



Technology Series

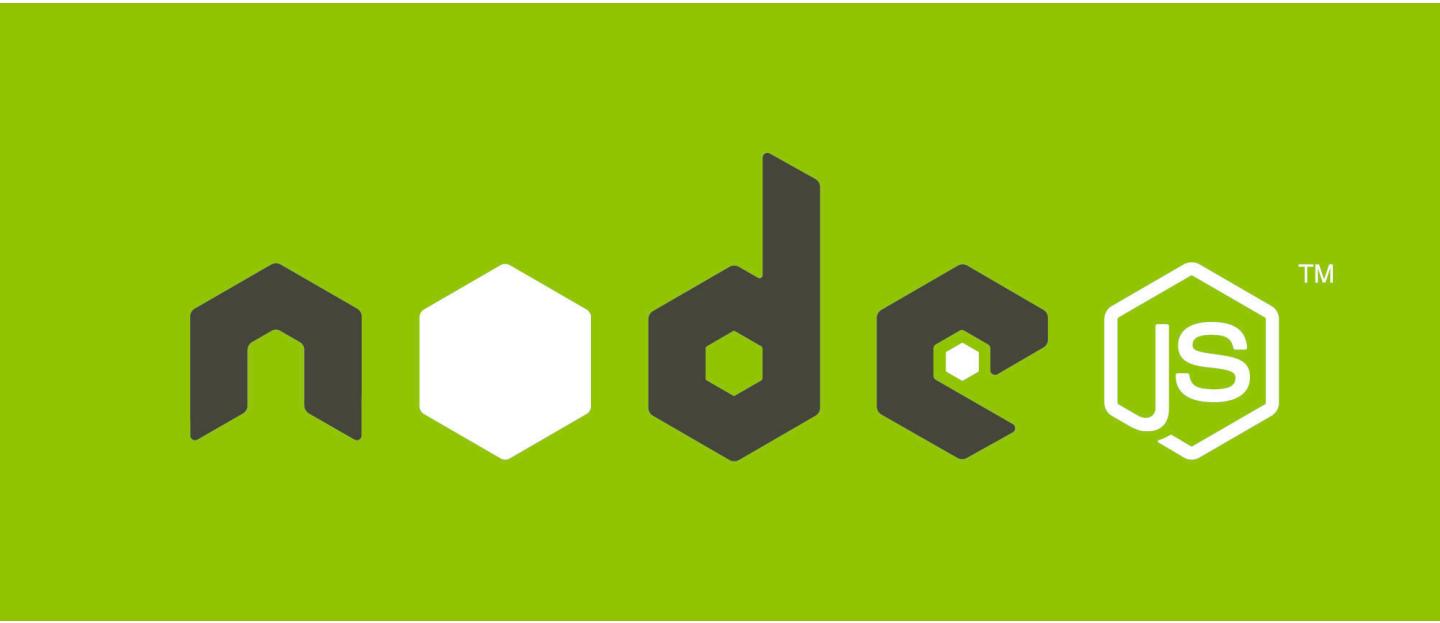
NodeJS

05

Presented by
Subhash EP



Subhash



Subhash

Node.JS Events

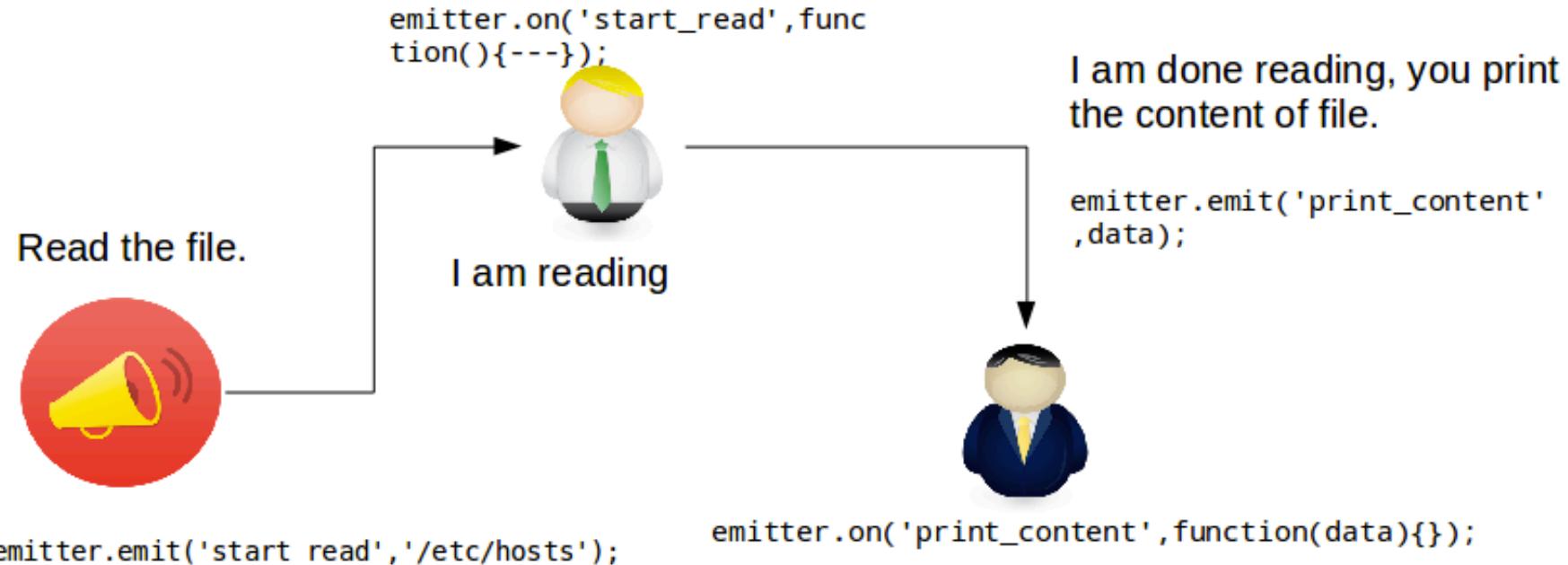
- Node.js comes with core events module.
 - `require('events')`
 - has one simple class "EventEmitter"

EventEmitter

- a class designed to make it easy to
 - emit events
 - subscribe to raised events

Multiple Subscribers

- built-in support for multiple subscribers



EventEmitter Demo

EventEmitter Memory Leaks

- memory leaks when
 - subscribing to events in a callback
 - but forgetting to unsubscribe at the end.
 - By default, EventEmitter will tolerate 10 listeners for each event type
 - anymore and it will print a warning to the console

Error Event

- An 'error' event is treated as a special exceptional case in Node.js.
- If there is no listener for it,
 - then the default action is
 - to print a stack trace and
 - exit the program

Creating Your Own Event Emitters

- call EventEmitter constructor
- use the util.inherits function to set up the prototype chain

Process Events

- A number of classes inside core Node.js inherit from EventEmitter.
- global process object is also an instance of EventEmitter

Global Exception Handler

- Any global unhandled exceptions can be intercepted by listening on the `uncaughtException` event on process.
- You should not resume execution outside this event handler
- The best strategy is to log the error for your convenience and exit the process with an error code

