**React Router 6 – Full Course**

This course will teach you to build real-world apps with React Router 6. Click here to get to the interactive version 👉 [https://scrimba.com/links/react-route...](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbUR6MUJETjh2N2R1TTZ4WkZockJEZHdtRU9nd3xBQ3Jtc0tuNHk5djF3S3hIWV85bFRiWVA0YUZMeGNUbDdKbTNiTFMwMkpiNmFhNVJYdmh0eUJIY01pY2Q2WVllODhqTkNxUzNvTi03V3ptVFg0TGMyd2VXV2FpNjhEVnltemZSLUEtc1lXal90cVdjY0dlYmFxTQ&q=https%3A%2F%2Fscrimba.com%2Flinks%2Freact-router-6-course&v=nDGA3km5He4) Throughout the course, you’ll be building an app called “VanLife” – an Airbnb-style web app dedicated to renting out travel vans for your next big road trip! As you build “VanLife”, you will learn all the important parts of React Router, such as layout and index routes, nested routes, filtering results with search parameters, protecting routes for authenticated users, and more. You will also learn about the new Remix-inspired data router APIs, including Loaders and Actions. This course was created by Bob Ziroll, Scrimba’s Head of Education. 🔗 Bob on Twitter here: [https://www.gstatic.com/youtube/img/watch/social_media/twitter_1x_v2.png / bobziroll](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbFJWUl9XdFVYQ2pNWk9TLThmQjlzdk8zRE4zd3xBQ3Jtc0tuSEhzUVhnWnhUakVVNmU0bDNVeVVlMGpJMWtfVkNkSXFwUXdFYlRxLUJCZzhqSnhpLU5ZQzVjU1lRenp5eFlZTVcwc2Q5endiNHQzdWRqczZkbnhjdDY0SE5MR1pqRHVoaWdDT1BXZW1pYTM5M1pScw&q=https%3A%2F%2Ftwitter.com%2Fbobziroll&v=nDGA3km5He4) 🔗 Scrimba on YouTube: [https://www.gstatic.com/youtube/img/watch/yt_favicon.png / scrimba](https://www.youtube.com/c/Scrimba) ⭐️ Get the code ⭐️ 🔗 Scrimba course: [https://scrimba.com/links/react-route...](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbHJhVl9pemN6QkxmUmpkeGVnMEF6NV9OWS1IQXxBQ3Jtc0tsSnF1UVJ1VGxRazl3ZzljdGJvbHpzbU13VG0ycDVDS0Y2VEpSYUdCOUZkRll1Tmg0TjRPMEQ1Y1RaTGlSOTRuSkp6M1NSb1ltY2FrZmlmejdwWFVyNmQ1UUVtcXgza3lkSkJXM19sa0dMTGI3ZU1wRQ&q=https%3A%2F%2Fscrimba.com%2Flinks%2Freact-router-6-course&v=nDGA3km5He4) 🔗 GitHub repo: [https://scrimba.com/links/react-route...](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbno4LU1iLWFzNTFoTTF2TTkteVBfWHFLc1dIUXxBQ3Jtc0trZUllUmJfMHBxN1ItTzRyM1NFVG9vbmoxdG92bWZmd21nakY1RlViQlpPVFJRcDd4czhUNXZHUnphUVI3aWFSRWZQcXBPSllwdmpLaHk4OGJ0ZXpKeUhqR3FtZTBjVHByQkxmeld4Vm1xcWhwR0UtMA&q=https%3A%2F%2Fscrimba.com%2Flinks%2Freact-router-course-github-repo&v=nDGA3km5He4) 💫 Links mentioned in course: 🔗 Scrimba’s Learn React Course - [https://scrimba.com/learn/learnreact](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqblZoeWNBNFdod0NtS2dtaVpXZU5SYlBOR2V2QXxBQ3Jtc0ttanBsWndkYXYxd1lXel9KdjZyLW9yakVKQm9KYmxidk5fTFBtM1o4RUdGWi1iMmlGMUFJU25DVE80TThRWmdpSzN6MV9yX2dzMnlzcEdJSEwwRzU4b0g0d2ZKZ3VWbEdKSFI5eVc4Tm1GVHVFdVJhWQ&q=https%3A%2F%2Fscrimba.com%2Flearn%2Flearnreact&v=nDGA3km5He4) 🔗 VanLife Figma Design - [https://scrimba.com/links/figma-vanlife](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqa1U2UWFnMEI2SlRISkJCRGFaaWhDWmNMeWdTZ3xBQ3Jtc0traENPR2t3eWlsZWpZQlhJWjFoSUtUWXdpbEN6dXVSdzc3X21SaXpBeWR5cGJ5VmFlbi1UOXpkYnV5RFhmSUdtbERvbG5sU0p6Y1dITE52MnJjV3ZxcVlTWF9VMS0wMnZTRlRHRXBpLUVPZDVaTGlQOA&q=https%3A%2F%2Fscrimba.com%2Flinks%2Ffigma-vanlife&v=nDGA3km5He4) 🔗 Firebase - [https://scrimba.com/links/firebase-ho...](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqa3hzelV3S2lpcVU0SDJ6MmtPNnhQQUNoLTVPd3xBQ3Jtc0tsZ002UDM5Y0F2aHBqdUk5MDFWSFktRTVOMXFlNE82Ml8zT1ZjZUJKQmVmeTlfTTZDV2JMeC1tRmhERlh2VnZYbGFDbnFSQmRyV252aDl3Z095RXpLVl9GdE43dlVWYWdpdWRiSjRKaFZNSG5BZUNKaw&q=https%3A%2F%2Fscrimba.com%2Flinks%2Ffirebase-homepage&v=nDGA3km5He4) 🔗 Firestore Docs, get all docs in collection - [https://scrimba.com/links/firestore-d...](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbmNiN2hRSHpYNUNRV0sybkt2Yy1uTi1TNHo4UXxBQ3Jtc0tscnV3NDhXcTZ4am8xN2hnN0dkaFEyY2pOdHpGZWs1TERLRnJ5bG1sNTdoaEZpSGtHTEVoWUVBLUxvYmJrMVZORGdtV205MjhtbmJCM3gyQWxDZlJZYVI4eFNHSkV4OGRxRGZ4alJEWjdtVTVxZmNHTQ&q=https%3A%2F%2Fscrimba.com%2Flinks%2Ffirestore-docs-get-all-docs-in-collectionfirestore-docs-get-all-docs-in-collection&v=nDGA3km5He4)) 🔗 Netlify - [https://scrimba.com/links/netlify-hom...](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbDA0VXpHZEZrV29rdUdPZ3ZrTDZ0OVVmMFZxUXxBQ3Jtc0ttYm1tRDBydGtKd3d5VkZCREE2Y2U5WVR1QS1Dbm91S01ZaWtHMEpGSHlTVy1LVThKWVZNNmlxMDI3WWJSeUVqRThqUV9IMWYtR013Q3NBOC16YWx3X001VHdoNWRsTFZlNGVrT3lzQUhMemtPb2FxUQ&q=https%3A%2F%2Fscrimba.com%2Flinks%2Fnetlify-home-page&v=nDGA3km5He4) 🔗 GitHub Desktop - [https://desktop.github.com/](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbmdWVzNMaTJoNmVkak5lZTV5bTJjdmJidnd6UXxBQ3Jtc0trNHk3aXRwZW4xeHJKMk1PRUE3bV9ESHVWRzJDVmg4RjNhZnoteUpkWndDRTRzSEl4UXNWT19VcHBoVlJlT1BNSEJoQVp5RWFxRWh1Y3Y3ZXFPN3k4b3VDbkhzT25IY1NqR1NLNmZmTXRMQVZBdGNXNA&q=https%3A%2F%2Fdesktop.github.com%2F&v=nDGA3km5He4) 🔗 Mirage JS - [https://miragejs.com/](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqblZSMVdiSTItSHhWTXhudDMxQ0lWUk5RcDZrUXxBQ3Jtc0trajQyXzlYT3NwS0pEM0RpU3JDZWVwc01jdEkxWkRCUFo0YzRjQWstOTliUm5hMnJWSVZBb3RxcmNDTFlsMHg4c3JtclR2a2cwbC1KSzc3SkVNZ1k2RUVTbTRZWVFrTHYyLVR6dkdOQ2V5VFQ3d015QQ&q=https%3A%2F%2Fmiragejs.com%2F&v=nDGA3km5He4)

