

Asynchronous React in Next.js

```
cd ..  
PS C:\Users\saito\source\repos\React_Refresher\6_Asynchronous_React_in_NextJs> npx create-next-app@latest  
✓ What is your project named? ... async-react-project  
✓ Would you like to use the recommended Next.js defaults? » No, customize settings  
✓ Would you like to use TypeScript? ... No / Yes  
✓ Which linter would you like to use? » None  
✓ Would you like to use React Compiler? ... No / Yes  
✓ Would you like to use Tailwind CSS? ... No / Yes  
✓ Would you like your code inside a `src/` directory? ... No / Yes  
✓ Would you like to use App Router? (recommended) ... No / Yes  
? Would you like to customize the import alias (`@/*` by default)? » No / Yes
```

Fetch data from API

```
async function getData() {
    const res = await fetch("https://snowtooth-api-rest.fly.dev");
    if (!res.ok) {
        throw new Error('Failed to fetch lift status');
    }
    return res.json();
}
```

Modified Hotels

The screenshots show three different ways of displaying the same hotel data for "Snowtooth mountain".

Screenshot 1: Shows a JSON object representing the hotel details.

```
{"id": "01", "name": "Grand Summit", "capacity": 200, "pool": true, "spa": false, "avgCost": 1000, "skiln": true}, {"id": "02", "name": "Universal Hotel", "capacity": 1000, "pool": false, "spa": false, "avgCost": 250, "skiln": false}, {"id": "04", "name": "Harrison Hostel", "capacity": 500, "pool": false, "spa": false, "avgCost": 300, "skiln": true}]
```

Screenshot 2: Shows a list of hotel names.

- Grand Summit
- Universal Hotel
- Chalet Lodge
- Harrison Hostel
- Majestic Peak Lodge

Screenshot 3: Shows individual hotel details as key-value pairs.

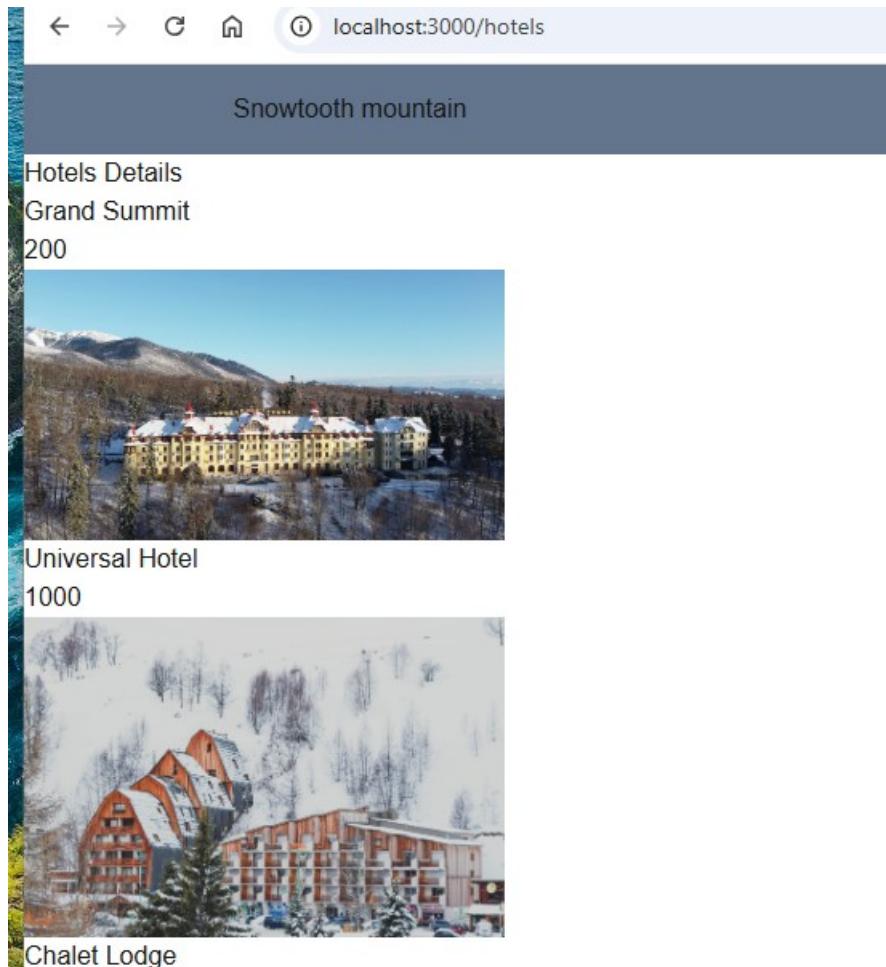
Hotel Name	Capacity	Pool	Avg Cost	Skiln
Grand Summit	200	True	1000	True
Universal Hotel	1000	False	250	False
Chalet Lodge	100	False	300	True
Harrison Hostel	500	False	300	True
Majestic Peak Lodge	300	True	1000	True

Display Client details using Client Component

Creating a Client component

Our HotelBlock is the Client component

Client component



Building a form

Typically we create the containers where the data would go then wire them, so everything works

Server actions are functions that can be executed on the server. These can be called from server components or client components to handle things like form submissions and other types of data mutations. We are going to put everything into the same file inside “contact” folder.

```
async function submitForm(formData) {
  "use server"; // directive it is a server-side function that can be called from client-side code
}
```

The screenshot shows a contact form with a light gray background and rounded corners. At the top center, it says "Contact us!". Below that is a form field labeled "Email" with the value "abc@microsoft.com". Underneath is another field labeled "Message" containing the text "Have we successfully built a server action?". At the bottom is a blue button labeled "Send Message".

we see in the server console

```
✓ Compiled in 545ms
  GET /contact 200 in 2.9s (compile: 2.0s, render: 893ms)
formFields {
  email: 'abc@microsoft.com',
  message: 'Have we successfully built a server action?'
}
TODO: send these form field values to a backend service
  POST /contact 200 in 14ms (compile: 4ms, render: 10ms)
□
```

Next steps:

Next.js data storage – learning to store data in databases, Rest API, GraphQL APIs

React server components

Caching

Look into <https://nextjs.org/docs> for all these things mentioned above