**React Router:**

See resources

**D01Basic**

1. Create a wrapper component( BrowserRouter o HashRouter) around the root Component
2. Create Link objects to point to urls (relative to root /)
3. Create Route components inside a Switch to map (priority from upper to bottom) all possibile urls

**TC: render new pages**1.1)Path in browser localhost:3000/products   
**1.**2) Usage of Links (click replace urls in browser)   
1.3) verify url pattern matching:

products/edit  
products/create  
products/whatever -> Products

**D02Nested routing Same Component (route.js):**

**Problem (1.3)**

Usually our links looks something like {entity/action}:

products/create  
products/edit

**Solution (nested path):**

Define some fully matching url on top of Switch component.   
product/create will be matched (found) before products (default for all missing paths)

TC:  
2.1) Verify from broswer url  
2.2) Verify throuogh links  
2.3) Verify Default matching

**D03Parameters in querystring (es product/edit/:id)**

**Problem:**Sometimes we need to pass in url path some parameters.  
Example (product/details/455)

**Solution:**  
In link: path/:paramName  
In destination Component **props**.**match.params.paramName**

**Details:**

Match object:  
<https://reacttraining.com/react-router/web/api/match>  
  
  
TC:  
3.1) Verify the props.match object in debugging time.

**D04Nested routing in route components (categories.jsx, products.jsx):**

One of the new feature in React Router 5 is the possibility to nest paths in wrapper components.  
For example delegate to a Product components management of all paths starting with /product.

**Implementation:**

**D05 Trigger Routing by code: (transition with other than Link as trigger)**

If this -> route DestinationPage1, else route to DestinationPage2  
 Table with dynamic cell button linked to row id.

**Solutions:**

1. <Redirect> declarative  
    (state -> evaluate -> render appropriate Redirect)
2. Imperative: (programmatic)  
   hashHistory.push('/contacts/details/'+id);

<https://tylermcginnis.com/react-router-programmatically-navigate/>

**Details:**

History:  
<https://github.com/ReactTraining/history/tree/master/docs>

**D06CustomRoute: (Tutorial for login)**Handling Redirection

and Authentication through PrivateRoute

**D07 Preventing transition:**

**D08 Router.render function**

**D09 Router config in separate file**

**D10 React Router with Redux**

**D11 React Router Authentication Permission/Roles**

Altre demo:

<https://reacttraining.com/react-router/web/example/query-parameters>

<https://reacttraining.com/react-router/web/example/auth-workflow>

<https://github.com/ReactTraining/react-router/blob/master/packages/react-router/docs/api/Redirect.md>

To break in:

1. Redirect (authenticated)