# 01 - Introduction to React Router

# npm install react-icons --save

[0:00](https://www.youtube.com/watch?v=nDGA3km5He4&t=0s) 1: Introduction to React Router 6

[4:56](https://www.youtube.com/watch?v=nDGA3km5He4&t=296s) 2: Multi-page vs single-page apps

[10:12](https://www.youtube.com/watch?v=nDGA3km5He4&t=612s) Extra: Local Development & GitHub Repo

[12:25](https://www.youtube.com/watch?v=nDGA3km5He4&t=745s) 3: React Router Setup & BrowserRouter

import {BrowserRouter } from 'react-router-dom';

<BrowserRouter>

</BrowserRouter>

[15:40](https://www.youtube.com/watch?v=nDGA3km5He4&t=940s) 4: Routes

import {BrowserRouter,Routes,Route } from 'react-router-dom';

[18:23](https://www.youtube.com/watch?v=nDGA3km5He4&t=1103s) 5: BrowserRouter & Routes Challenge

[19:32](https://www.youtube.com/watch?v=nDGA3km5He4&t=1172s) 6: Route, Path, & Element

<Routes>

<Route path='/about' element={<ComponentToRender />}/>

</Routes>

[23:48](https://www.youtube.com/watch?v=nDGA3km5He4&t=1428s) 7: Quick Re-org

[24:53](https://www.youtube.com/watch?v=nDGA3km5He4&t=1493s) 8: Link

import { Link } from "react-router-dom";

<Link to="/about">reverse project</Link>

[28:55](https://www.youtube.com/watch?v=nDGA3km5He4&t=1735s) 9: VanLife project bootstrapping

[37:01](https://www.youtube.com/watch?v=nDGA3km5He4&t=2221s) 10: Initial Deploy to Netlify

[48:47](https://www.youtube.com/watch?v=nDGA3km5He4&t=2927s) 11: Mirage JS Server

[50:41](https://www.youtube.com/watch?v=nDGA3km5He4&t=3041s) 12: Challenge: Vans Page - Part 1

13: Challenge: Vans Page - Part 2

Fetching and storing data example

Rule Of Thumb Always Check The Data Structure

const [vans, setVans] = React.useState([])

    React.useEffect(() => {

        fetch("/api/vans")

            .then(res => res.json())

            .then(data => setVans(data.vans))

    }, [])

    const vanElements = vans.map(van => (

        <div key={van.id} className="van-tile">

            <Link to={`/two/vans/${van.id}`}>

                <img alt={van.name} src={van.imageUrl} />

                <div className="van-info">

                    <h3>{van.name}</h3>

                    <p>${van.price}<span>/day</span></p>

                </div>

                <i className={`van-type ${van.type} selected`}>{van.type}</i>

            </Link>

        </div>

    ))

  return (

        <div className="van-list-container">

            <h1>Explore our van options</h1>

            <div className="van-list">

                {vanElements}

            </div>

        </div>

    )

[1:02:09](https://www.youtube.com/watch?v=nDGA3km5He4&t=3729s) 14: Route Params

:id indicates a variable

<Route path='vans/:id' element={<SecondVansDetails />} />

<Link to={`/two/vans/${van.id}`}> </Link>

${van.id} same as :id

Using the id from the other page in the details page hence https://homepage/two/vans/2

import { useParams } from "react-router-dom"

use case

const params = useParams()

    const [van, setVan] = React.useState(null)

    React.useEffect(() => {

        fetch(`/api/vans/${params.id}`)

            .then(res => res.json())

            .then(data => setVan(data.vans))

    }, [params.id])

[1:25:09](https://www.youtube.com/watch?v=nDGA3km5He4&t=5109s) 19: Nested Routes Intro

[1:34:04](https://www.youtube.com/watch?v=nDGA3km5He4&t=5644s) 20: Fixing the Navbar with a Layout Route

[1:46:45](https://www.youtube.com/watch?v=nDGA3km5He4&t=6405s) 22: Bootstrap the Host pages

[1:50:45](https://www.youtube.com/watch?v=nDGA3km5He4&t=6645s) 23: Nesting the /host routes

[1:54:34](https://www.youtube.com/watch?v=nDGA3km5He4&t=6874s) 24: Creating the Host Layout

[2:01:04](https://www.youtube.com/watch?v=nDGA3km5He4&t=7264s) 25: Relative Paths

[2:05:32](https://www.youtube.com/watch?v=nDGA3km5He4&t=7532s) 26: Index Routes

[2:09:24](https://www.youtube.com/watch?v=nDGA3km5He4&t=7764s) 27: To nest or not to nest?

