EventEmitter

Akash Pundir

System Programming I

School of Computer Science and Engineering

- **Node.js uses events module to create and handle custom events. The EventEmitter class can be used to create and handle custom events module.**
- **Key capabilities of EventEmitter:**

- **Emitting events:** Use the emit(event, [arg1, arg2, ...]) method to signal the occurrence of specific events within your code.
- **Attaching event listeners:** Employ the on(event, listener) method to register callback functions that will be invoked when the corresponding events are emitted.

Importing the Module and Creating an EventEmitter Instance

- Import the events module and create an instance of the EventEmitter class

```
const EventEmitter = require('events’);
```

- Create an instance of the EventEmitter class. This instance can be used to emit events and handle listeners.

```
const myEmitter = new EventEmitter();
```

Event Registration - on Method:

The on method is used to register an event listener for a specific event. Multiple listeners can be attached to the same event.

```
myEmitter.on(‘firstevent', (arg1, arg2) => {
    // Event handler logic
    console.log('Event occurred with arguments:', arg1, arg2);
  });
```

Event Emission - emit Method:

- The emit method triggers an event. All registered listeners for that event will execute, and any additional arguments passed to emit will be provided to the listeners.

```
myEmitter.emit('firstevent', 'Hello', 'Hi');
```

Once Method:

- The once method registers a listener that will be called only once for a specific event.

```
myEmitter.once('firstevent', (arg1,arg2) => {
    console.log('This listener will be executed only once.');
  });
```

Try this..to understand better

```
const EventEmitter = require('events');
const myEmitter = new EventEmitter();

myEmitter.on('firstevent', (arg1, arg2) => {
    // Event handler logic
    console.log('Event occurred with arguments:', arg1, arg2);
  });
myEmitter.once('firstevent', (arg1,arg2) => {
    console.log('This listener will be executed only once.');
  });

for(var i=0;i<2;i++){
    myEmitter.emit('firstevent', 'Hello', 'Hi');
}
```

Removing Event Listeners - removeListener Method:

- The removeListener method is used to remove a specific listener for a given event.

```
const EventEmitter = require('events');
const myEmitter = new EventEmitter();

const eventHandler = (arg1, arg2) => {
    console.log('Event occurred with arguments:', arg1, arg2);
};

// Add the event handler
myEmitter.on('firstevent', eventHandler);

for (var i = 0; i < 5; i++) {

    // Remove the listener after the first emission
    if (i ==2) {
        myEmitter.removeListener('firstevent', eventHandler);
    }
myEmitter.emit('firstevent', 'Hello', 'Hi');
}
```

Removing All Listeners for an Event - removeAllListeners Method:

```
const EventEmitter = require('events');
const myEmitter = new EventEmitter();

// Event handler
const eventHandler = (arg1, arg2) => {
    console.log('Event occurred with arguments:', arg1, arg2);
};

// Add the event handler
myEmitter.on('firstevent', eventHandler);

for (var i = 0; i < 4; i++) {
    myEmitter.emit('firstevent', 'Hello', 'Hi');

    // Remove the listener after the first emission
    if (i ==2) {
        myEmitter.removeAllListeners('firstevent');
    }
}
```

Error Events:

The EventEmitter emits an 'error' event. If there is no listener for this event, unhandled errors can cause the Node.js process to terminate.

```
const EventEmitter = require('events');
const myEmitter = new EventEmitter();

// Event handler
const eventHandler = (arg1, arg2) => {
    console.log('Event occurred with arguments:', arg1, arg2);
};
// Add the event handler
myEmitter.on('firstevent', eventHandler);
// Add error event handler
myEmitter.on('error', (error) => {
    console.error('Error occurred:', error.message);
});
// Trigger 'error' event
myEmitter.emit('error', new Error('This is an example error.'));
```