Mapping Returned Data

Deborah Kurata
CONSULTANT | SPEAKER | AUTHOR | MVP | GDE

@deborahkurata | blogs.msmvps.com/deborahk/

Why Map Returned Data?

Modify a value: `price = price * 1.5`

Transform a value: 'Y' -> true; 'N' -> false

Change a field name: p_nm -> productName

Add a calculated field: `profit = price - cost`

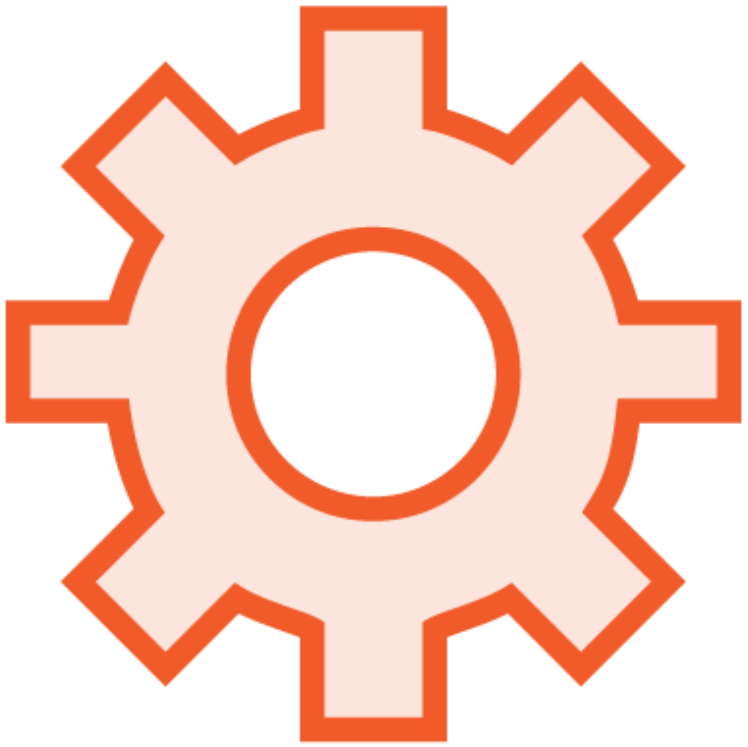
Simple mapping
doesn't always work.

Module Overview

Mapping an HTTP response

Mapping array elements

RxJS Features



`map`

Mapping a Simple Observable

```
of(2, 4, 6)
    .pipe(
      map(item => item * 2)
    ).subscribe(console.log); // 4 8 12
```

Mapping an HTTP Response

Product Service

```
private productsUrl = 'api/products';

product$ =
  this.http.get<Product>(`${this.productsUrl}/${id}`);
```

Response

{hammer}

Response

[{cart}, {hammer}]

Backend Server

```
export interface Product {
  id: number;
  productName: string;
  productCode: string;
}
```

```
{
  id: undefined,
  productId: 2,
  productName: undefined
  name: 'cart',
  productCode: 'GDN-0023'
}
```

```
{
  productId: 2,
  name: 'cart',
  productCode: 'GDN-0023'
}
```

Mapping an HTTP Response

Map the emitted array

Map each element in the array

Transform each element

Mapping an Emitted Array

```
products$ = this.http.get<Product[]>(this.productsUrl)
  .pipe(
    map(products =>
        ...
    ),
    catchError(this.handleError)
);
```

Mapping an HTTP Response

Map the emitted array

Map each element in the array

Mapping Array Elements

```
products$ = this.http.get<Product[]>(this.productsUrl)
 .pipe(
   map(products =>
     products.map(product => ...)
   ),
   catchError(this.handleError)
);
```

Mapping an HTTP Response

Map the emitted array

Map each element in the array

Transform each element

Transforming Array Elements

```
products.map(product => ({
   id: product.id,
   productName: product.productName,
   productCode: product.productCode,
   description: product.description,
   price: product.price * 1.5,
   searchKey: [product.productName]
}) as Product)
```

```
products.map(product => ({
   ...product,
   price: product.price * 1.5,
   searchKey: [product.productName]
}) as Product)
```

Checklist: Mapping an HTTP Response

- Map the emitted array
- Map the array elements
- Transform each array element

Checklist: Mapping an Emitted Array

Map the array

```
products$ = this.http.get<Product[]>(this.url)
  .pipe(
    map(products => ...)
  );
```

Checklist: Mapping Array Elements

Use the array's map method

```
products$ = this.http.get<Product[]>(this.url)
  .pipe(
    map(products =>
      products.map(product => { ... })
    )
  );
```

Checklist: Transforming Array Elements

- Define an object literal of the desired type
- Leverage the spread operator
- Replace or add fields as needed
- Use the as clause to type the result

```
products.map(product => ({
   ...product,
   price: product.price * 1.5,
   searchKey: [product.productName]
}) as Product)
```