

## FSD - Node JS / Express JS



**BigLeap Solutions**  
*gain momentum*

<http://bigleap.co.in>

## Introduction

- Name
- Technical background
- Experience & Expectations
  - Experience with event driven systems



## Introduction

- Anilkumar GT
  - Co-Founder and Partner : BigLeap Solutions
  - Certified Ethical Hacker (EC-CEH)
  - Wildlife and Landscape Photographer:
    - <http://www.FleetingMoments.net>
- Youtube Channel:
  - Channel Name -> BigLeap Solutions
  - Videos on Design Thinking



## Node JS

- Platform to build scalable, high performant, networked applications
- JS runtime built on Chrome's V8 Engine
- Event driven, non-blocking I/O makes it lightweight and efficient
- Code in JavaScript
- Ryan Dahl - 2009
- Joyent maintained



## CPU Cost - I/O

L1	3 cycles
L2	14 cycles
RAM	250 cycles
Disk	41,000,000
Network	240,000,000

Source: Ryan Dahl's 2008.11.08 node.js presentation



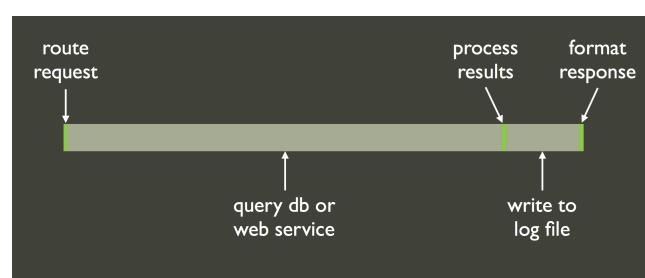
	L2	RAM	Disk	Network
L1	5	83	13,666,666	80,000,000
L2		18	2,928,571	17,142,857
RAM			164,000	960,000
Disk				6



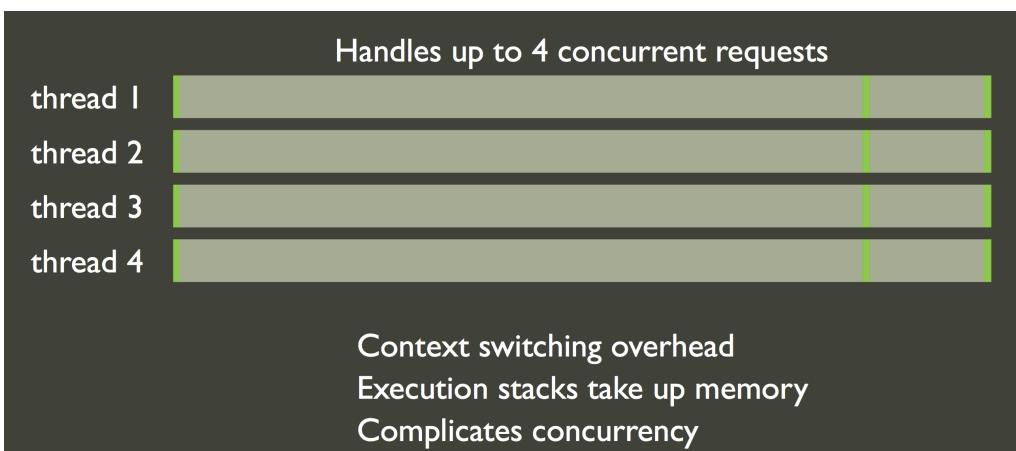
## Blocking

- Synchronous call
- Till the DB returns, this thread is blocked

```
$dbQuery = getUsers("Starks");
print $dbQuery->result();
print "Hello Users";
```