[2:14:34](https://www.youtube.com/watch?v=nDGA3km5He4&t=8074s) 28: Nested Routes Quiz

[2:19:26](https://www.youtube.com/watch?v=nDGA3km5He4&t=8366s) 29: Add Footer

[2:22:47](https://www.youtube.com/watch?v=nDGA3km5He4&t=8567s) 30: NavLink

[2:30:03](https://www.youtube.com/watch?v=nDGA3km5He4&t=9003s) 31: Active Link Styling with NavLink

[2:39:14](https://www.youtube.com/watch?v=nDGA3km5He4&t=9554s) 33: Adding Host Vans Routes

[2:44:28](https://www.youtube.com/watch?v=nDGA3km5He4&t=9868s) 34: Optimal Side Quest

[2:47:49](https://www.youtube.com/watch?v=nDGA3km5He4&t=10069s) 35: Building the Host Van Detail page

[2:56:47](https://www.youtube.com/watch?v=nDGA3km5He4&t=10607s) 36: Relative Links

[3:03:51](https://www.youtube.com/watch?v=nDGA3km5He4&t=11031s) 37: Back to all vans ==========

[3:09:08](https://www.youtube.com/watch?v=nDGA3km5He4&t=11348s) 38: Add /host/vans/:id Nested Routes

[3:17:17](https://www.youtube.com/watch?v=nDGA3km5He4&t=11837s) 39: Add the Final Navbar

[3:23:53](https://www.youtube.com/watch?v=nDGA3km5He4&t=12233s) 40: Outlet Context

[3:29:27](https://www.youtube.com/watch?v=nDGA3km5He4&t=12567s) 41: Update deployed version on Netlify

[3:32:45](https://www.youtube.com/watch?v=nDGA3km5He4&t=12765s) 42: Search Params Intro

[3:40:04](https://www.youtube.com/watch?v=nDGA3km5He4&t=13204s) 43: useSearchParams

[3:48:55](https://www.youtube.com/watch?v=nDGA3km5He4&t=13735s) 45: Filter the array w/ the search param

[3:55:47](https://www.youtube.com/watch?v=nDGA3km5He4&t=14147s) 47: Using Links to add search params

[4:01:57](https://www.youtube.com/watch?v=nDGA3km5He4&t=14517s) 49: Using the search params setter function

[4:08:05](https://www.youtube.com/watch?v=nDGA3km5He4&t=14885s) 51: Caveats to setting params

[4:09:38](https://www.youtube.com/watch?v=nDGA3km5He4&t=14978s) 52: Merging search params

[4:21:13](https://www.youtube.com/watch?v=nDGA3km5He4&t=15673s) 54: Challenge: Conditional rendering practice

[4:25:56](https://www.youtube.com/watch?v=nDGA3km5He4&t=15956s) 55: Fix remaining absolute paths

[4:27:50](https://www.youtube.com/watch?v=nDGA3km5He4&t=16070s) 56: Back to all vans

[4:30:05](https://www.youtube.com/watch?v=nDGA3km5He4&t=16205s) 57: Link state

[4:36:37](https://www.youtube.com/watch?v=nDGA3km5He4&t=16597s) 58: useLocation

[4:47:31](https://www.youtube.com/watch?v=nDGA3km5He4&t=17251s) 60: 404 Page

[4:53:22](https://www.youtube.com/watch?v=nDGA3km5He4&t=17602s) 61: Happy Path vs Sad Path

[4:56:01](https://www.youtube.com/watch?v=nDGA3km5He4&t=17761s) 62: Update to our fetching code

[4:59:02](https://www.youtube.com/watch?v=nDGA3km5He4&t=17942s) 63: Coding the Sad Path

[5:07:37](https://www.youtube.com/watch?v=nDGA3km5He4&t=18457s) 65: Loaders intro

[5:12:03](https://www.youtube.com/watch?v=nDGA3km5He4&t=18723s) 66: createBrowserRouter

[5:18:13](https://www.youtube.com/watch?v=nDGA3km5He4&t=19093s) 67: Setting up the data router

[5:21:00](https://www.youtube.com/watch?v=nDGA3km5He4&t=19260s) 68: Loader function

[5:25:17](https://www.youtube.com/watch?v=nDGA3km5He4&t=19517s) 70: useLoaderData

[5:29:52](https://www.youtube.com/watch?v=nDGA3km5He4&t=19792s) 72: Use the loader data instead of the useEffect

[5:33:51](https://www.youtube.com/watch?v=nDGA3km5He4&t=20031s) 73: Loaders Quiz

[5:37:06](https://www.youtube.com/watch?v=nDGA3km5He4&t=20226s) 74: Handling errors

[5:39:49](https://www.youtube.com/watch?v=nDGA3km5He4&t=20389s) 75: Add errorElement to vans route

[5:42:40](https://www.youtube.com/watch?v=nDGA3km5He4&t=20560s) 76: useRouteError

[5:49:06](https://www.youtube.com/watch?v=nDGA3km5He4&t=20946s) 77: Initial Login Form

[5:51:31](https://www.youtube.com/watch?v=nDGA3km5He4&t=21091s) 78: Importing image assets in Vite

