Google Developer Student Clubs
Techno India University

Understading Promises, Callbacks , APIs

Gourav Ghosal - Web Developer | Graphics Designer
bit.ly/gg221

Topics for Discussion:
1. Promises
2. Callbacks
3. fetch() function
4. Tailwind css
5. Project Demo

What is a Promise?

The Promise object supports two properties: state and result.

While a Promise object is "pending" (working), the result is undefined.

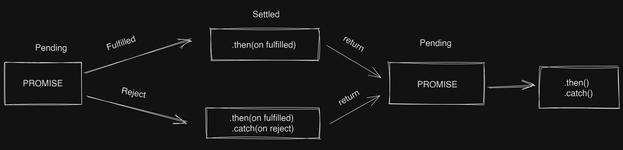
When a Promise object is "fulfilled", the result is a value.

When a Promise object is "rejected", the result is an error object.

```
let p = new Promise(function(resolve, reject) {
  let x = 1+1;

  if (x == 2) {
    resolve("OK");
  } else {
    reject("Error");
  }
});

p.then((value)=> displayer(value))
p.catch((error)=>displayer(error))
```



What is a Callback?

Callback is a function which is to be executed after another function has finished execution. A callback's primary purpose is to execute code in response to an event. These events might be user-initiated, such as mouse clicks or typing. With a callback, you may instruct your application to "execute this code every time the user clicks a key on the keyboard."

```
const printCount=(count,callback)=>
    {
        setTimeout(() => {
            callback()
            document.body.append(count)
        }, 2000);
    }

    const delayCounter=()=>
    {
        setTimeout(() => {
        document.body.append('delayed Output')
        }, 5000);
    }

    printCount(1,delayCounter)
```

Welcome to Callback Hell

```
fs.readdir(source, function (err, files) {
  if (err) {
    console.log('Error finding files: ' + err)
  } else {
    files.forEach(function (filename, fileIndex) {
      console.log(filename)
      gm(source + filename).size(function (err, values) {
        if (err) {
          console.log('Error identifying file size: ' + err)
        } else {
          console.log(filename + ' : ' + values)
          aspect = (values.width / values.height)
          widths.forEach(function (width, widthIndex) {
            height = Math.round(width / aspect)
            console.log('resizing ' + filename + 'to ' + height + 'x' + height)
            this.resize(width, height).write(dest + 'w' + width + '_' + filename, function(err) {
              if (err) console.log('Error writing file: ' + err)
            })
          }.bind(this))
        }
      })
    })
  }
})
```

The fetch() Function

The Fetch API provides a JavaScript interface for accessing and manipulating parts of the HTTP pipeline, such as requests and responses. It also provides a global fetch() method that provides an easy, logical way to fetch resources asynchronously across the network.

```
fetch('http://example.com/movies.json')
  .then((response) => response.json())
  .then((data) => console.log(data))
  .catch((err)=>console.log(err))
```

Application Programming Interface

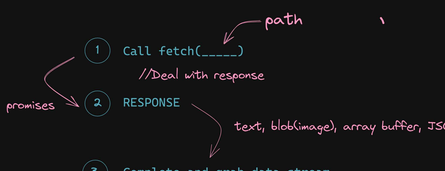
PUBLIC APIs

AUTHENTICATED APIs

API KEY

"The unsung heroes of our connected world"

GET, POST HTTP requests



Find me at:

LinkedIn : Gourav Ghosal
Instagram : g.o.u.r.a.v_221b
Website: bit.ly/gg221