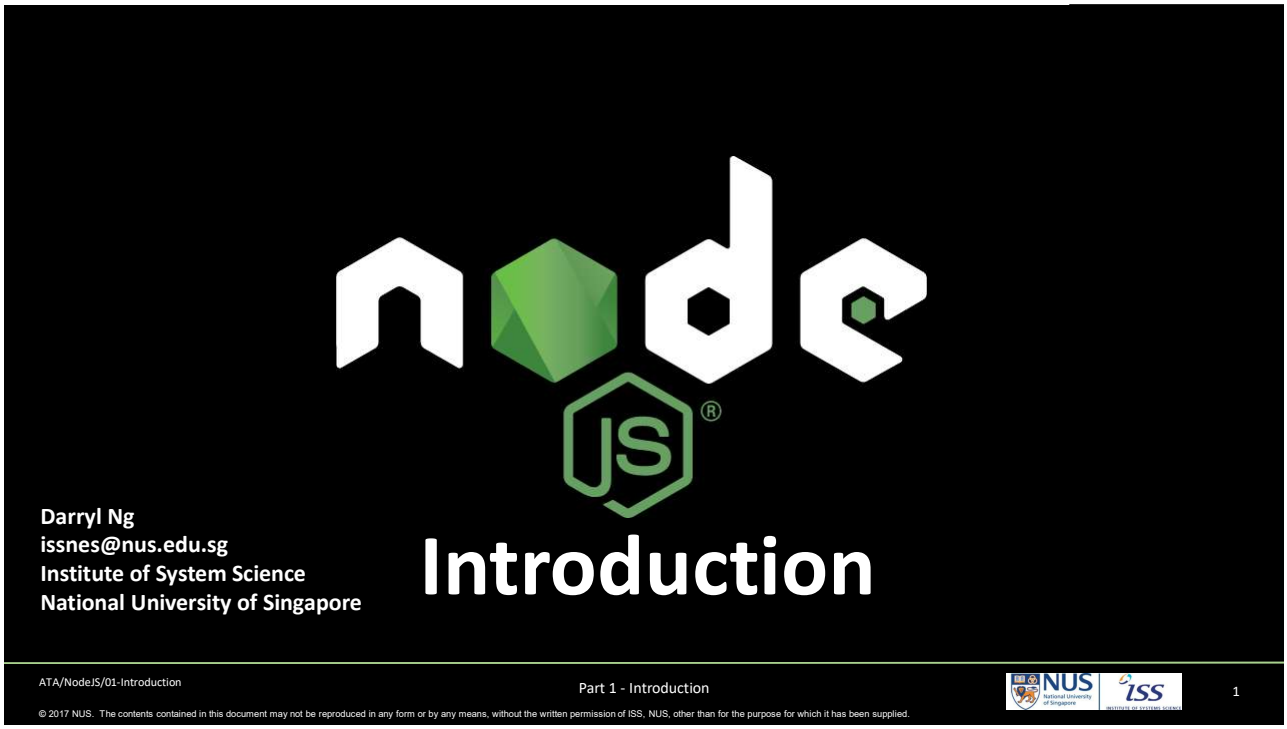


Darryl Ng
issnes@nus.edu.sg
Institute of System Science
National University of Singapore

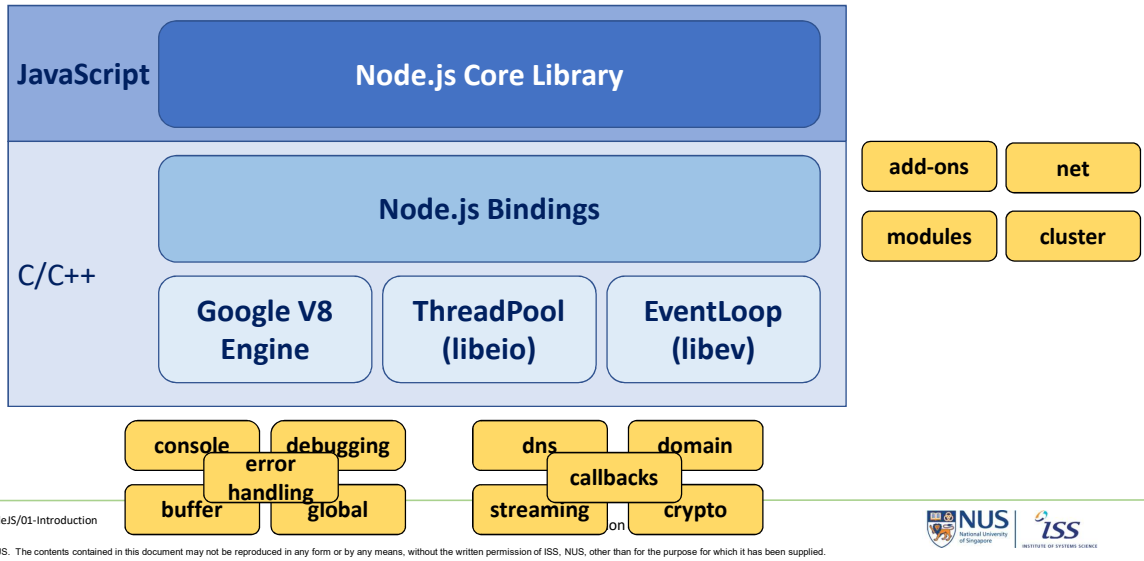
Intro

- JavaScript-based framework/platform
- Open source, cross platform
- Event-driven, lightweight
- I/O and data intensive web app
- Run on distributed devices

