



Task 2.2 : Rate Limiter FunctionCreate a function r...

1 message

Bikash <bikash.caim@gmail.com>
To: virtual.bikash@gmail.com

Sat, Mar 8, 2025 at 6:53 PM

JavaScript

Task 2: Rate Limiter Function

Create a function `rateLimiter(fn, limit)` that returns a new function. The returned function allows calling `fn` only once within a limit time in milliseconds. If it is called again before the limit is reached, it should return "Rate limit exceeded".

```
function rateLimiter(fn, limit) {
  let lastCalled = 0;

  return function(...args) {
    const now = Date.now();

    if (now - lastCalled >= limit || lastCalled === 0) {
      lastCalled = now;
      return fn(...args);
    } else {
      return "Rate limit exceeded";
    }
  };
}

// Example usage:
function myFunc(message) {
  return `Function called with: ${message}`;
}

const limitedFunc = rateLimiter(myFunc, 1000); // Limit to 1 call per second

console.log(limitedFunc("First call")); // Output: "Function called with: First call"
console.log(limitedFunc("Second call (too soon)")); // Output: "Rate limit exceeded"

setTimeout(() => {
  console.log(limitedFunc("Third call (after 1 second)")); // Output: "Function called with:
Third call (after 1 second)"
}, 1100);

console.log(limitedFunc("Fourth Call (too soon)")); //Output: Rate limit exceeded
```

Explanation:

1. `rateLimiter(fn, limit)` **function:**
 - o Takes the function `fn` to be rate-limited and the `limit` (in milliseconds) as arguments.

- Initializes `lastCalled` to 0. This will be updated with the timestamp of the last successful call.
- Returns a new anonymous function (the rate-limited function).

2. Returned (rate-limited) function:

- Gets the current timestamp using `Date.now()`.
- Checks if the difference between the current time and `lastCalled` is greater than or equal to the `limit` or if `lastCalled` is still its initial value of 0. If either of these conditions are true, it means enough time has passed since the last call (or it's the first call).
 - If the limit has not been exceeded, it updates `lastCalled` to the current time.
 - It then calls the original function `fn` using `fn(...args)` (passing any arguments received) and returns its result.
- If the limit has been exceeded (the time difference is less than the `limit`), it returns the string "Rate limit exceeded".
- The use of the spread operator (`...args`) allows the limited function to take any number of arguments.