

1. Requests examples

for redirect or re navigate, use this,

```
import { useNavigate } from "react-router-dom";
```

using functions, kono page e redirect korar jonno function er vitor jevabe likhte hoy,

```
const submitForm = (e) => {  
  e.preventDefault();  
  return navigate("/jobs");  
};
```

Post request data format with async and await,

```
const App = () => {  
  const addJob = async (newJob) => {  
    const res = await fetch("/api/jobs", {  
      method: "POST",  
      headers: {  
        "content-type": "application/json",  
      },  
      body: JSON.stringify(newJob),  
    });  
  
    return;  
  };  
};
```

Delete Request,

```
const onDeleteData = async () => {
  console.log(id);

  try {
    const deleteData = await fetch(`http://127.0.0.1:8000/jobs/${id}`, {
      method: "DELETE",
    });

    if (!deleteData.ok) {
      throw new Error("Network response was not ok");
    }

    navigate("/jobs");
  } catch (error) {
    console.error("Failed to delete the job:", error);
  }
};
```

Getting all data using useEffects(),

```
useEffect(() => {
  const fetchJobs = async () => {
    try {
      const res = await fetch("/api/jobs");
      const data = await res.json();
      setJobs(data);
    } catch (error) {
      console.log("Error fetching data");
    } finally {
      setLoading(false);
    }
  };

  fetchJobs();
}, []);
```

Viewing specific data based on id with job loader, for more please check jobLoader docs,

```
const jobLoader = async ({ params }) => {
  const res = await fetch(`/api/jobs/${params.id}`);
  const data = await res.json();
  return data;
};

export { JobPage as default, jobLoader };
```

EditJob code snippet,

```
const submitForm = async (e) => {
  e.preventDefault();

  try {
    const updateJobObject = {
      title,
      type,
      location,
      description,
      salary,
      company: {
        name: companyName,
        description: companyDescription,
        contactEmail,
        contactPhone,
      },
    };

    const updateJob = await fetch(`/api/jobs/${id}`, {
      method: "PUT",
      headers: {
        "content-type": "application/json",
      },
      body: JSON.stringify(updateJobObject),
    });

    navigate("/jobs");
  } catch (error) {
    console.log(error);
  }
};
```