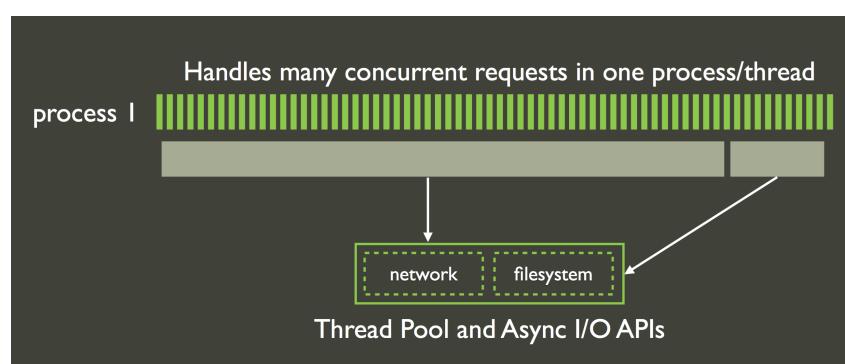
## Synchronous, Multi Threaded



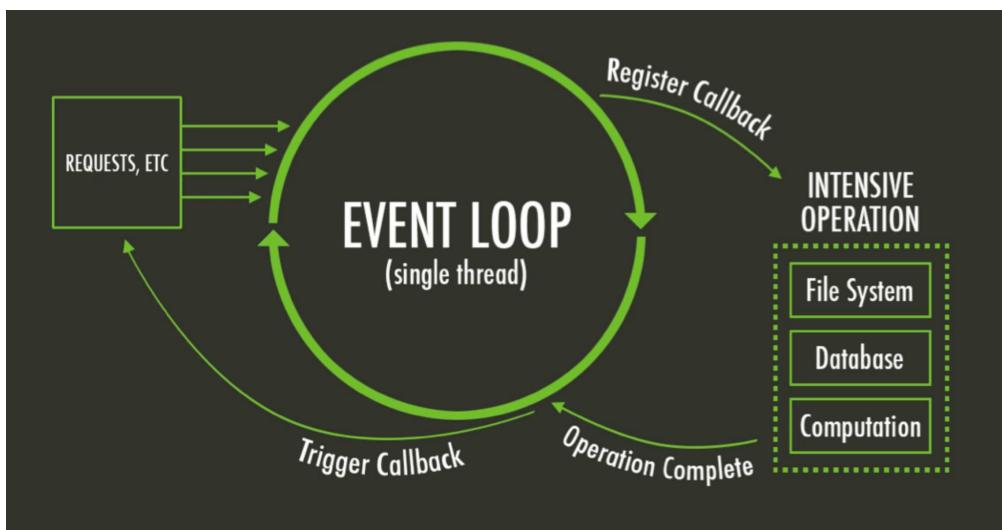
## Non-Blocking

- Asynchronous, Functional programming

```
var dbQuery = getUsers("Starks", function(err,result){
    console.log(result);
});
console.log "Hello Users";
```



## Event Loop



## Node JS

### Platform

node standard library

node bindings  
(http, socket, file system)

V8

thread pool  
(libeio)

event loop  
(libev)

crypto  
(OpenSSL)

DNS  
(c-ares)



# Serverless

- JEE / .Net / RoR
  - Server provided -> Web Server / Application Server
  - Developers write application code and 'deploy' to server
  - PL | BL | DAL
  
- NodeJS
  - Write the Server + Application code
  - Server + PL | BL | DAL



# Used By

- Walmart, eBay, Microsoft, Yahoo, LinkedIn, Paypal
  
- <http://nearform.com/nodecrunch/node-js-becoming-go-technology-enterprise/>

## High-Performance

PayPal reported: double the number of requests per-second and reduced response time by 35% or 200 milliseconds.

WalMart Labs had a bumper launch with Node.js in 2013, where they put all of their Mobile traffic through Node.js on black-friday, the busiest shopping period of the year.

The team at WalMart Labs live tweeted against #nodebf tag showing the performance of the Node.js application.

On Black Friday the WalMart servers didn't go over 1% CPU utilisation and the team did a deploy with 200,000,000 users online.