Exit

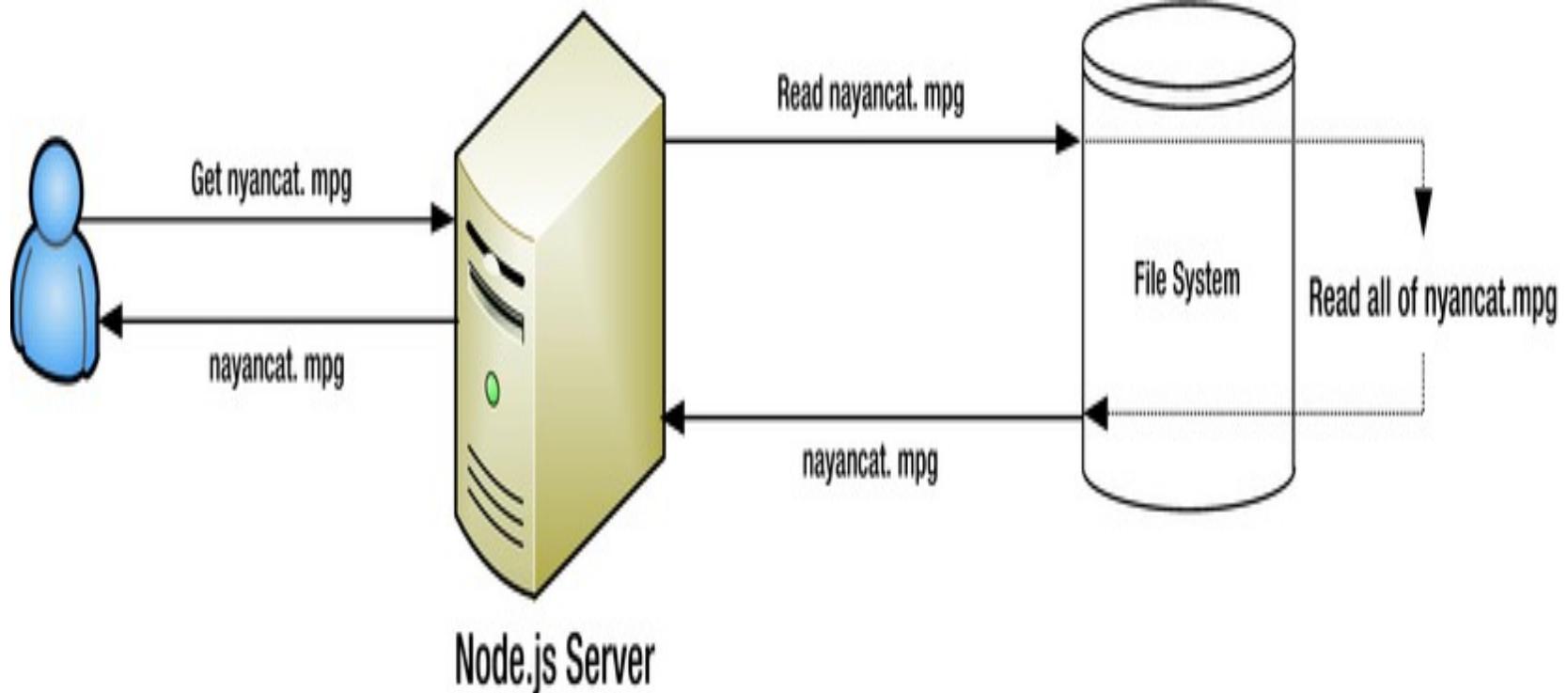
- The exit event is emitted when the process is about to exit.
- There is no way to abort exiting at this point.
- The event loop is already in teardown so you cannot do any async operations at this point

