

Memoization

Shawn Leberknight

Software Engineer

So what is it?

Memoization is an optimization technique used in many programming languages to reduce the number of redundant, expensive function calls.

How does it work?

- Caching the values that the function returns after its initial execution.
- If the input values remain the same, the memoized function returns the cached response.
- The program does not have to recalculate anything.

Simple Example

Before memoization

```
function sum(a, b) {  
  console.log("add");  
  return a + b;  
}  
console.log(sum(1, 2));  
console.log(sum(1, 2));  
  
// will output the following:  
// add  
// 3  
// add  
// 3
```

After memoization

```
function sum(a, b) {  
  console.log("add");  
  return a + b;  
}  
  
const memSum = memoize(sum);  
console.log(memSum(1, 2));  
console.log(memSum(1, 2));  
  
// will output the following:  
// add  
// 3  
// 3
```

Some live coding!!!

Some other things to remember

Network & database calls

- Be wary of using memoization for this and know the possible issues
- Expire cache after given period of time
- Clear cache manually if needed

React.memo, useCallback, & useMemo

Resources

- <https://dev.to/nas5w/what-is-memoization-4lod>
- <https://medium.com/better-programming/react-memo-vs-memoize-71f85eb4e1a>
- <https://codeburst.io/understanding-memoization-in-3-minutes-2e58daf33a19>
- <https://reactjs.org/docs/hooks-reference.html#usememo>
- <https://reactjs.org/docs/react-api.html#reactmemo>
- <https://www.digitalocean.com/community/tutorials/react-usememo>
- <https://www.digitalocean.com/community/tutorials/react-learning-react-memo>
- <https://www.youtube.com/watch?v=3cYtqrNUiVw>