

# | Promise

# Promise

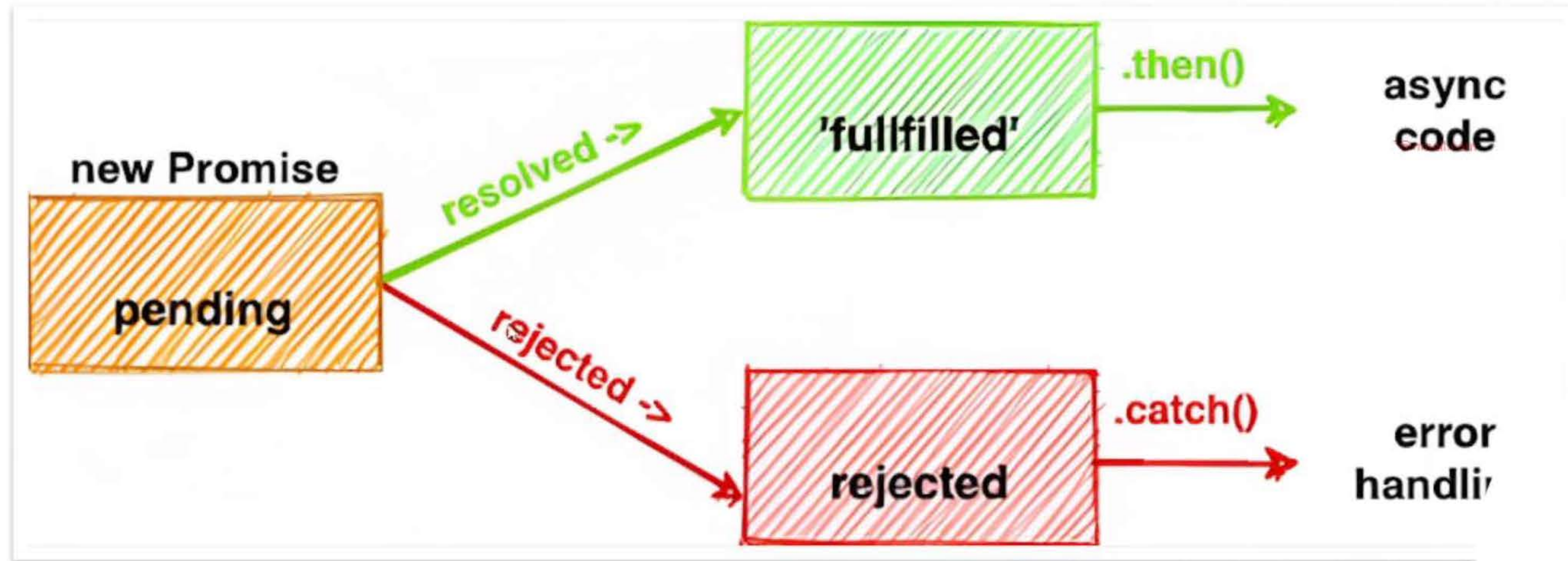
---

The Promise object represents the eventual **completion** (or **failure**) of an **asynchronous** operation and its resulting value. A Promise is in one of these states:





# Promise Structure



# Examples

```
let promise = new Promise(function(resolve,  
reject) {
```

executor function

```
  resolve("I'm Resolved!");
```

```
});
```

```
const consumer = () => {  
  promise.then(  
    result => {},  
    error => {}  
  )  
}
```

consuming function

 @tapasadhikary



# Create & Consume

PAGE 06

## create

```
const ride = new Promise((resolve, reject) => {  
  if (arrived) {  
    resolve('driver arrived 🚗');  
  } else {  
    ⚡ reject('driver bailed 😞');  
  }  
});
```

## consume

### then

function that handles fulfillment

```
ride  
  .then(value => {  
    console.log(value);  
    // driver arrived 🚗  
  })  
  ⚡ .catch(error => {  
    console.log(error);  
  })
```

