Refactor Cart Service

Currently: Basic code for finding item in cart

© luv2code LLC

Currently: Basic code for finding item in cart

File: cart.service.ts

```
    // check if we already have the item in our cart
    let alreadyExistsInCart: boolean = false;
    let existingCartItem: CartItem = undefined;

    if (this.cartItems.length > 0) {
      // find the item in the cart based on item id

      for (let tempCartItem of this.cartItems) {
        if (tempCartItem.id === theCartItem.id) {
          existingCartItem = tempCartItem;
          break;
        }
      }

      // check if we found it
      alreadyExistsInCart = (existingCartItem != undefined)
    }
```


© luv2code LLC

Currently: Basic code for finding item in cart

File: cart.service.ts

```
// check if we already have the item in our cart
let alreadyExistsInCart: boolean = false;
let existingCartItem: CartItem = undefined;

if (this.cartItems.length > 0) {
  // find the item in the cart based on item id

  for (let tempCartItem of this.cartItems) {
    if (tempCartItem.id === theCartItem.id) {
      existingCartItem = tempCartItem;
      break;
    }
  }

  // check if we found it
  alreadyExistsInCart = (existingCartItem != undefined)
}
```


© luv2code LLC

Currently: Basic code for finding item in cart

File: cart.service.ts

```
// check if we already have the item in our cart
let alreadyExistsInCart: boolean = false;
let existingCartItem: CartItem = undefined;

if (this.cartItems.length > 0) {
  // find the item in the cart based on item id

  for (let tempCartItem of this.cartItems) {
    if (tempCartItem.id === theCartItem.id) {
      existingCartItem = tempCartItem;
      break;
    }
  }

  // check if we found it
  alreadyExistsInCart = (existingCartItem != undefined)
}
```

Let's refactor this code

© luv2code LLC

Array.find(...)

© luv2code LLC

Array.find(...)

- Method returns the first element in an array that passes a given test

© luv2code LLC

Array.find(...)

- Method returns the first element in an array that passes a given test

```
existingCartItem = this.cartItems.find( tempCartItem => tempCartItem.id === theCartItem.id );
```


© luv2code LLC

Array.find(...)

- Method returns the first element in an array that passes a given test

```
existingCartItem = this.cartItems.find( tempCartItem => tempCartItem.id === theCartItem.id );
```

the `find` method

instance method available on all arrays

© luv2code LLC

Array.find(...)

- Method returns the first element in an array that passes a given test

```
existingCartItem = this.cartItems.find( tempCartItem => tempCartItem.id === theCartItem.id );
```

the `find` method

instance method available on all arrays

© luv2code LLC

Array.find(...)

- Method returns the first element in an array that passes a given test

```
existingCartItem = this.cartItems.find( tempCartItem => tempCartItem.id === theCartItem.id );
```

- Executes the test for each element in the array until the test passes

© luv2code LLC

Array.find(...)

- Method returns the first element in an array that passes a given test

```
existingCartItem = this.cartItems.find( tempCartItem => tempCartItem.id === theCartItem.id );
```

- Executes the test for each element in the array until the test passes
- If test passes, then returns the first element in the array that passed

© luv2code LLC

Array.find(...)

- Method returns the first element in an array that passes a given test

```
existingCartItem = this.cartItems.find( tempCartItem => tempCartItem.id === theCartItem.id );
```

- Executes the test for each element in the array until the test passes
- If test passes, then returns the first element in the array that passed
- If test fails for ALL elements in the array, then returns `undefined`

© luv2code LLC

Array.find(...)

- Method returns the first element in an array that passes a given test

```
existingCartItem = this.cartItems.find( tempCartItem => tempCartItem.id === theCartItem.id );
```

- Executes the test for each element in the array until the test passes
- If tes

www.luv2code.com/array-find

- If test fails for ALL elements in the array, then returns `undefined`

© luv2code LLC

Array.find(...)

© luv2code LLC

Array.find(...)

```
for (let tempCartItem of this.cartItems) {

  if (tempCartItem.id === theCartItem.id) {
     existingCartItem = tempCartItem;
     break;
  }

}
```


© luv2code LLC

Array.find(...)

Current array element

```
for (let tempCartItem of this.cartItems) {

  if (tempCartItem.id === theCartItem.id) {
     existingCartItem = tempCartItem;
     break;
  }

}
```


© luv2code LLC

Array.find(...)

Current array element

Test Conditional

```
for (let tempCartItem of this.cartItems) {

  if (tempCartItem.id === theCartItem.id) {
     existingCartItem = tempCartItem;
     break;
  }

}
```


© luv2code LLC

Array.find(...)

Current array element

Test Conditional

```
for (let tempCartItem of this.cartItems) {

  if (tempCartItem.id === theCartItem.id) {
     existingCartItem = tempCartItem;
     break;
  }

}
```

```
existingCartItem = this.cartItems.find( tempCartItem => tempCartItem.id === theCartItem.id );
```


© luv2code LLC

Array.find(...)

