Saif Mujawar
@webbysaif

What is the use of promises in Javascript?

Saif Mujawar
@webbysaif

Before promises, **callbacks** were used to handle **asynchronous** operations. But due to the **limited functionality** of callbacks, using multiple callbacks to handle **asynchronous** code can lead to **unmanageable** code.

Promise object has four states :-

- **Pending** - Initial state of promise. This state represents that the promise has neither been fulfilled nor been rejected, it is in the pending state.
-
- **Fulfilled** - This state represents that the promise has been fulfilled, meaning the async operation is completed.
-
- **Rejected** - This state represents that the promise has been rejected for some reason, meaning the async operation has failed.
-
- **Settled** - This state represents that the promise has been either rejected or fulfilled.

Saif Mujawar
@webbysaif

A promise is created using the **Promise** constructor which takes in a callback function with two parameters, **resolve** and **reject** respectively.

new Promise()

resolve()
Go to next action

reject()
Handle Error

resolve is a function that will be called when the async operation has been successfully completed.

reject is a function that will be called, when the async operation fails or if some error occurs.

Saif Mujawar
@webbysaif

Example of a promise:

Promises are used to handle asynchronous operations like server requests, for ease of understanding, we are using an operation to calculate the sum of three elements.

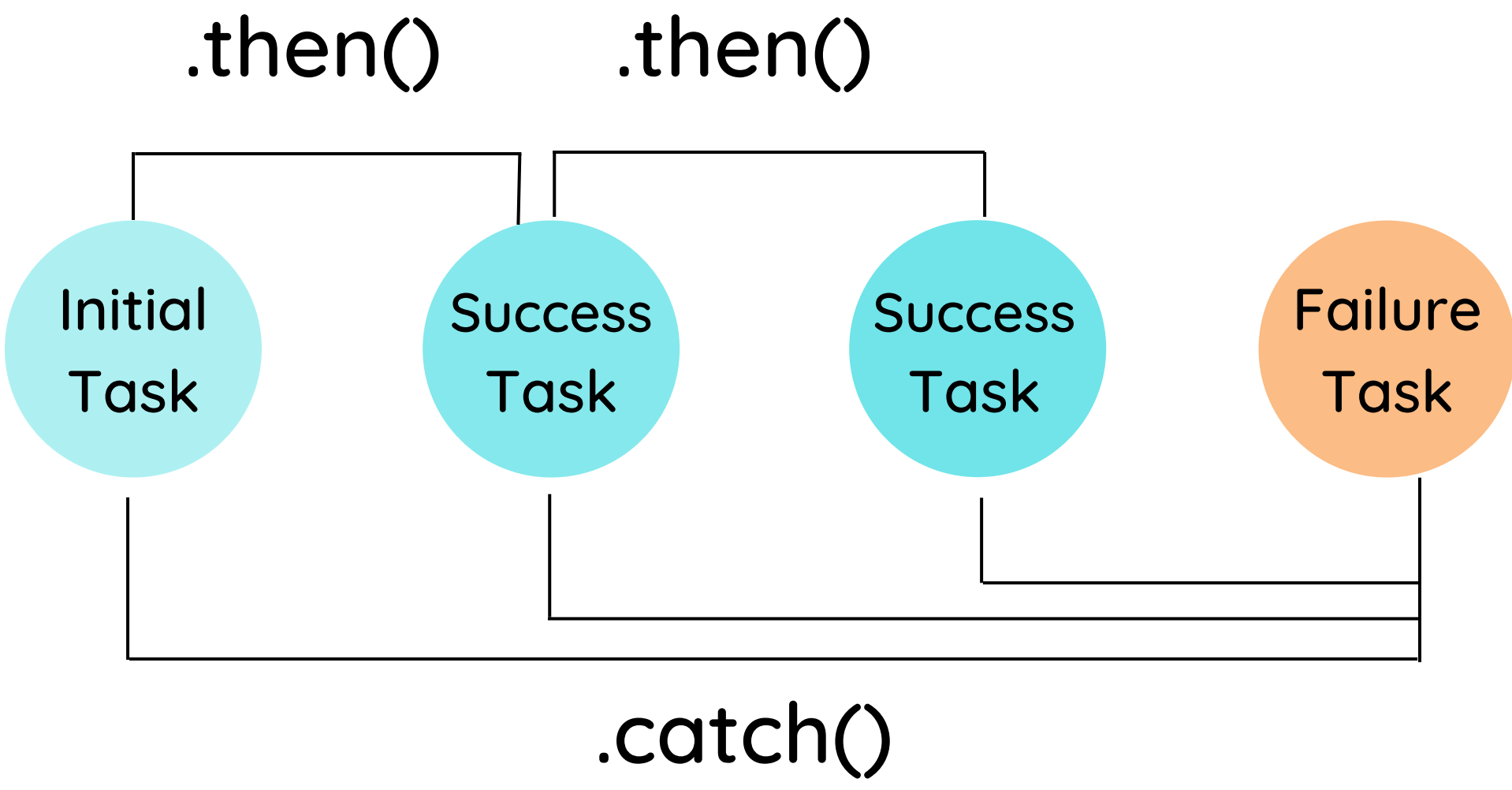
Promise Example

```
function sumOfThreeElements(...elements){
  return new Promise((resolve,reject)=>{
    if(elements.length > 3 ){
      reject("Only three elements or less are allowed");
    }
    else{
      let sum = 0;
      let i = 0;
      while(i < elements.length){
        sum += elements[i];
        i++;
      }
      resolve("Sum has been calculated: "+sum);
    }
  })
}
```

Saif Mujawar
@webbysaif

In the code above, we are calculating the sum of three elements, if the length of the elements array is more than 3, a promise is rejected, or else the promise is resolved and the sum is returned.

We can consume any promise by attaching then() and catch() methods to the consumer.



Saif Mujawar
@webbysaif

then() method is used to access the result when the promise is fulfilled.

catch() method is used to access the result/error when the promise is rejected. In the code below, we are consuming the promise:

output explaination

```
sumOfThreeElements(4, 5, 6)
.then(result=> console.log(result))
.catch(error=> console.log(error));
// In the code above, the promise is fulfilled so the
then() method gets executed

sumOfThreeElements(7, 0, 33, 41)
.then(result => console.log(result))
.catch(error=> console.log(error));
// In the code above, the promise is rejected hence the
catch() method gets executed
```

Turn on notifications 🔔

And That's it!!!

Did you find it useful?

Saif Mujawar
@webbysaif

Follow for more!!