New New API references, TypeScript code snippets, and docs DX

Back to integrations Add your integration

React Router v6

Use React Router v6 in your stories

```
npm install storybook-addon-react-router-v6
```

README View on GitHub

Storybook Addon React Router v6

Storybook npm v1.0.2 Release passing downloads 237k/month

> Use React Router v6 in your stories.

Recent changes

✅ Added support for route `id`

✅ Support for Storybook 7 has been added.

Version `1.x` only support Storybook 7.
If you use Storybook 6,
`yarn add -D storybook-addon-react-router-v6@0.3.6`.
Features and fixes will continue to be backported for a while.

Getting Started

Install the package

```
yarn add -D storybook-addon-react-router-v6
```

Add it to your storybook configuration:

```
// .storybook/main.ts
module.exports = {
    addons: ["storybook-addon-react-router-v6"],
};
```

How to use it as a component decorator

To add the router to all the stories of a component, simply add it to the `decorators` array.

Note that the `parameters.reactRouter` property is optional, by default the router will render the component at `/`.

```
import { withRouter } from 'storybook-addon-react-router-v6';

export default {
  title: 'User Profile',
  component: UserProfile,
  decorators: [withRouter],
  parameters: {
    reactRouter: {
      routePath: '/users/:userId',
      routeParams: { userId: '42' },
    }
  }
};

export const Example = () => <UserProfile />;
```

Usage at the story level

If you want to change the router config just for one story you can do the following :

```
import { withRouter } from 'storybook-addon-react-router-v6';

export default {
  title: 'User Profile',
  component: UserProfile,
  decorators: [withRouter],
};

export const Example = () => <UserProfile />;
Example.story = {
  parameters: {
    reactRouter: {
      routePath: '/users/:userId',
      routeParams: { userId: '42' },
      routeHandle: "Profile",
      searchParams: { tab: 'activityLog' },
      routeState: { fromPage: 'homePage' },
    }
  }
};
```

Define a global default

If you want you can wrap all your stories inside a router by adding the decorator in your `preview.js` file.

```
// preview.js

export const decorators = [withRouter];

// you can also define global defaults parameters
export const parameters = {
  reactRouter: {
    // ...
  }
}
```

Data Router

If you use the data routers of `react-router 6.4+`, such as `<BrowserRouter />`, you can use the following properties :

```
export const Example = () => <Articles />;
Example.story = {
  parameters: {
    reactRouter: {
      routePath: '/articles',
      loader: fetchArticlesFunction,
      action: articlesActionFunction,
      errorElement: <FancyErrorComponent />,
    }
  }
};
```

Outlet

If your component renders an outlet, you can set the `outlet` property :

```
export const Example = () => <Articles />;
Example.story = {
    parameters: {
        reactRouter: {
            routePath: '/articles',
            outlet: {
                element: <Article />,
                handle: "Article",
                path: ':articleId',
                loader: yourLoaderFunction,
                action: yourActionFunction,
                errorElement: <FancyErrorComponent />,
            },
            // Or simply
            outlet: <MostRecentArticles />,
        }
    }
};
```

Descendant Routes

`<Route>` can be nested to handle layouts & outlets. But components can also render a `<Routes>` component with its set of `<Route>`, leading to a deep nesting called `Descendant Routes`. In this case, in order for the whole component tree to render in your story with matching params, you will need to set the `browserPath` property :