Signals

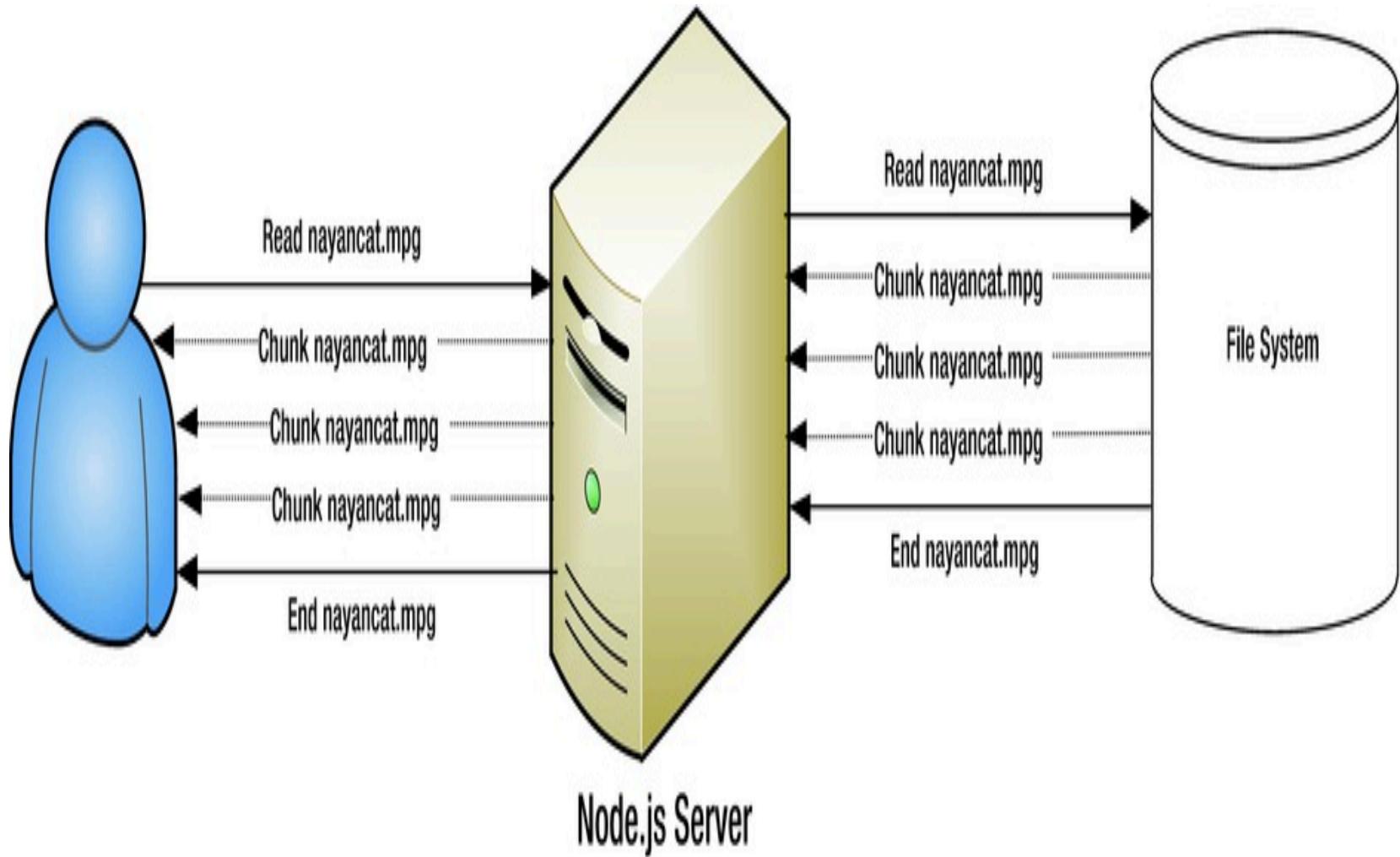
- Node.js process supports
 - UNIX concept of signals
 - a form of inter-process communication

Advanced EventEmitter Demo

Buffering



Streaming



Streams in Node.js

- Readable streams
- Writable streams
- Duplex streams
- Transform streams

Readable Stream

- A readable stream is one that you can read data from but not write to.
- A good example of this is `process.stdin`, which can be used to stream data from the standard input

Writable Stream

- A good example of this is `process.stdin`, which can be used to stream data from the standard input.
- A writable stream is one that you can write to but not read from.
- A good example is `process.stdout`, which can be used to stream data to the standard output

Duplex Streams

- A duplex stream is one that you can both read from and write to.
- A good example of this is the network socket.
- You can write data to the network socket as well as read data from it.

Transform Streams

- A transform stream is a special case of a duplex stream
 - output of the stream is computed from input.
- These are also called through streams.
- A good example of these is encryption and compression streams.

Pipe

- All the streams support a pipe operation that can be done using the pipe member function.
- This is one of the things that make streams in Node.js so awesome

Streams Demo

That's all



End of Session

©

subhash.ep@gmail.com
linkedin.com/in/subhashep