

Example: For a blog website. [Blog object keys: title, body, author]

1. Custom hook to get data: Gives data, loading state, error msg.

```
import { useState, useEffect } from 'react';

const useFetch = (url) => {
  const [data, setData] = useState(null);
  const [isPending, setIsPending] = useState(true);
  const [error, setError] = useState(null);

  useEffect(() => {
    const abortCont = new AbortController();

    setTimeout(() => {
      fetch(url, { signal: abortCont.signal })
        .then(res => {
          if (!res.ok) { // error coming back from server
            throw Error('could not fetch the data for that resource');
          }
          return res.json();
        })
        .then(data => {
          setIsPending(false);
          setData(data);
          setError(null);
        })
        .catch(err => {
          if (err.name === 'AbortError') {
            console.log('fetch aborted')
          } else {
            // auto catches network / connection error
            setIsPending(false);
            setError(err.message);
          }
        })
      }, 1000);

    // abort the fetch
    return () => abortCont.abort();
  }, [url])

  return { data, isPending, error };
}

export default useFetch;
```

2. Fetching all data objects

```
const {data: blogs, isPending, error} = useFetch('http://localhost:8000/blogs');
```

3. Fetching data of single object using id

```
const {data: blogs, isPending, error} = useFetch('http://localhost:8000/blogs/' + blog.id);
```

4. Adding an object

```
const handleSubmit = (e) => {
  e.preventDefault();
  const blog = { title, body, author };

  fetch('http://localhost:8000/blogs/', {
    method: 'POST',
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify(blog)
  }).then(() => {
    // history.go(-1);
    history.push('/');
  })
}
```

5. Deleting an object using id

```
fetch('http://localhost:8000/blogs/' + blog.id, {
  method: 'DELETE'
}).then(() => {
  history.push('/');
})
```

6. Usage example:

a) To fetch and display details of a single object using id & deleting that object on click.

```
const BlogDetails = () => {
  const { id } = useParams();
  const { data: blog, error, isPending } = useFetch('http://localhost:8000/blogs/' + id);
  const history = useHistory();

  const handleClick = () => {
    fetch('http://localhost:8000/blogs/' + blog.id, {
      method: 'DELETE'
    }).then(() => {
      history.push('/');
    })
  }

  return (
    <div className="blog-details">
      { isPending && <div>Loading...</div> }
      { error && <div>{ error }</div> }
      { blog && (
        <article>
          <h2>{ blog.title }</h2>

```

```

        <p>Written by { blog.author }</p>
        <div>{ blog.body }</div>
        <button onClick={handleClick}>delete</button>
    </article>
  )}
</div>
);
}

export default BlogDetails;

```

b) To fetch all objects & display them using map & Link that object to its details page on click.

//For fetching

```

const Home = () => {

  const {data: blogs, isPending, error} = useFetch('http://localhost:8000/blogs');

  return (
    <div className="home">
      { error && <div>{ error }</div> }
      { isPending && <div>Loading...</div> }
      { blogs && <BlogList blogs={blogs} /> }
    </div>
  );
}

export default Home;

```

//For displaying

```

const BlogList = (props) => {
  const blogs = props.blogs;
  return (
    <div className="blog-list">
      <h2>{props.title}</h2>
      {blogs.map(blog => (
        <div className="blog-preview" key={blog.id} >
          <Link to={` /blogs/${blog.id}`}>
            <h2>{ blog.title }</h2>
            <p>Written by { blog.author }</p>
          </Link>
        </div>
      ))}
    </div>
  );
}

export default BlogList;

```