**What is difference between route parameter and query parameter?**

**Search on ChatGPT.**

Route parameters and query parameters are both ways to pass information in a URL, but they have different purposes and are used in different ways.

**Route Parameters**

* **Definition**: Route parameters are part of the URL path and are defined in the route configuration of a web application. They are typically used to capture values from the URL that identify resources or entities.
* **Syntax**: Route parameters are specified using a colon (:) followed by the parameter name in the route definition.

javascript

Copy code

<Route path="/user/:id" element={<UserProfile />} />

* **Example URL**:

arduino

Copy code

http://example.com/user/123

Here, 123 is a route parameter for id.

* **Accessing**: In React Router, route parameters can be accessed using the useParams hook.

javascript

Copy code

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

function UserProfile() {

const { id } = useParams();

return <div>User ID: {id}</div>;

}

* **Use Case**: Route parameters are used for identifying specific resources, such as user profiles, product details, etc.

bash

Copy code

/product/:productId

/category/:categoryName

**Query Parameters**

* **Definition**: Query parameters are part of the URL that comes after the question mark (?). They are used to pass additional data to the server or client, often used for filtering, sorting, pagination, and other similar purposes.
* **Syntax**: Query parameters are specified as key-value pairs separated by & and come after the ? in the URL.

bash

Copy code

http://example.com/search?query=react&sort=asc

* **Example URL**:

bash

Copy code

http://example.com/search?query=react&sort=asc

Here, query and sort are query parameters.

* **Accessing**: Query parameters can be accessed using the URLSearchParams API or other similar methods.

javascript

Copy code

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

function SearchPage() {

const { search } = useLocation();

const queryParams = new URLSearchParams(search);

const query = queryParams.get('query');

const sort = queryParams.get('sort');

return (

<div>

Query: {query}, Sort: {sort}

</div>

);

}

* **Use Case**: Query parameters are often used for additional, optional data that affects the view or behavior of a page without changing the resource being accessed.

bash

Copy code

/search?query=react

/products?page=2&sort=price\_asc

**Key Differences**

1. **Placement in URL**:
   * **Route Parameters**: Part of the path.

bash

Copy code

/user/:id

* + **Query Parameters**: After the ? in the URL.

bash

Copy code

/search?query=react

1. **Usage**:
   * **Route Parameters**: Typically used to identify a specific resource or entity.

bash

Copy code

/user/123

/product/456

* + **Query Parameters**: Used to pass additional, optional information to the server or client, often for filtering, sorting, or pagination.

bash

Copy code

/search?query=react

/products?page=2&sort=price\_asc

1. **Accessing in Code**:
   * **Route Parameters**: Accessed using route-matching tools (like useParams in React Router).

javascript

Copy code

const { id } = useParams();

* + **Query Parameters**: Accessed using query string parsing tools (like URLSearchParams).

javascript

Copy code

const queryParams = new URLSearchParams(search);

const query = queryParams.get('query');

**Summary**

* **Route Parameters**: Used for essential parts of the URL that identify specific resources. They are part of the URL path and are defined in the route configuration.
* **Query Parameters**: Used for additional information that modifies the resource or affects the behavior of the page. They come after the ? in the URL and are passed as key-value pairs.

**Note: We don’t have to mention query params in routing file while we must have to mention route params in route URL like below example.**

{

        id: 1,

        path: "/page/:pageNumber",

        element: <HomeScreen />,

      },

      {

        id: 1,

        path: "/search/:keyword/page/:pageNumber",

        element: <HomeScreen />,

      },

      {

        id: 2,

        path: "/product/:id",

        element: <ProductScreen />,

      },

**Here id and pageNumber are route params variable mentioned in routing file.**