

Javascript By Pankaj kumar agarwal

Contact : 7348820668

Whatsapp : 7348820668

Email : pankaj.7613@gmail.com

Blog : <http://pankajagarwal.in/>

Git Repository : <https://github.com/seecog>

Facebook : <https://www.facebook.com/Pankaj.Spybird>

NodeJs Syllabus

Introduction and first steps :

Introduction

Node Architecture (**Important**)

History of nodeJs

What is Node.js

Installing NODE.js

NodeJs module system

File system and command line argument

Core Fundamentals

IDE's recommended to use

The Global object

The Process object

Readline

Custom Events

Module

Exporting

Debugging NodeJs

Debugging Node.js

Error Messages

Asynchronous NodeJs

Asynchronous Basics

Call Stack, Callback Queue, and Event Loop

Making HTTP Requests

Customizing HTTP Requests

An HTTP Request Challenge

Handling Errors

The Callback Function

Callback Abstraction

Callback Abstraction Challenge

Callback Chaining

ES6 Aside: Object Property Shorthand and

Destructuring

Destructuring and Property Shorthand Challenge

HTTP Requests Without a Library

Adding templating engine

Working with PUG

Working with Handlebars

MongoDb

All queries

MongoDb Cloud (MLAB)

Create sample database

Create user

Connect with nodejs

Section Intro: REST APIs and Mongoose

Setting up Mongoose

Creating a Mongoose Model

Data Validation and Sanitization: Part I

Data Validation and Sanitization: Part II

Structuring a REST API

Installing Postman

Resource Creation Endpoints: Part I

Resource Creation Endpoints: Part II

Resource Reading Endpoints: Part I

Resource Reading Endpoints: Part II

Promise Chaining

Promise Chaining Challenge

Async/Await

Async/Await: Part II

Integrating Async/Await

Resource Updating Endpoints: Part I

Resource Updating Endpoints: Part II

Resource Deleting Endpoints

Separate Route Files

Mysql queries

All important queries

Nodejs with mysql

Comparing SQL and NoSQL

Setting Up MySQL

Connecting our App to the SQL Database

Basic SQL & Creating a Table

Retrieving Data

Fetching Products

Fetching Products - Time to Practice

Inserting Data Into the Database

Fetching a Single Product

Introducing sequelize

Module Introduction

What is Sequelize?

Connecting to the Database

Defining a Model

Syncing JS Definitions to the Database

Inserting Data & Creating a Product

findById() in Sequelize 5

Retrieving Data & Finding Products

Getting a Single Product with the "where" Condition

Fetching Admin Products

Updating Products

Deleting Products

Creating a User Model

Adding a One-To-Many Relationship

Creating & Managing a Dummy User

Using Magic Association Methods

Fetching Related Products

One-To-Many & Many-To-Many Relations

Creating & Fetching a Cart

Adding New Products to the Cart

Adding Existing Products & Retrieving Cart Items

Deleting Related Items & Deleting Cart Products

Adding an Order Model

Storing Cartitems as Orderitems

Resetting the Cart & Fetching and Outputting Orders

The Express Framework

The Intro

Routes

Route Parameters

Middlewares

Submitting Post Data with Express

Promise in nodeJs

Create Promise in nodejs

Promise.all

Async await

Closures in nodejs

Node JS Cluster Programming

Create an express application with help of cluster programming. It will create sub process

Multer file operation

Install multer through npm

Upload the file with help of multer module

Async Module(Parallel,series,waterfall) / Await

Give example for async.series()

Give example of passing and object also

async.parallel()

async.waterfall()

JWT authentication ,Hashing

Create a small login application with angular6 + nodejs jsonwebtoken. Create as well as verify the token

Microservices in NodeJs

NodeJs with redis

Websockets - Small Chat App

Intro and Connection

Sending Form Data

Broadcasting

Client Side Data Data Handling

Communication with all clients

Server sent event

Create a live data through SSE

Sending email in NodeJs

Exploring SendGrid

Sending Welcome and Cancellation Emails

Environment Variables

Creating a Production MongoDB Database

AWS Deployment

Session and cookies

What is a Cookie?

