

Promises

Earlier we were passing the function to another function. Now we are attaching the function to the promise object. Promise object is the placeholder which is filled from an asynchronous operation later by value. A container for future value. A promise is the object that represents the eventual completion or failure of an asynchronous operation to solve the issue of the callback hell and inversion of control. A promise can be in three states

1. Fulfilled
2. Rejected
3. Pending

Example:

```
const myPromise = new Promise((resolve, reject) => {  
  // Asynchronous operation, e.g., fetching data  
  setTimeout(() => {  
    const success = true;  
    if (success) {  
      resolve("Data successfully fetched");  
    } else {  
      reject("Error fetching data");  
    }  
  }, 2000); // Simulating 2 seconds delay  
});  
  
// Consuming the promise  
myPromise  
  .then((result) => {  
    console.log("Success:", result);  
  })  
  .catch((error) => {  
    console.error("Error:", error);  
  });
```