

## Copilot

Certainly! The ?? is called the **nullish coalescing operator** in JavaScript. Let's break down how it works:

### 1. Purpose:

- The nullish coalescing operator provides a way to handle **null** or **undefined** values by providing a fallback value when the left-hand side expression evaluates to one of these falsy values.
- It's often used as a concise way to provide a default value when the original value is null or undefined.

### 2. Syntax:

- The syntax is `expression1 ?? expression2`.
- If `expression1` is **not null** or **not undefined**, it returns `expression1`.
- If `expression1` is **null** or **undefined**, it returns `expression2`.

### 3. Example:

```
const value = null;
const defaultValue = "Fallback Value";

const result = value ?? defaultValue;
// If value is null or undefined, result will be "Fallback Value"
// Otherwise, result will be the actual value (if not null/undefined)
```

### 4. Use Cases:

- **Default Values:** It's commonly used to provide default values for variables or function parameters.
- **Avoiding Falsy Values:** Unlike the logical OR (||) operator, which considers empty strings, 0, and false as falsy values, the nullish coalescing operator only considers null and undefined as falsy.

In your specific code snippet:

```
fetcher?.data?.find((el) => el.id === item.pizzaId)?.ingredients ?? []
```

- If `fetcher.data` is null or undefined, it returns an empty array (`[]`).
- Otherwise, it finds the menu item with an id matching `item.pizzaId` and retrieves its ingredients.

Feel free to ask if you have any more questions! 😊