Optional: Creating the Login Form

Adding the Request Driven Login Solution

Setting a Cookie

Manipulating Cookies

Configuring Cookies

What is a Session?

Initializing the Session Middleware

Using the Session Middleware

Using MongoDB to Store Sessions

Sessions & Cookies

Deleting a Cookie

Adding payments (Stripe)

GraphQL

What is GraphQL?

Understanding the Setup & Writing our First Query

Defining a Mutation Schema

Adding a Mutation Resolver & GraphQL
Adding Input Validation
Handling Errors
Connecting the Frontend to the GraphQL API
Adding a Login Query & a Resolver
Adding Login Functionality
Adding a Create Post Mutation
Extracting User Data From the Auth Token
Sending the "Create Post" Query
Fixing a Bug & Adding New Posts Correctly
Adding a "Get Post" Query & Resolver
Sending "Create Post" and "Get Post" Queries
Adding Pagination
Uploading Images
Fetching the imageUrl
Viewing a Single Post
Updating Posts
Deleting Posts
Adding Posts & Pagination
Managing the User Status
Using Variables
Fixing a Pagination Bug

Testing your app

Mocha & Basic testing
Watch & auto restart
Using assertion library
Testing asynchronous Code

Load Balancing in NodeJs

Pm2

NgInx

Docker

All basic commands for Docker

Deployment with AWS

Project in NodeJs application

Setting up Node.js

Introduction

Setting up and Installing Dependencies

Adding all the middle wares

Running your first server

Setting up MLab

Setting up Configuration file for database, port number and secret key

Adding CORS

Setting up Angular 5

Overview

Installing Dependencies – Angular

Adding Dependencies to App Module

Setting up Navbar and Footer

Setting up Navbar and Footer Episode

Angular Router Explanation

Setting up Home Page

Setting up REST Service

Setting up Data Service, Message Component

Authentication - Node.js

Overview

Create your first UserSchema model Part 1: Why

UserSchema model Part 2: Attributes

UserSchema model Part 3: Hash Password

UserSchema model Part 4: Compare Password
UserSchema model Part 5: Gravatar
UserSchema model Part 6: Conclusion
Create a signup API1
Create a login API

Authentication - Angular.js

Overview, JWT on the frontend, localStorage
Create Registration Component, Logic
Registration Page Design
Create Login Component, Logic
Login Page Design
Auth Guard
Features Demo

Profile and Address API - Node.js

Overview
Create A JSONWEBTOKEN middleware
Profile API - GET
Profile API - Editing your profile

Profile and Address Pages - Angular

Get Profile Logic
Profile Page
Settings Page
Create Address Page
Features Demo

Adding a category and product as a seller - Node.js

Overview
Create a category Schema
Create a category Schema
Get all categories API
Create a Product Schema
Create amazon web services account
Install dependencies
Create an upload function using s3 and multer
Add a product API - Seller
Get my Product API

Using faker library to increase the amount of products stored

Adding a category and product as a seller - Angular

Create Categories Component

Add Category (Under Categories)

Create Post Product Component

Create My-Products Component

Products API - Node.js

Getting all products in a specific category

Pagination concept

Async.waterfall, params and query

Getting all products in a specific category API Part 1 - Counting

Getting all products in specific category API Part 2 - Pagination

Getting all products in a specific category API Part 3 - Getting category name

Getting all products in a specific category API Part 4 - Using async.parallel

Getting a single product API

Getting all products API

Products API - Angular

Pagination concept

Create Category Component

Create Product Component

Add Product Cards to Home

Review API - Node.js

Create a Review Schema

Create a Review API - POST

Install deepPopulate

Calculate average rating using virtuals on Product Schema

Review feature - Angular

Post a review

Search API with Algolia - Node.js

Create Algolia account

Add Algolia plugin to Product Schema
Create a search API

Search API with Algolia - Angular

Search Component

Correction

Payment API with Stripe - Node.js

Create a Stripe Account

Install dependency

Order Schema

Payment API

Payment API with Stripe - Angular

Setting up Stripe

Cart Functionality

Cart Component

Order API - Node.js

Orders API

Single Order API