Promise,
Polyfill ,
Promise.race,
Promise.allSettled &
Promise.all
In Javascript

Amit Ranjan
Web Developer

PROMISES IN JAVASCRIPT

- A Promise in JavaScript is an object representing the eventual completion or failure of an asynchronous operation.

- Promises are used to handle asynchronous tasks like fetching data from a server, reading files, etc.

- Promises provide a cleaner and more manageable way to work with asynchronous code compared to traditional callback functions.

Example :

```
let ourPromise = new
Promise((resolve, reject) => {
   // asynchronous operation
   if (/* success */) {
      resolve('Success!');
   } else {
      reject('Failure!');
   }
});

ourPromise.then(result => {
   console.log(result); // 'Success!' if
resolved
}).catch(error => {
   console.log(error); // 'Failure!' if
rejected
});
```

Polyfill for Promises

- A polyfill is code that adds support for functionality in older browsers that do not support certain modern JavaScript features, such as Promises.

- If a browser does not support Promises, a polyfill can be used to provide similar functionality.

Promise.race

- Promise.race is a method that takes an array of Promises and returns a new Promise that resolves or rejects as soon as one of the Promises in the array resolves or rejects.
- The result or error of the first settled Promise is used.

```
let promise1 = new Promise((resolve)
=> setTimeout(resolve, 800, 'First'));
let promise2 = new Promise((resolve)
=> setTimeout(resolve, 200,
'Second'));

Promise.race([promise1, promise2])
   .then((value) => {
      console.log(value); // 'Second'
because promise2 resolves first
   });
```

Promise.allSettled

1. Promise.allSettled is a method that takes an array of Promises and returns a new Promise that resolves after all of the Promises in the array have settled, either by resolving or rejecting.
2. It returns an array of objects, each describing the outcome of each Promise.

```
let promise1 = new Promise((resolve) =>
setTimeout(resolve, 800 , 'First'));
let promise2 = new Promise((reject) =>
setTimeout(reject, 200, 'Second'));

Promise.allSettled([promise1, promise2])
   .then((results) => {
      console.log(results);
      /* [
       { status: "fulfilled", value: "First" },
       { status: "rejected", reason: "Second" }
      */ ]
   });
```

Promise.all

1. Promise.all is a method that takes an array of Promises and returns a new Promise that resolves when all the Promises in the array have resolved or rejects as soon as one of the Promises rejects.
2. The resolved value is an array of resolved values from each Promise.

```
let promise1 = Promise.resolve('First');
let promise2 = Promise.resolve('Second');

Promise.all([promise1, promise2])
  .then((values) => {
    console.log(values); // ['First', 'Second']
  })
  .catch((error) => {
    console.error(error);
  });
```

Summary

Promise.race: Resolves or rejects as soon as any one of the Promises in an array settles.

Promise.allSettled: Waits for all Promises to settle and returns an array describing the outcome of each.

Promise.all: Resolves when all Promises in the array resolve, or rejects if any Promise rejects.

THANK

YOU

FOLLOW FOR MORE

Amit Ranjan
Web Developer