

```
import React, { useEffect, useState } from "react";

import {
  ActivityIndicator,
  TextInput,
  FlatList,
  Text,
  View,
  Button,
} from "react-native";

const App = () => {
  const [isLoading, setLoading] = useState(true);
  const [data, setData] = useState([]);

  const [isAdded, setAdded] = useState(false);
  const [isRemoved, setRemoved] = useState(false);
  const [isUpdated, setUpdated] = useState(false);
  const [bkId, setBkId] = useState("");

  const [bk, setBk] = useState({
    name: "",
    category: "",
    author: "",
  });

  const onChangeName = (value) => {
    setBk({ ...bk, name: value });
  };
};
```

```
const onChangeCategory = (value) => {  
  setBk({ ...bk, category: value });  
};
```

```
const onChangeAuthor = (value) => {  
  setBk({ ...bk, author: value });  
};
```

```
const addBook = () => {  
  let newbook = {};  
  // let d = `name="Morning"&category="2"&author="Life"`;  
  let d = `name=${bk.name}&category=${bk.category}&author=${bk.author}`;
```

```
  fetch("http://10.0.0.5:44350/helloworldWebService1.asmx/AddBook", {  
    method: "POST",  
    headers: {  
      "Content-Type": "application/x-www-form-urlencoded",  
    },  
    body: d,  
  })
```

```
    .then((responseData) => {  
      setAdded(true);  
      getMovies();  
      clearFomrBkFields();  
      console.log("Done");  
    })  
    .catch((err) => {  
      console.log(err);
```

```
});  
};
```

```
const clearFomrBkFields = () => {  
  bk.name = "";  
  bk.category = "";  
  bk.author = "";  
};
```

```
const getMovies = async () => {  
  try {  
    const response = await fetch(  
      "http://10.0.0.5:44350/helloworldWebService1.asmx/GetBooks"  
    );  
    const json = await response.json();  
    setData(json);  
  } catch (error) {  
    console.error(error);  
  } finally {  
    setLoading(false);  
  }  
};
```

```
const deleteBook = () => {  
  console.log("bkId=" + bkId);  
  let book = {  
    id: bkId,  
    //id: 19,
```

```

};

// let d = `name="Morning"&category="2"&author="Life"`;

let d = `id=${book.id}`;

fetch("http://localhost:44350/helloworldWebService1.asmx/DeleteBook", {
  method: "POST",
  headers: {
    "Content-Type": "application/x-www-form-urlencoded",
  },
  body: d,
})
  .then((responseData) => {
    setRemoved(true);
    getMovies();
    console.log("Done");
  })
  .catch((err) => {
    console.log(err);
  });
};

```

```

const updateBook = () => {
  let book = {
    // id: "12",
    // name: "Night",
    // category: "3",
    // author: "Moon",

```

```
id: bkId,  
name: bk.name,  
category: bk.category,  
author: bk.author,  
};
```

```
let d =  
`id=${book.id}&name=${book.name}&category=${book.category}&author=${book.author}`;
```

```
fetch("http://localhost:44350/helloworldWebService1.asmx/UpdateBook", {  
  method: "POST",  
  headers: {  
    "Content-Type": "application/x-www-form-urlencoded",  
  },  
  body: d,  
})  
  .then((responseData) => {  
    setUpdated(true);  
    getMovies();  
    console.log("Done");  
  })  
  .catch((err) => {  
    console.log(err);  
  });  
};
```

```
useEffect(() => {  
  getMovies();
```

```
}, []);
```

```
return (
```

```
<View style={{ flex: 1, padding: 24 }}>
```

```
{isLoading ? (
```

```
<ActivityIndicator />
```

```
): (
```

```
<View>
```

```
<TextInput
```

```
placeholder={"Book name"}
```

```
onChangeText={(value) => onChangeName(value)}
```

```
value={bk.name}
```

```
/>
```

```
<TextInput
```

```
placeholder={"Category"}
```

```
onChangeText={(value) => onChangeCategory(value)}
```

```
value={bk.category}
```

```
/>
```

```
<TextInput
```

```
placeholder={"Author"}
```

```
onChangeText={(value) => onChangeAuthor(value)}
```

```
value={bk.author}
```

```
/>
```

```
<Text>{isLoading ? "Loading..." : ""}</Text>
```

```
<Button title="Add a book" onPress={addBook}></Button>
```

```

<TextInput
  placeholder={"Id to be updated or removed"}
  onChangeText={(value) => setBkId(value)}
  value={bkId}
/>

<Text>{isRemoved ? "Removed" : ""}</Text>

<Button title="Delete a book" onPress={deleteBook}></Button>

<Button title="Update a book" onPress={updateBook}></Button>

<Text>{isUpdated ? "Updated" : ""}</Text>

<FlatList
  data={data}
  keyExtractor={({ id }) => id}
  renderItem={({ item }) => (
    <Text>
      {item.Id}. {item.Name} by {item.Author}.
    </Text>
  )}
/>
</View>
)}
</View>
);
};

export default App;

```