### catch

handle rejection

# Fetch API

01

The `fetch()` method starts the process of fetching a resource from a server

02

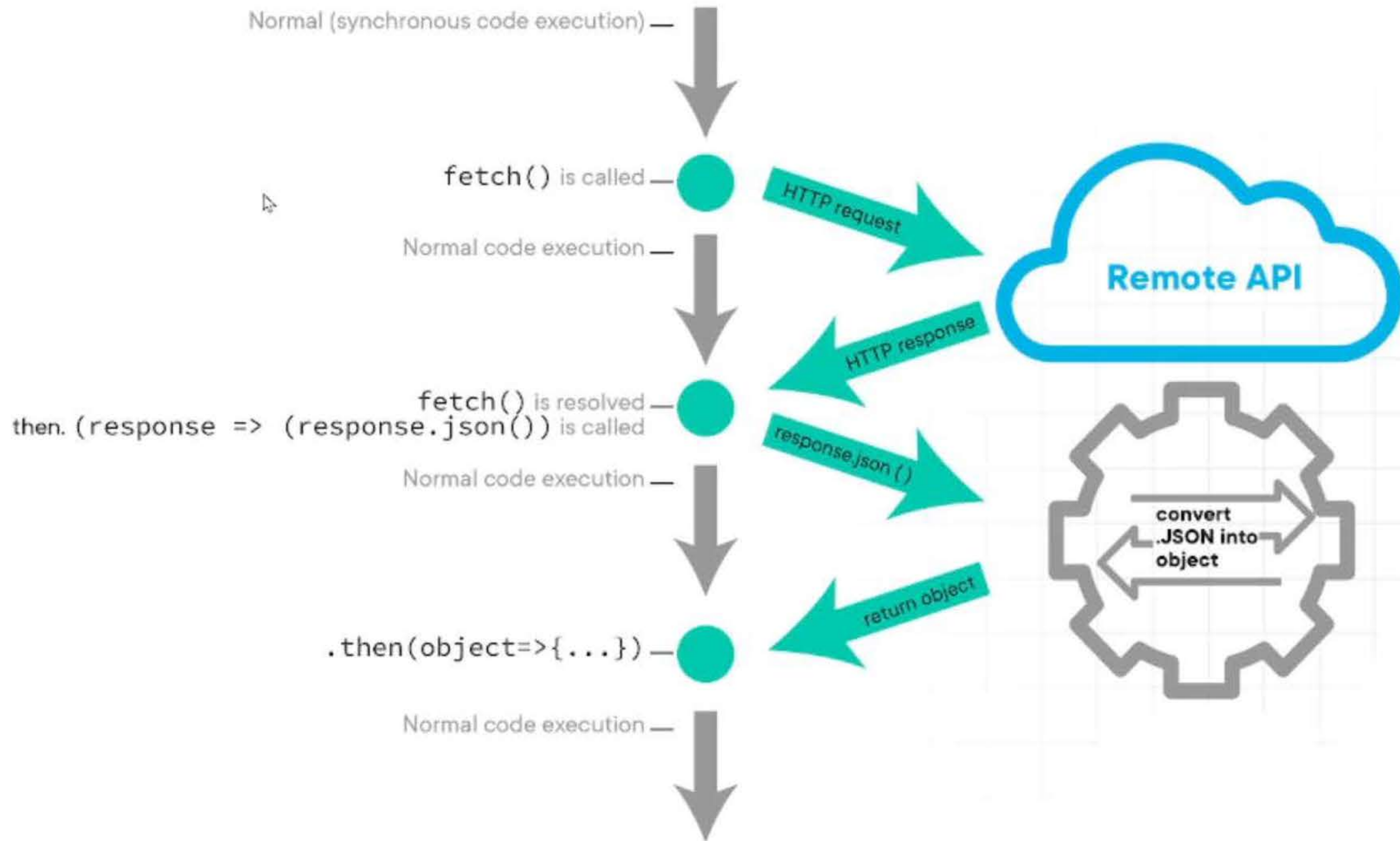
The `fetch()` method returns a Promise that resolves to a Response object.

03

A `fetch()` method only rejects when a network error is encountered.



# Fetch API



# | Async/Await



# Async/Await

- ❖ “`async/await`” is a special syntax to work with promises in a more comfortable way
- ❖ It's surprisingly `easy to understand` and `use`.
- ❖ The `await` keyword can only be used inside an `async` function.
- ❖ The `await` keyword makes the function pause the execution and
- ❖ wait for a resolved promise before it continues

# Async/Await

Use `async` keyword  
before function  
definition

`await` keyword that works only  
inside `async` functions

`Fetch` keyword for  
network calls (i.e HTTP  
Client)

```
async function getUsers(url) {  
  let response = await fetch(url),  
  let data = await response.json();  
  console.log(data);  
  return data;  
}
```

If promise fulfills, you  
will get the values  
back else rejected  
value is thrown

When you `await`, a  
promise function  
paused in non blocking  
way until it settles



# Fetch vs Async/Await

```
1  function getFetch1(getURL) {  
2      fetch(getURL)  
3          .then(resp => resp.json())  
4          .then(data => {  
5              console.log(data)  
6          })  
7          .catch(err => {  
8              console.log(err.message)  
9          })  
10 }
```

```
1  async function getFetch2(getURL) {  
2      try {  
3          const resp = await fetch(getURL)  
4          const data = await resp.json()  
5          console.log(data)  
6      }  
7      catch (err) {  
8          console.log(err);  
9      }  
10 }
```



**We will learn more about JS in  
our next videos**