## JavaScript

- Scripting language for web
- Lightweight, interpreted language with first class functions
- JS is a prototype-based, multi-paradigm, dynamic scripting language.
  - Supporting object-oriented, imperative, and declarative (e.g. functional programming) styles
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript>



## Demo / Exercise

- Node REPL
- DataTypes
- Objects
- Exporting functions
- Passing arguments
- Reading files / dirs using fs
- Sync v/s Async
- Simple Web Server / Serving static HTML files



## NPM

- Node Package Manager
  - <http://www.npmjs.com>
- Installs, Publishes and Manage Node Programs
- Always use NPM ... do not manually copy / paste / move etc



## Node JS

- NodeJS Built-in modules
  - <https://nodejs.org/api/index.html>
- Can build applications using just these modules
- Low level abstractions, time consuming and may break DRY
- Need a higher level abstraction, that make application development easier, faster, better



## Express JS

- Fast, unopinionated, minimalist **web** framework for Node.js
  - <https://expressjs.com/>
- Minimal and flexible Node.js web application **framework**
- Robust set of features for web and mobile applications
- HTTP Middleware and utilities
- Performant, thin layer



## Demo / Exercise

- Install ExpressGen
- Customise generated app with Bootstrap:
  - Send email module



## Middleware

- Express app is essentially a series of middleware function calls
- Any one of the middleware functions could return a response to the request
- `next()` is used to denote work is done by this function

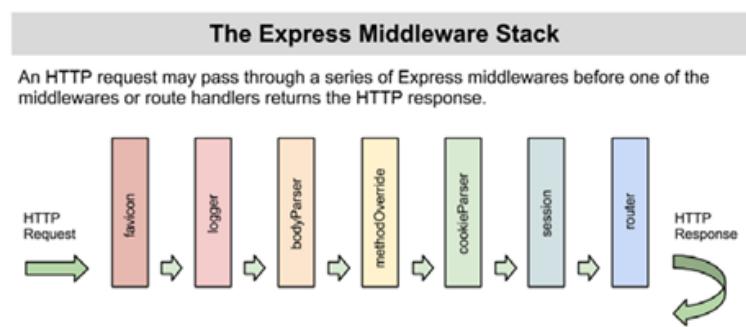


Figure 1



## Jade Templates

- Jade is an elegant templating language focused on enabling quick HTML coding
- Renamed as Pug
  - <http://pugjs.org>
- Default template engine for Node
- Node Routes redirect requests / actions to Jade templates



## Demo / Example

- Code login template
- Write user related routes
- Implement validation in the registration template



## Middleware

- Express app is essentially a series of middleware function calls
- Any middleware functions could return a response to the request
- `next()` is used to denote work is done by this function

### The Express Middleware Stack

An HTTP request may pass through a series of Express middlewares before one of the middlewares or route handlers returns the HTTP response.

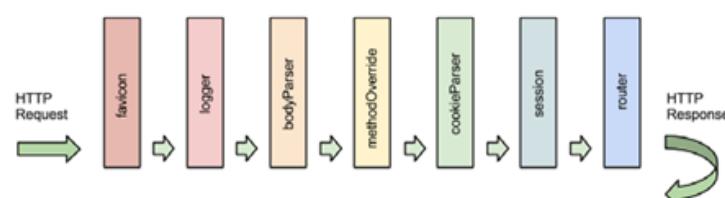


Figure 1



# Sessions

- Express JS Middleware
  - express-session
- Sessions and Cookies
- Can choose where to store the session data
- Default : memoryStore in local m/c memory
- Projects: Redis, Mongo DB



# Session

- **cookie**: Options for the session ID cookie. The default value is { path: '/', httpOnly: true, secure: false, maxAge: null }
- **genid**: Function to generate the session ID. Default is to use uuid
- **name**:The name of the session ID cookie to set / read in the response
- **proxy**: Trust the reverse proxy when setting secure cookies
- **resave**: If true forces a session to be saved back to store even if it was not modified in the request.
- **rolling**: Forces a cookie to be set on every request.
- **saveUninitialized**: If true it forces a newly created session without any modifications to be saved to the session store.
- **secret**: It is a required option and is used for signing the session ID cookie.
- **store**: Session store instance. Default is to use memory store.
- **unset**: Controls the handling of session object in the store after it is unset. Either delete or keep the session object. Default is to keep the session object



## Demo / Exercise

- Change app.js to add sessions and cookies



## NoSQL

- RDBMS
- Schema -> Tubular form / tables, columns
- Data stored in rows
- Vertically Scalable
  
- Variety, Velocity of web scale data challenges traditional storage mechanisms
- Non Relational / Non SQL
  
- Not Only SQL



# MongoDB

- Free and open source document based NoSQL store
- **Humongous**
- JSON-like documents with dynamic schemas (MongoDB calls the format BSON)
- Horizontally Scalable



