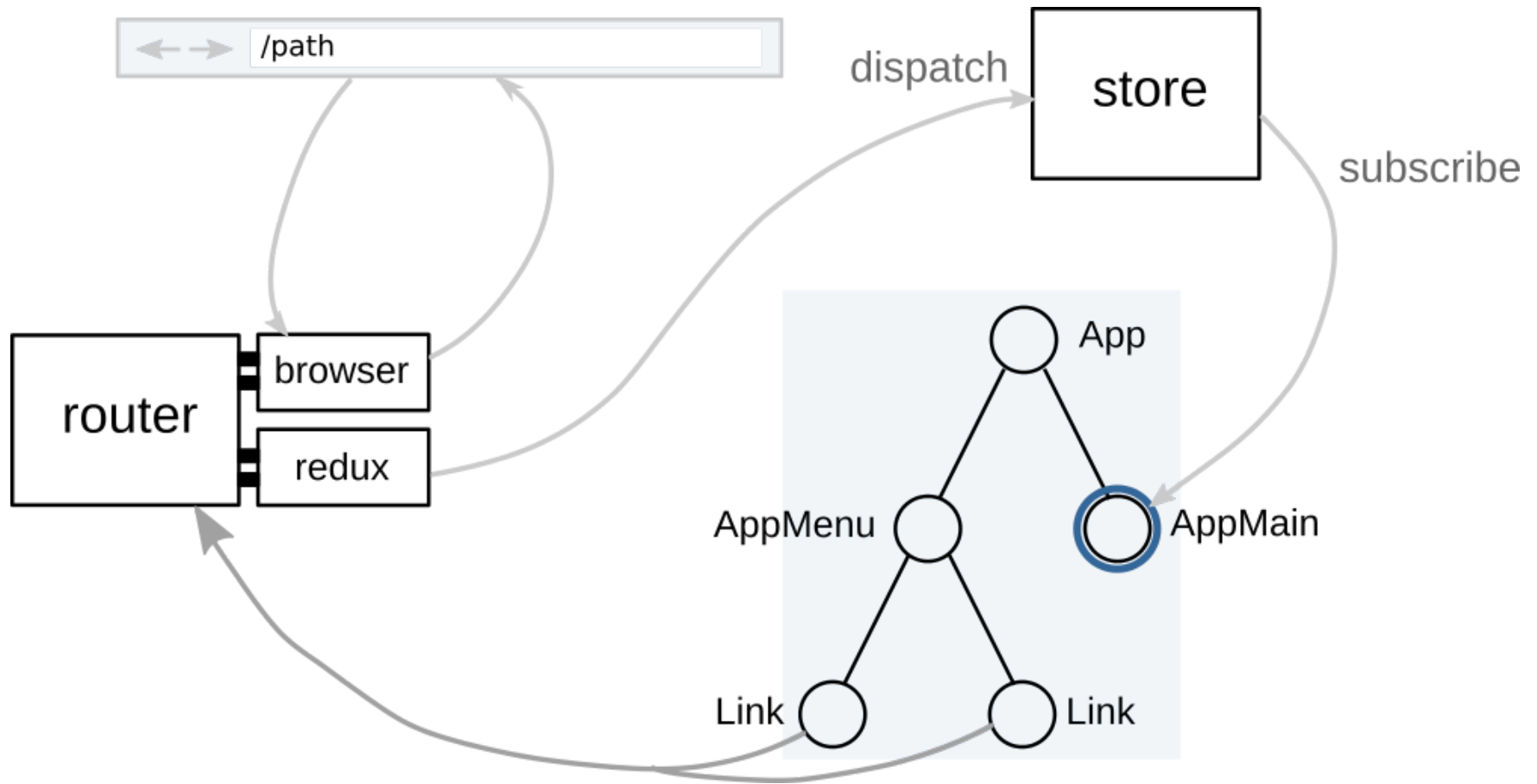


# Middleware functions

- Delay a transition
- Prevent a transition
- Hook

Integration becomes straight forward

# Integration with redux



# Observables

```
import createRouter from 'router5'
import createObservables from 'rxjs-router5'
import routes from './routers'

const router = createRouter(routes);
const {
  route$,
  routeNode,
  transitionError$,
  transitionRoute$
} = createObservables(router)
```

## Routing state / view separation

- Easier to integrate
- Leveraging existing component layers
- Code splitting
- Simpler universal architecture
- Native?



# FanDuel



Offices in Edinburgh and Glasgow  
UK product: react, redux and router5

A screenshot of the FanDuel website's football team selection interface. The top navigation bar is blue with white text for 'FANDUEL' and links for 'Contests', 'Upcoming', 'Live', 'Settled', 'How to play', 'Help', and 'Log in'. Below the navigation bar, a header section displays '£5,000 Fan Favourite (£750 to top)', '1047 of 1176 ENTRIES LEFT', '£5000 TOTAL PRIZES', and '£5 PER ENTRY'. The main content area is divided into two sections. On the left, a list of players is shown with columns for GK, DEF, MID, and FWD. Each player entry includes a team logo, player name, match details, and a price. On the right, a football pitch diagram shows the selected team formation, with players placed in their respective positions. The pitch is green with white lines, and the players are represented by their team's kit colors. The formation shown is a 4-2-3-1.

[www.fanduel.com/careers](http://www.fanduel.com/careers)

# Thank you!

Github @troch  
Twitter @tcroch