Current array element

Test Conditional

```
for (let tempCartItem of this.cartItems) {

  if (tempCartItem.id === theCartItem.id) {
     existingCartItem = tempCartItem;
     break;
  }

}
```

```
existingCartItem = this.cartItems.find( tempCartItem => tempCartItem.id === theCartItem.id );
```

Current array element

© luv2code LLC

Array.find(...)

Current array element

Test Conditional

```
for (let tempCartItem of this.cartItems) {

  if (tempCartItem.id === theCartItem.id) {
     existingCartItem = tempCartItem;
     break;
  }

}
```

```
existingCartItem = this.cartItems.find( tempCartItem => tempCartItem.id === theCartItem.id );
```

Current array element

Test Conditional

© luv2code LLC

Array.find(...)

Current array element

Test Conditional

```
for (let tempCartItem of this.cartItems) {

  if (tempCartItem.id === theCartItem.id) {
     existingCartItem = tempCartItem;
     break;
  }

}
```

Executes test for each element in the array until test passes

```
existingCartItem = this.cartItems.find( tempCartItem => tempCartItem.id === theCartItem.id );
```

Current array element

Test Conditional

© luv2code LLC

Array.find(...)

Current array element

Test Conditional

```
for (let tempCartItem of this.cartItems) {

  if (tempCartItem.id === theCartItem.id) {
     existingCartItem = tempCartItem;
     break;
  }

}
```

Returns first element that passes else returns `undefined`

Executes test for each element in the array until test passes

```
existingCartItem = this.cartItems.find( tempCartItem => tempCartItem.id === theCartItem.id );
```

Current array element

Test Conditional

© luv2code LLC

Refactored: Before and After

© luv2code LLC

Refactored: Before and After

BEFORE

```
// check if we already have the item in our cart
let alreadyExistsInCart: boolean = false;
let existingCartItem: CartItem = undefined;

if (this.cartItems.length > 0) {
  // find the item in the cart based on item id

  for (let tempCartItem of this.cartItems) {
    if (tempCartItem.id === theCartItem.id) {
      existingCartItem = tempCartItem;
      break;
    }
  }

  // check if we found it
  alreadyExistsInCart = (existingCartItem != undefined)
}
```


© luv2code LLC

Refactored: Before and After

BEFORE

```
// check if we already have the item in our cart
let alreadyExistsInCart: boolean = false;
let existingCartItem: CartItem = undefined;

if (this.cartItems.length > 0) {
  // find the item in the cart based on item id

  for (let tempCartItem of this.cartItems) {
    if (tempCartItem.id === theCartItem.id) {
      existingCartItem = tempCartItem;
      break;
    }
  }

  // check if we found it
  alreadyExistsInCart = (existingCartItem != undefined)
}
```


© luv2code LLC

Refactored: Before and After

BEFORE

```
// check if we already have the item in our cart
let alreadyExistsInCart: boolean = false;
let existingCartItem: CartItem = undefined;

if (this.cartItems.length > 0) {
  // find the item in the cart based on item id

  for (let tempCartItem of this.cartItems) {
    if (tempCartItem.id === theCartItem.id) {
      existingCartItem = tempCartItem;
      break;
    }
  }

  // check if we found it
  alreadyExistsInCart = (existingCartItem != undefined)
}
```

AFTER

```
// check if we already have the item in our cart
let alreadyExistsInCart: boolean = false;
let existingCartItem: CartItem = undefined;

if (this.cartItems.length > 0) {
  // find the item in the cart based on item id

  existingCartItem = this.cartItems.find( tempCartItem => tempCartItem.id === theCartItem.id );

  // check if we found it
  alreadyExistsInCart = (existingCartItem != undefined)
}
```


© luv2code LLC

Refactored: Before and After

BEFORE

```
// check if we already have the item in our cart
let alreadyExistsInCart: boolean = false;
let existingCartItem: CartItem = undefined;

if (this.cartItems.length > 0) {
  // find the item in the cart based on item id

  for (let tempCartItem of this.cartItems) {
    if (tempCartItem.id === theCartItem.id) {
      existingCartItem = tempCartItem;
      break;
    }
  }

  // check if we found it
  alreadyExistsInCart = (existingCartItem != undefined)
}
```

AFTER

```
// check if we already have the item in our cart
let alreadyExistsInCart: boolean = false;
let existingCartItem: CartItem = undefined;

if (this.cartItems.length > 0) {
  // find the item in the cart based on item id

  existingCartItem = this.cartItems.find( tempCartItem => tempCartItem.id === theCartItem.id );

  // check if we found it
  alreadyExistsInCart = (existingCartItem != undefined)
}
```


© luv2code LLC