# Flexible Schema

```
{
  "name": "John Doe", "email": "john@email.com", "phone": 987239402},
  {
    "name": "Kevin Flynn",
    "email": "kevin@encom.com",
    "phone": 987239402,
    "skypeID": "kevfln" ←
  },
  {
    "name": "Hugo Roberts",
    "email": "hugo@gmail.com",
    "phone": 987229193,
    "twitter": "hugoroberts" ←
  },
  {"name": "Ed Dillinger", "email": "ed@encom.com", "phone": 902839392}
```

```
{
  "name": "John Doe",
  "email": "john@email.com",
  "phone": 987239402,
  "fruitsHeLikes": ['Apples', 'Oranges', 'Kiwi'],
  "postalAddress": {
    "address": "901, RiverDale St.",
    "city": "RiverTown"
  },
  {
    "address": "Suite 403, Mongoplex, Street 20",
    "city": "RiverTown"
  }
}
```



# MongoDB

- Identifier : \_id
- Database -> Database
- Table -> Collection
- Row(s) -> Document(s)
- Columns -> Fields



## Demo / Exercise

- Setup MongoDB
- Try out various commands via CLI
- Create mongoNativeAccess.js and access MongoDB



## Mongoose

- Elegant Mongo object modeling for NodeJS
  - <http://mongoosejs.org>
- Writing MongoDB validations, casting, callbacks and boilerplates is no fun
- Schemas -> Collection Structure
- Schema is converted to a Model
- Instance of a Model is a document



## Demo / Exercise

- Create a User Schema
- Add persistence of a new user to the registration flow



## Passport JS

- Simple, unobtrusive authentication for Node.js
  - <http://passportjs.org>
- Express middleware configuration and routes integration
- 300+ strategies
- Local Strategy



## Demo / Exercise

- Add Login and Logout functionality



## Rest API

- Express JS makes it very easy to create Rest API as well
  - Consumed by Web Apps / Mobile Apps
- **R**Epresentational **S**tate **T**ransfer
- Architectural style to build web services without SOAP



## REST

- Resource centric approach
  - Nouns are separated from the Verbs
  - Apply the standard HTTP Verbs to any noun
  - Programming model becomes simpler

Method	CRUD Operation	Description
POST	Create	Create a new Resource
GET	Retrieve	Retrieve a representation of a Resource
PUT	Update	Update a Resource
DELETE	Delete	Delete a Resource



## Demo / Exercise

- Add a new Rest API for books



## Mocha / Chai

- Mocha - feature rich JS unit testing framework
- <http://mochajs.org>
- BDD style tests / specs
- Can test synchronous / asynchronous code bases
- Hooks: before, after, before each, after each
- Chai provides the assertion library
- Mocha works well with Chai



# Chai

- Chai provides TDD / BDD style assertion libraries
- <http://chaijs.com>



## Should

```
chai.should();

foo.should.be.a('string');
foo.should.equal('bar');
foo.should.have.length(3);
tea.should.have.property('flavors')
  .with.length(3);
```

[Visit Should Guide](#) ↗

## Expect

```
var expect = chai.expect;

expect(foo).to.be.a('string');
expect(foo).to.equal('bar');
expect(foo).to.have.length(3);
expect(tea).to.have.property('flavors')
  .with.length(3);
```

[Visit Expect Guide](#) ↗

## Assert

```
var assert = chai.assert;

assert.typeOf(foo, 'string');
assert.equal(foo, 'bar');
assert.lengthOf(foo, 3)
assert.property(tea, 'flavors');
assert.lengthOf(tea.flavors, 3);
```

[Visit Assert Guide](#) ↗



# Demo / Exercise

- Write unit tests for the Books Rest API



# Gulp

- Automate and enhance Node JS workflow
- <http://gulpjs.com>
- Grunt came earlier .... uses intermediate files
- Gulp uses streaming
- Plugin based
- Think of a task and you will have a grunt plugin for that

*Gulp*



## Demo / Exercise

- Install gulp globally and locally as well....



Thank You !!!



**BigLeap Solutions**  
*gain momentum*

<http://bigleap.co.in>