



# React: React Router

## React – react-router

---

The following exercise contains the following subjects:

- map
- Ajax
- Components

### Instructions

We are going to create a small part of an e-commerce website. We will provide you with some product data and you need to display them on the screen correctly using React Router.

Click the [link for a demo](#)

This exercise is divided into 4 parts.

### Part 1

Before displaying our products on the screen let's create our components and wire up our React Router.

### Create the components

1. Homepage component
2. Products component
3. Header component

## **React Router**

Homepage component path is “/”

Products component path is “/products”

Header component should always be visible

## **Part 2**

Let's create and display our data on the screen

## **Our data**

Create a store.js file inside your src folder.

Store this data (next page) in a variable and export default it.

```
[
  {
    title: "Stylish hat",
    imageUrl: "https://i.ibb.co/pPSQYfX/hat.jpg",
    price: 25,
    size: "Fits for all",
    id: 1,
  },
  {
    title: "Beautiful Jacket",
    imageUrl: "https://i.ibb.co/kq23hr4/jacket.jpg",
    price: 55,
    size: "Medium",
    id: 2
  },
  {
    title: "Fashionable Jeans",
    imageUrl: "https://i.ibb.co/C62CZRy/jean.jpg",
    price: 39,
    size: "Medium",
    id: 3
  },
  {
    title: "Smart tie",
    imageUrl: "https://i.ibb.co/z8GM1Mh/tie.jpg",
    price: 10,
    size: "Fits for all",
    id: 4
  },
];
```

## Store it to State

Import this file to your Products component.

When the component mounts to the screen, store it in the components state.

## Display the products

Display only each product title on the screen

## Part 4

Let's finish up with some final touches

### ProductDetail back button

Create a back button in our ProductDetail component that will direct back to the products page with the help of the Link tag.

### Header links

Create the following links

1. Link to the homepage
2. Link to the products page

### Extra:

#### **Make your routes inside your Link tag future-proof.**

If you ever want to change a route in the future, you would need to change all your `<Link>` tag routes as well. Find out a way that you don't need to hard code the route in your `<Link>` tags.

React router has a feature that gives you the current path which you can embed in your Link tag that will ensure whenever you change the route, the `<Link>` tag route will change as well.

#### **Create a 404 component.**

If a user goes to a route that does not exist, we should display to them a 404 page.

### **Pass state with React Routers Link Tag**

When creating a link to each product in the Products component, there is a way that you can pass the state to the Link tag and access that state in the ProductDetail component via props.

This could be useful in some use cases. We could pass to the Link tag's state the specific product and access it via props. So we don't need to import our store data and look for a specific product.

But there is a serious flaw in this approach. Do you know what it is?