**Таймер** – механизм, позволяющий генерировать событие или выполнить некоторое действие, через заданный промежуток времени.

Процесс Node.js работает до тех пор, пока есть события, требующие обработки. Если выполнить для таймера unref(), то события, генерируемые таймером, не будут учитываться при завершении работы Node.js. Метод unref() есть не только у таймеров, есть еще у серверов (server.unref()), сетевых сокетов (socket.unref()) и др.

Если кратко:

**1. setTimeout**

позволяет вызвать функцию один раз через определённый интервал времени.

**2. setInterval**

позволяет вызывать функцию регулярно, повторяя вызов через определённый интервал времени.

**3. ref**

противоположный unref

**4. unref**

timer является второстепенным, не нужно проверять, не влияет на завершение процесса.

1. **setTimeout**

Является макротаской; Принимает следующие параметры:

* коллбэк, который вызовется и исполнится по истечению n миллисекунд
* число миллисекунд, через которое нужно выполнить коллбэк
* дополнительные параметры, которые могут быть переданы в коллбэк-функцию

setTimeout не гарантирует, что коллбэк будет вызван ровно через n миллисекунд. Он гарантирует только то, что функция не будет вызвана раньше, чем через n миллисекунд. Это происходит из-за того, что микротаски (например, nextTick) могут задерживать EventLoop и отодвигать таймер setTimeout дальше в очереди.

1. **setInterval**

Также является макротаской; По сути, это бесконечный цикл, который будет выполняться с определенной периодичностью. Принимает следующие параметры:

* коллбэк, который будет вызываться каждые n миллисекунд
* число миллисекунд периодичности коллбэка

setInterval также не гарантирует точного выполнения вплоть до миллисекунды по той же причине: таймер – это макротаска.

1. **unref**

Как setInterval, так и setTimeout возвращают объект типа Timeout. Мало того, что выполнение этого таймера можно отменить, передав объект Timeout в параметры функции clearTimeout или clearInterval соответственно, но можно также и изменить поведение этого таймера с помощью функций ref и unref.

Если вызвать для объекта таймера timer.unref(), то по сути это указывает, что нам похую на этот таймер и он является второстепенным. То есть этот таймер не надо проверять, чтобы понимать, завершится процесс или нет. В 5-ой лабе это помогает сделать так, чтобы даже когда у нас выполняется команды sc (выполнение коммита каждые n секунд через setInterval), то использование scTimer.unref() сделает так, чтобы я в любой момент мог прописать команду sd (дисконнект от сервера) и он отключится, несмотря на то, что у нас есть незавершенный таймер для коммитов.

1. **ref**

То же самое, что и unref, только наоборот. С помощью этой команды можно при желании отменить действие unref. На практике применяется редко.