Key features

- Asynchronous and event driven
- Fast code execution
- Single threaded event looping
- No buffering, output chunk data

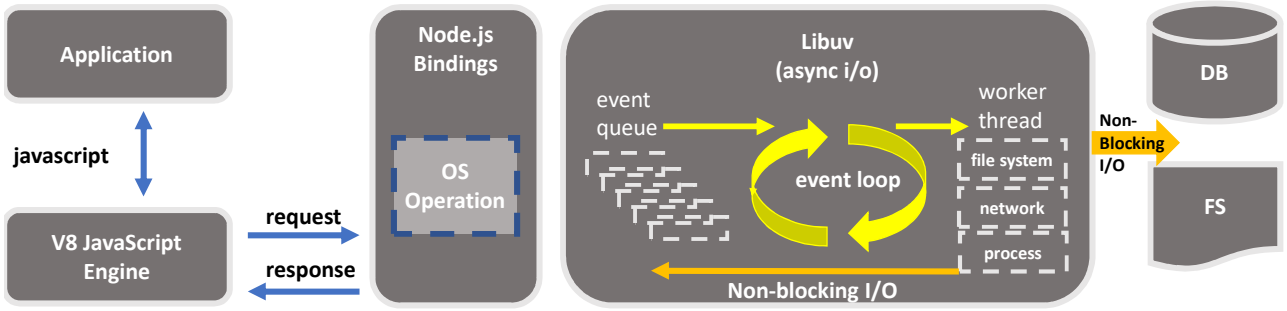
Architecture



Event driven

- single-threaded application
- support concurrency
- **event** and **callbacks**
- **async function calls**
- thread keeps an event loop
- fires the corresponding event which signals the event-listener function to execute when a task completes

Overarching System



Dos and Donts

✓I/O bound Applications
✓Data Streaming Applications
✓Data Intensive Real-time Applications (DIRT)
✓JSON APIs based Applications
✓Single Page Applications
× CPU intensive applications

http://stackoverflow.com/questions/5062614/how-to-decide-when-to-use-nodejs

https://www.quora.com/What-are-the-disadvantages-of-using-Node-js

REPL

command	description
ctrl + c	terminate the current command
ctrl + c twice	terminate the Node REPL
ctrl + d	terminate the Node REPL
Up/Down Keys	see command history and modify previous commands
tab Keys	list of current commands
.help	list of all commands
.break	exit from multiline expression
.clear	exit from multiline expression
.save *filename*	save the current Node REPL session to a file
.load *filename*	load file content in current Node REPL session

Useful Commands

command	description
npm - v	Check version of npm
npm version	List of installed common modules version
npm list	List of modules
npm list <package>	List specific module specified
npm install <package> --save	Install new node package module
npm init	Create new package.json file in your project root directory
npm update	Check and update all installed node packages
npm update <package>	Check and update specific install node package
npm link <package>	Link an installed package to your project in your project root directory
node -v	Check node version

Package Manager

- Online repositories for node.js packages/modules
- Command line utility
- Version management
- Dependency management

Package.json

attribute	description
name	package name
version	package version
description	package description
homepage	package homepage
author	package author
contributors	name of the contributors to the package
dependencies	list of dependencies. NPM automatically installs all the dependencies mentioned here in the node_module folder of the package
repository	repository type and URL of the package
main	entry point of the package
keywords	keywords

Common Modules

- express
  - Simpler html server, e.g. syntax, DB connection
- socket.io
  - Real-time apps
- nodemon
  - Monitor node.js and push changes automatically
- node-windows
  - Support for Windows services, event logging, UAC, and several helper methods for interacting with the OS
- node-mysql
  - Handles database operations, transactions and ORM for MySQL

Common Modules

- mssql
  - Handles database operations, transactions and ORM for MS SQL Server
- chartjs
  - For drawing charts, pies and graphs using nodejs
- recordrtc
  - Recording web audio and video stream
- stdio / prompt
  - Request/Prompt for input and output
- node-datetime
  - Manage date and time with formats

Key components

- Import required modules
  - Require directive
- Create server
  - Listen to request
- Read request and return response

Hello World! example

```
// import to include the required http module
var http = require('http');

// create a webserver
http.createServer(function (req, res) {

    // return respond to any incoming http request
    res.writeHead(200, {'Content-Type': 'text/plain'});
    res.end('Hello World\n');

}).listen(1337, '127.0.0.1');

// log what that we started listening on localhost:1337
console.log('Server running at 127.0.0.1:1337');
```


Express Hello World!

```
//
var express = require('express');

//
var app = express();

//
app.get('/', function(req, res){
    res.send('Hello World');
});

app.listen(1337);

//
console.log('Server running at 127.0.0.1:1337');
```

Http Response Codes

Code	Meaning	Description
200	OK	Everything went fine.
301	Moved Permanently	The requested URL has been moved, and the client should rerequest it at the URL specified in the response.
400	Bad Request	The format of the client's request is invalid and needs to be fixed.
401	Unauthorized	The client has asked for something it does not have permission to view.
403	Forbidden	The server is refusing to process this request. This is not the same as 401, where the client can try again with authentication.
404	Not Found	The client has asked for something that does not exist.
500	Internal Server Error	Something happened resulting in the server being unable to process the request.
503	Service Unavailable	This indicates some sort of runtime failure.