

React frontend with your **Express.js backend** to display dynamic data — for example, sending a list of services, team members, or FAQs from the backend to show in your React app.

□ What We'll Do

1. **Create a simple Express.js backend** that serves JSON data.
2. **Fetch that data in your React frontend** using `fetch` or `axios`.
3. **Display the data** in a component (e.g. in services)

1. Backend – Express-js API

Assuming you're working in a folder like `express-backend/`:

Express js setup :-

1)open command prompt and type following command:-

As shown below

```
mkdir express-backend
```

```
cd express-backend
```

2)and then type following command:-

```
npm init -y
```

as shown below:-

```
D:\express-backend>npm init -y
Wrote to D:\express-backend\package.json:

{
  "name": "express-backend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "type": "commonjs"
}
```

And install it also :-

npm install mongoose

Create app.js file under your project folder and write following code:-

```
const express = require('express');

const cors = require('cors');

const mongoose = require('mongoose');

const app = express();

const PORT = 5000;


// Middleware

app.use(cors()); // Allow requests from frontend

app.use(express.json());


// MongoDB connection
```

```
const mongoURI = 'mongodb://localhost:27017/mydb'; // Local MongoDB URI (your db name is 'mydb')

mongoose.connect(mongoURI, { useNewUrlParser: true, useUnifiedTopology: true })

  .then(() => {

    console.log('MongoDB connected');

  })

  .catch((err) => {

    console.log('MongoDB connection error:', err);

  });


// MongoDB Schema for 'Service'

const serviceSchema = new mongoose.Schema({

  name: String,

  description: String

});


// MongoDB Model

const Service = mongoose.model('Service', serviceSchema);


// Route to get services from MongoDB

app.get('/api/services', async (req, res) => {

  try {

    const services = await Service.find();

    console.log('Fetched services:', services); // <-- add this

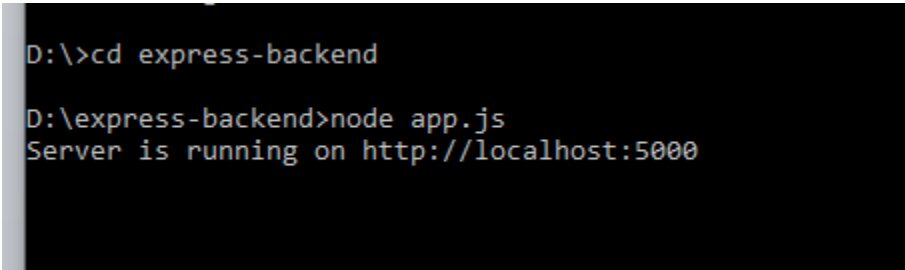
    res.json(services);

  } catch (err) {
```

```
    console.error('Error fetching services:', err); // <-- add this

    res.status(500).json({ message: 'Failed to fetch services', error: err });
  }
});

// Start the server
app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});
```



```
D:\>cd express-backend
D:\express-backend>node app.js
Server is running on http://localhost:5000
```

And check backend output at :-

<http://localhost:5000/api/services>

React – Fetch & Display Data

Assuming your React project is in a separate folder (e.g., react-pages-app):

Front End React js setup :-

```
D:\>npx create-react-app react-pages-app

Creating a new React app in D:\react-pages-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
|_
```

And install axios :-

```
D:\react-pages-app>npm install axios

added 3 packages, and audited 1344 packages in 4s

269 packages are looking for funding
  run `npm fund` for details
```

And install react-router-dom for routing :-

```
npm list react-router-dom
```

And your app.js file code :-

```
import logo from './logo.svg';

import './App.css';

import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';

import Navbar from './components/Navbar'; // Import Navbar

import Services from './pages/Services'; // Import Services
```

```
function App() {

  return (
```

```
<>
```

```
<h1> Welcome Home </h1>
```

```
<Router>
```

```
  <Navbar />
```

```
  <Routes>
```

```
    <Route path="/services" element={<Services />} /> { /* ? This is it */}
```

```
  </Routes>
```

```
</Router>
```

```
</>
```

```
);
```

```
}
```

```
export default App;
```

src/pages/Services.js file code :-

```
// src/pages/Services.js
```

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

```
import Navbar from '../components/Navbar';
```

```
import axios from 'axios';
```

```
function Services() {
```

```
  const [services, setServices] = useState([]);
```

```

useEffect(() => {
  axios.get('http://localhost:5000/api/services')
    .then((response) => {
      setServices(response.data);
    })
    .catch((error) => {
      console.error('Error fetching services:', error);
    });
}, []);

