Routing between diff parts of an app

Routing

React Router is the most popular solution

Inspired by Ember’s router solution

React is a small, focused view engine and has no concept of routing

Plain React may be ok for a small app

As apps get bigger, it becomes desirable to link between aspects of your app using a maintainable solution

React Router

Simple nested approach

Declarative configure routes

Routes can be nested

Very similar to Ember’s approach

Supports handling complex url’s and query strings

Supports centralized management of links

Avoids a complete postback when routing

## Params and Query Strings

React router automatically maps url / query string components into the handlers for the given route

<route path=”/course/:courseId” handler={Course}/>

/course/clean-code?module=3

var Course = React.createClass({

render: function() {

this.props.params.courseId; //”clean-code”

this.props.query.module; //”3”

this.props.path; // “/course/clean-code/?module=3”

}

});

## Links

Abstraction over links

Leverages routes so don’t need to repeat whenever you want to link to a certain page

Normalizes references

Placeholder

In following route, :courseId is a placeholder

<route path=”/course/:courseId” handler={Course}/>

To reference a route in a link, give the route a name attribute

<route name=”foo” path=”/course/:courseId” handler={Course}/>

<Link to=”user” params={{courseId: 1}}>Test Class</Link>

Can also be used for displaying images

Makes maintaining large apps much easier.

## API

It’s only in 1.0, so will change often.

Route

RouteHandler

## Configuration

Route

Routes defined using the Route component

Typically, one route per page in app

If route isn’t given a path, React assumes that the name is also the path.

<Route name=”authors”...../>

/authors will load this route.

<Route name=”authors” path=”/foo”...../>

/foo will load this route now

DefaultRoute

what is displayed when user first accesses your app

Router.DefaultRoute

NotFoundRoute

404

Router.NotFoundRoute

Redirect

Redirect to another route

Useful to avoid sending someone to a 404, or to change the url in user’s browser

Lets you keep old url’s working

Can use asterisks / patterns, query strings / parms.

Handle common typos

## Transitions

Happens entirely on client side

Control when a user can transition to another route, or from current route.

## Locations

Clean url’s using router configuration

No hashes

Location styles

Hash

Html5 history style

Need to configure server to support client-side clean url’s

Direct all requests for app through index page

Client-side routing

Two approaches

Hash location and HTML5 history via push, replace and pop state

Hash location

Bit of a hack as it has hash signs in the url

Works in all browsers

No support for server-side rendering

HTML5

Doesn’t work in all devices, but 90% does

Between components and pages

willTransitionFrom

Run checks before user navigates away

Static function

Verify there’s no unsaved data before user navigates away

willTransitionTo

Determine if page should be transitioned to

Static function

Prevent transitioning to a page if not authenticated

var Settings = React.createClass({

statics: {

willTransitionTo: function(transition, params, query, callback) {

If (!isLoggedIn) {

transition.abort();

callback();

}

},

willTransitionFrom: function(transition, component) {

If (component.formHasUnsavedData()) {

If (!confirm(‘Sure you want to leave?’)) {  
 transition.abort();

}

}

}

}

});

Global variables

You can use eslint mechanisms to avoid “use strict” where needed without having eslint generate errors

/\*eslint-disable strict \*/ //Disable checking because we can’t run strict mode. Need global vars

Disable for entire page

Disable for just a variable

…

## Mixins

Run code in a number of different components

Configure mixins property with name of components.

Navigation mixin

Supports:

Programmatic navigation

Defined as an array within a React class

React.createClass({

Mixins: [

]

this.transitionTo(‘foo’)

this.replaceWith(‘contact’)

this.goBack

makePath(routeName, params, query)

## Terms

Deep linking

Routing

How requests (url’s) get routed to different parts of a system

HTML5 push state

HTML5 history