Promise.all()

vs

Promise.allSettled()

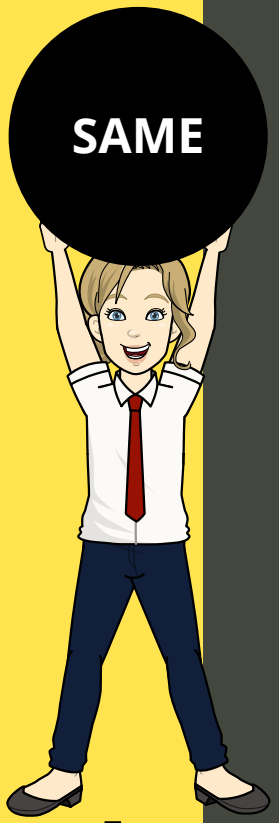
https://www.linkedin.com/in/shaluweb/

What are Promise.all() and Promise.allSettled() ?

Both Promise.all() and Promise.allSettled() methods, are the methods of a Promise object which are used to handle multiple promises results simultaneously. The input of both methods is an array containing promises which further contains some data within them.

Syntax for Promise.all() method

```
Promise.all([promise_1 , promise_2,  ...]).then(
    // do something...
)
```



Syntax for Promise.allSettled() method

```
Promise.all([promise_1 , promise_2,  ...]).then(
    // do something...
)
```

Promise.all()

```
const userDetails = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("User Details fetched ");
  }, 1000);
});

const buyerDetails = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("Buyer details fetched");
  }, 1000);
});

const productDetails = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("Product details fetched");
  }, 1000);
});

Promise.all([userDetails, buyerDetails, productDetails]).then(
  (values) => console.log(values)
);
```

Promise.all() method returns an array as an output containing promise data inside several indexes.

```
▼ (3) ['User Details fetched ', 'Buyer details fetched', 'Product details fetched']
    0: "User Details fetched "
    1: "Buyer details fetched"
    2: "Product details fetched"
    length: 3
  ▶ [[Prototype]]: Array(0)
```

Promise.allSettled()

```
const userDetails = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("User Details fetched ");
  }, 1000);
});

const buyerDetails = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("Buyer details fetched");
  }, 1000);
});

const productDetails = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("Product details fetched");
  }, 1000);
});

Promise.allSettled([userDetails, buyerDetails, productDetails]).then(
  (values) => console.log(values)
);
```

Promise.allSettled() method returns an array of objects and each of these objects further contains two properties further **status** and **value**

```
▼ (3) [{…}, {…}, {…}]
  ▶ 0: {status: 'fulfilled', value: 'User Details fetched '}
  ▶ 1: {status: 'fulfilled', value: 'Buyer details fetched'}
  ▶ 2: {status: 'fulfilled', value: 'Product details fetched'}
    length: 3
  ▶ [[Prototype]]: Array(0)
```

Promise.all() vs Promise.allSettled() ?

Promise.all()

this method **rejects itself** if any of the passed in promise input inside an array is rejected

Promise.allSettled()

this method **will not reject itself** if any of the passed in promise input inside an array is rejected

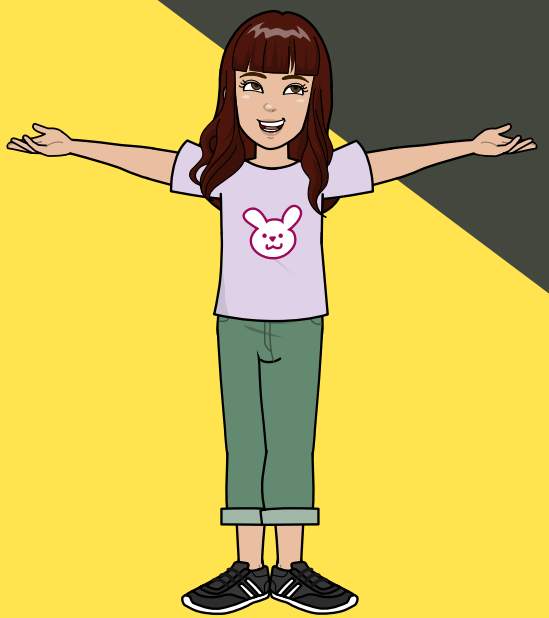
Promise.all()

```
const userDetails = new Promise((resolve, reject) => {
  setTimeout(() => {
    reject("User Details fetched ");
  }, 1000);
});

const buyerDetails = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("Buyer details fetched");
  }, 1000);
});

const productDetails = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("Product details fetched");
  }, 1000);
});

Promise.all([userDetails, buyerDetails, productDetails]).then(
  (values) => console.log(values)
);
```



```
Uncaught (in promise) User Details fetched
>
```

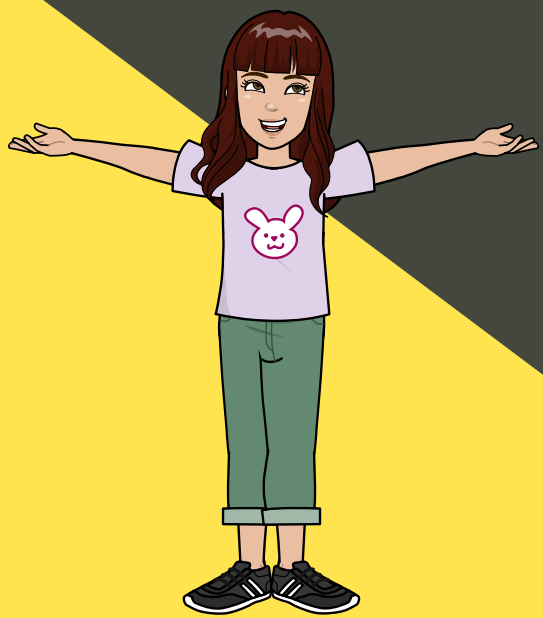
Promise.allSettled()

```
const userDetails = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("User Details fetched ");
  }, 1000);
});

const buyerDetails = new Promise((resolve, reject) => {
  setTimeout(() => {
    reject("Buyer details fetched");
  }, 1000);
});

const productDetails = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("Product details fetched");
  }, 1000);
});

Promise.allSettled([userDetails, buyerDetails, productDetails]).then(
  (values) => console.log(values)
);
```



```
▼ (3) [{…}, {…}, {…}]
  ▶ 0: {status: 'fulfilled', value: 'User Details fetched '}
  ▶ 1: {status: 'rejected', reason: 'Buyer details fetched'}
  ▶ 2: {status: 'fulfilled', value: 'Product details fetched'}
    length: 3
  ▶ [[Prototype]]: Array(0)
```

Shalu Chaddha

https://www.linkedin.com/in/shaluweb/

Thank You

Do you find it helpful?

like **share**