```

```

return (
  <>
    <Navbar />
    <h1>Our Services</h1>
    {services.length === 0 ? (
      <p>Loading services...</p>
    ) : (
      <ul>
        {services.map(service => (
          <li key={service._id}>
            <h3>{service.name}</h3>
            <p>{service.description}</p>
          </li>
        ))}
      </ul>
    )}
  </>
)

```

```
        </ul>
      )}
    </>
  );
}
```

```
export default Services;
```

now your `src/components/Navbar.js` file code :-

```
import { Link } from 'react-router-dom';
```

```
function Navbar() {
  return (
    <nav style={{ padding: '1rem', borderBottom: '1px solid #ccc' }}>
      <Link to="/">Home</Link> |{" "}
      <Link to="/services">Services</Link> |{" "}

    </nav>
  );
}
```

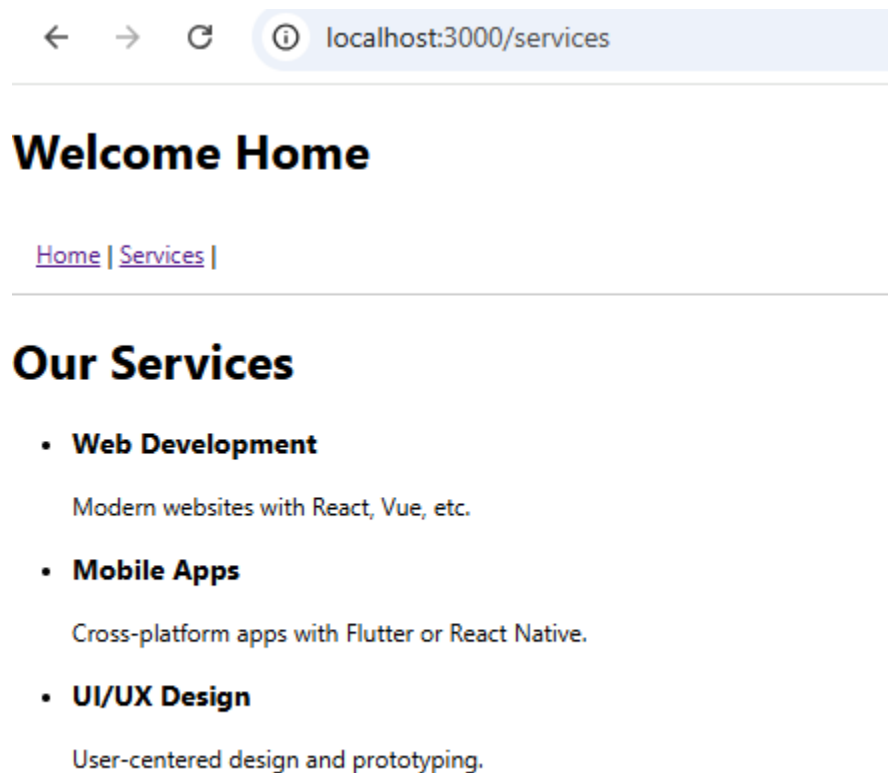
```
export default Navbar;
```


now you run it :-