[5:54:22](https://www.youtube.com/watch?v=nDGA3km5He4&t=21262s) 79: Protected Routes

[6:18:28](https://www.youtube.com/watch?v=nDGA3km5He4&t=22708s) 85: Parallel Loaders Demo

[6:22:02](https://www.youtube.com/watch?v=nDGA3km5He4&t=22922s) 86: Challenge - Protected Routes

[6:43:51](https://www.youtube.com/watch?v=nDGA3km5He4&t=24231s) 91: Send login message prompt to login page

[6:46:56](https://www.youtube.com/watch?v=nDGA3km5He4&t=24416s) 92: Consume message

[6:54:37](https://www.youtube.com/watch?v=nDGA3km5He4&t=24877s) 93: Pass message to Login page

[6:58:26](https://www.youtube.com/watch?v=nDGA3km5He4&t=25106s) 94: Hot Take: Forms in React are bad

[7:00:58](https://www.youtube.com/watch?v=nDGA3km5He4&t=25258s) 95: Setting up for auth

[7:13:57](https://www.youtube.com/watch?v=nDGA3km5He4&t=26037s) 97: useNavigate()

[7:17:44](https://www.youtube.com/watch?v=nDGA3km5He4&t=26264s) 98: React Router Form Component

[7:20:41](https://www.youtube.com/watch?v=nDGA3km5He4&t=26441s) 99: Setting up the action function

[7:25:13](https://www.youtube.com/watch?v=nDGA3km5He4&t=26713s) 100: Add form and action to VanLife

[7:27:31](https://www.youtube.com/watch?v=nDGA3km5He4&t=26851s) 101: Action function

[7:32:39](https://www.youtube.com/watch?v=nDGA3km5He4&t=27159s) 103: Get form data in VanLife

[7:34:22](https://www.youtube.com/watch?v=nDGA3km5He4&t=27262s) 104: Use data in action to log in

[7:36:57](https://www.youtube.com/watch?v=nDGA3km5He4&t=27417s) 105: Better (but still fake) auth

[7:44:08](https://www.youtube.com/watch?v=nDGA3km5He4&t=27848s) 107: Form replace

[7:49:23](https://www.youtube.com/watch?v=nDGA3km5He4&t=28163s) 108: useActionData

[7:53:50](https://www.youtube.com/watch?v=nDGA3km5He4&t=28430s) 109: Action error handling

[8:00:00](https://www.youtube.com/watch?v=nDGA3km5He4&t=28800s) 111: useNavigation()

[8:08:07](https://www.youtube.com/watch?v=nDGA3km5He4&t=29287s) 113: Get previous route pathname

[8:15:05](https://www.youtube.com/watch?v=nDGA3km5He4&t=29705s) 114: redirectTo

[8:30:37](https://www.youtube.com/watch?v=nDGA3km5He4&t=30637s) 117: Deferring data

[8:33:35](https://www.youtube.com/watch?v=nDGA3km5He4&t=30815s) 118: Promises and defer()

[8:39:19](https://www.youtube.com/watch?v=nDGA3km5He4&t=31159s) 119: defer getVans()

[8:41:04](https://www.youtube.com/watch?v=nDGA3km5He4&t=31264s) 120: Await component

[8:55:28](https://www.youtube.com/watch?v=nDGA3km5He4&t=32128s) 123: React Suspense

[9:00:27](https://www.youtube.com/watch?v=nDGA3km5He4&t=32427s) 124: Suspense in VanLife

[9:03:51](https://www.youtube.com/watch?v=nDGA3km5He4&t=32631s) 125: Putting it all together - Defer, Await, Suspense in HostVans

[9:08:31](https://www.youtube.com/watch?v=nDGA3km5He4&t=32911s) 126: errorElements in remaining van loading pages

[9:11:55](https://www.youtube.com/watch?v=nDGA3km5He4&t=33115s) 127: Placeholders are gone!

[9:13:57](https://www.youtube.com/watch?v=nDGA3km5He4&t=33237s) 128: Cloud Firestore

[9:23:25](https://www.youtube.com/watch?v=nDGA3km5He4&t=33805s) 130: Collection reference and getVans() function

[9:30:54](https://www.youtube.com/watch?v=nDGA3km5He4&t=34254s) 131: Create getVan() function

[9:35:42](https://www.youtube.com/watch?v=nDGA3km5He4&t=34542s) 132: Refactor getHostVans function

[9:39:29](https://www.youtube.com/watch?v=nDGA3km5He4&t=34769s) 133: Final loose ends