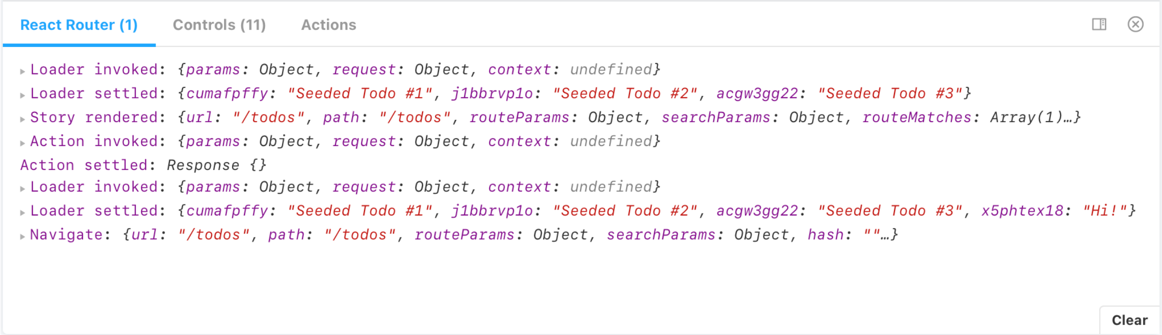
```
export default {
  title: 'Descendant Routes',
  component: SettingsPage, // this component renders a <Routes> witl
  decorators: [withRouter],
};

Default.story = {
  parameters: {
    reactRouter: {
      browserPath: '/billing',
    }
  }
};

// If you want to render at a specific path, like `/settings`, Reac
SpecificPath.story = {
  parameters: {
    reactRouter: {
      routePath: '/settings/*',
      browserPath: '/settings/billing',
    }
  }
}
```

Dedicated panel

Navigation events, loader and actions are logged, for you to better understand the lifecycle of your components.



```
▸ Loader invoked: {params: Object, request: Object, context: undefined}
▸ Loader settled: {cumafpffy: "Seeded Todo #1", j1bbrvp1o: "Seeded Todo #2", acgw3gg22: "Seeded Todo #3"}
▸ Story rendered: {url: "/todos", path: "/todos", routeParams: Object, searchParams: Object, routeMatches: Array(1)…}
▸ Action invoked: {params: Object, request: Object, context: undefined}
Action settled: Response {}
▸ Loader invoked: {params: Object, request: Object, context: undefined}
▸ Loader settled: {cumafpffy: "Seeded Todo #1", j1bbrvp1o: "Seeded Todo #2", acgw3gg22: "Seeded Todo #3", x5phtex18: "Hi!"}
▸ Navigate: {url: "/todos", path: "/todos", routeParams: Object, searchParams: Object, hash: ""…}
```

Available Parameters

Every parameter is optional. In most cases they follow the same type used by Route Router itself, sometimes they offer a sugar syntax.

Parameter	Type
routePath	`string`
routeParams	`Record<string, string>`
routeState	`any`
routeHandle	`any`
searchParams	`string[][] | Record<string, string> | string | URLSearch`
outlet	`React.ReactNode | OutletProps`
browserPath	`string`
loader	`LoaderFunction`
action	`ActionFunction`
errorElement	`React.ReactNode | null`
hydrationData	`HydrationState`
shouldRevalidate	`ShouldRevalidateFunction`
routeId	`string`

```
type OutletProps = {
  element: React.ReactNode;
  path?: string;
  handle?: unknown;
  loader?: LoaderFunction;
  action?: ActionFunction;
  errorElement?: React.ReactNode | null;
}
```

Compatibility

This package aims to support `Storybook > 6.4` and `React > 16`. Storybook versions prior `6.4` are very likely to work, I just didn't test them.

If you have an issue with any version, open an issue.

✅ Storybook 6.4
✅ Storybook 6.5
✅ Storybook 7.0

✅ React 16
✅ React 17
✅ React 18

Contribution

Contributions are welcome.

Before writing any code, file an issue to showcase the bug or the use case for the feature you want to see in this addon.

MADE BY

jesus_the_hun

WORKS WITH

React

TAGS

React-router React-router-v6 Router

How to install addons

Create an addon

JOIN THE COMMUNITY

Get news emailed to you

6,207 developers and counting

WHY

Why Storybook

Component-driven UI

DOCS

Guides

Tutorials

Changelog

Telemetry

Status

COMMUNITY

Integrations

Get involved

Blog

We're hiring

SHOWCASE

Explore

Projects

Component glossary

Open source software Maintained by

Special thanks to Netlify and CircleCI