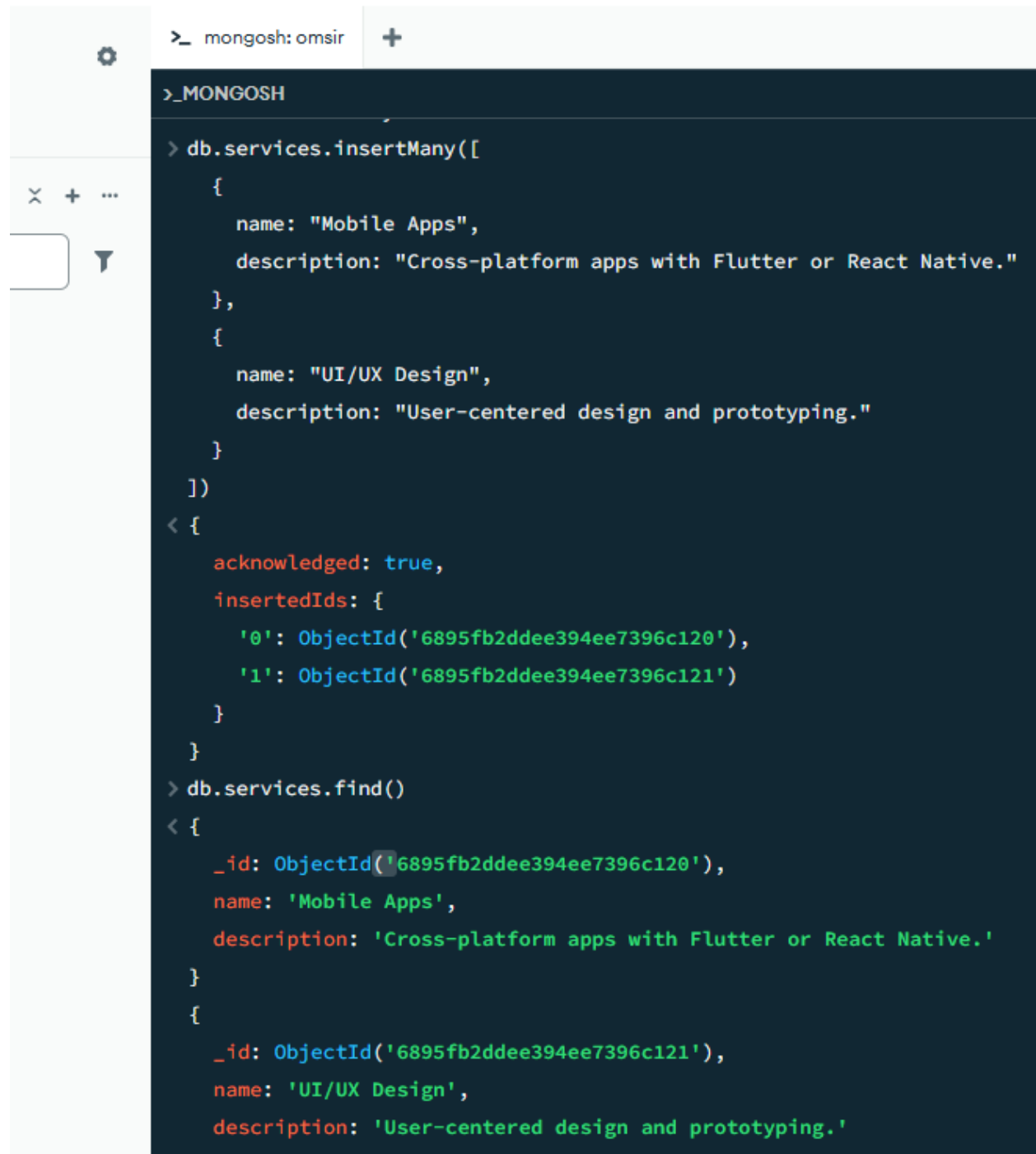
npm start

Output:-

<http://localhost:3000/services>



Note: first insert data to your services collection:-



```
>_ mongosh: omsir +
>_MONGOSH
> db.services.insertMany([
  {
    name: "Mobile Apps",
    description: "Cross-platform apps with Flutter or React Native."
  },
  {
    name: "UI/UX Design",
    description: "User-centered design and prototyping."
  }
])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6895fb2ddee394ee7396c120'),
    '1': ObjectId('6895fb2ddee394ee7396c121')
  }
}
> db.services.find()
< {
  _id: ObjectId('6895fb2ddee394ee7396c120'),
  name: 'Mobile Apps',
  description: 'Cross-platform apps with Flutter or React Native.'
}
{
  _id: ObjectId('6895fb2ddee394ee7396c121'),
  name: 'UI/UX Design',
  description: 'User-centered design and prototyping.'
```