**Annotated Bibliography**

Bhandari, R. (2020). *Rubiin/nestjs-easyconfig* [TypeScript]. https://github.com/rubiin/nestjs-easyconfig (Original work published 2019)

This article describes how to use the nestjs-easyconfig tool to load config files that have been written in the dotenv format. It also describes how to create a default config file that demonstrates which values should be included in each environment config file. This is important as it allows the default file to remain in source control without exposing sensitive data.

*Documentation | NestJS - A progressive Node.js framework*. (n.d.). Documentation | NestJS - A Progressive Node.Js Framework. Retrieved May 23, 2020, from <https://docs.nestjs.com>

This documentation describes how to use the NestJS framework. It details the main features of NestJS and as well as fundamental techniques and patterns. The documentation also discusses the NestJS CLI and how to use it to speed up development.

*Home · OrientDB Manual*. (n.d.). Retrieved May 23, 2020, from <https://orientdb.com/docs/3.0.x/>

This manual describes the basic principles behind OrientDB. It also details OrientDB’s features, functionality, and provides some examples of its usage. However, it is not exhaustive in its descriptions and more advanced uses of OrientDB are not included.

Lasorsa, Y. (2020, July 8). *Build your first serverless app with Angular, NestJS and Azure—DEV*. <https://dev.to/azure/build-your-first-serverless-app-with-angular-nestjs-and-azure-108h>

This article goes through all the steps to bootstrap, build, and deploy a complete application using Angular, NestJS, and Azure serverless platform. It includes a small sample application and details each step including the installation of the necessary technologies.

Maislos, A. (2017, May 30). *Hybrid Databases: Combining Relational and NoSQL -*. <https://www.stratoscale.com/blog/dbaas/hybrid-databases-combining-relational-nosql/>

This article discusses the strengths and weaknesses of Relational and NoSQL databases and how they two technologies are used together in Hybrid databases. It also goes over three use cases for Hybrid databases and discusses scaling. Lastly, the article compares a few popular Hybrid databases.

Morony, J. (2019, March 7). *Creating a Simple Live Chat Server with NestJS and WebSockets*. Joshmorony - Learn Ionic & Build Mobile Apps with Web Tech. <https://www.joshmorony.com/creating-a-simple-live-chat-server-with-nestjs-websockets/>

This tutorial is on how to use WebSockets. It teaches the technology through the creation of a simple chat application using NestJS and Angular. The application broadcasts messages to any listening client and notifies when a client connects or disconnects.

*Orientechnologies/orientjs*. (2020). [JavaScript]. OrientDB. https://github.com/orientechnologies/orientjs (Original work published 2014)

This documentation describes the use of the official orientdb driver. It provides installation steps and how to make simple queries. The documentation only covers the most basic use cases but it does provide some example projects.

Pavithra, D., & Balakrishnan, R. (2015). IoT based monitoring and control system for home automation. *2015 Global Conference on Communication Technologies (GCCT)*, 169–173. <https://doi.org/10.1109/GCCT.2015.7342646>

The project proposes an efficient implementation for IoT (Internet of Things) used for monitoring and controlling the home appliances via World Wide Web. Home automation system uses the portable devices as a user interface. They can communicate with home automation network through an Internet gateway, by means of low power communication protocols like Zigbee, Wi-Fi etc. This project aims at controlling home appliances via Smartphone using Wi-Fi as communication protocol and raspberry pi as server system. The user here will move directly with the system through a web-based interface over the web, whereas home appliances like lights, fan and door lock are remotely controlled through easy website. An extra feature that enhances the facet of protection from fireplace accidents is its capability of sleuthing the smoke in order that within the event of any fireplace, associates an alerting message and an image is sent to Smartphone. The server will be interfaced with relay hardware circuits that control the appliances running at home. The communication with server allows the user to select the appropriate device. The communication with server permits the user to pick out the acceptable device. The server communicates with the corresponding relays. If the web affiliation is down or the server isn't up, the embedded system board still will manage and operate the appliances domestically. By this we provide a climbable and price effective Home Automation system.

Real-time Websocket Connection between Sensor Reading ESP8266 and Node.JS Server. (2018, July 6). *ESP8266 Shop*. <https://esp8266-shop.com/blog/websocket-connection-between-esp8266-and-node-js-server/>

In this Project we will establish a Websocket connection between multiple ESP8266 and a local Node.JS server. The Node.JS server will be running on a PC, laptop or a on Raspberry Pi, while we use C/C++ code on the Arduino IDE for the ESP8266. Multiple clients (browser, ESP8266) can connect to this Node.Js Websocket server at the same time. Each ESP8266 will read random ADC values and send these every 300 ms to the Node.Js server that will print them on the console and broadcast them to all other connected clients (similar to a group chat). The sender ESP8266 from which the particular ADC reading originates, will receive an edited message, which can be printed in a serial monitor.

*Repository Pattern | DevIQ*. (n.d.). Retrieved May 27, 2020, from <https://deviq.com/repository-pattern/>

The Repository Pattern has gained quite a bit of popularity since it was first introduced as a part of Domain-Driven Design in 2004. Essentially, it provides an abstraction of data, so that your application can work with a simple abstraction that has an interface approximating that of a collection. Adding, removing, updating, and selecting items from this collection is done through a series of straightforward methods, without the need to deal with database concerns like connections, commands, cursors, or readers. Using this pattern can help achieve loose coupling and can keep domain objects persistence ignorant. Although the pattern is very popular (or perhaps because of this), it is also frequently misunderstood and misused. There are many different ways to implement the Repository pattern. Let’s consider a few of them, and their merits and drawbacks.

Sanatan, M. (n.d.). *Managing Environment Variables in Node.js with dotenv*. Stack Abuse. Retrieved August 21, 2020, from <https://stackabuse.com/managing-environment-variables-in-node-js-with-dotenv/>

Deploying an application requires developers to put thought and consideration into how it is configured. Many apps are deployed in a development environment before being deployed to the production environment. We need to ensure each environment is configured correctly, it could be disastrous if our production application was using our development database, for example.

Environment variables allow us to manage the configuration of our applications separate from our codebase. Separating configurations make it easier for our application to be deployed in different environments.

Large applications tend to have many environment variables. To better manage them we can use the dotenv library, which allows us to load environment variables from a file

Wanago, M. (2020, May 25). API with NestJS #3. Authenticating users with bcrypt, Passport, JWT, and cookies. *Marcin Wanago Blog - JavaScript, Both Frontend and Backend*. <https://wanago.io/2020/05/25/api-nestjs-authenticating-users-bcrypt-passport-jwt-cookies>

Authentication is a crucial part of almost every web application. There are many ways to approach it, and we’ve handled it manually in our TypeScript Express series. This time we look into the passport, which is the most popular Node.js authentication library. We also register users and make their passwords secure by hashing.

Witalec, S. (2018, August 23). *Apps That Work Natively on the Web and Mobile—Angular Blog*. <https://blog.angular.io/apps-that-work-natively-on-the-web-and-mobile-9b26852495e7>

This article covers the core functionality around sharing code between the web and mobile apps with Angular and NativeScript. It provides instructions on how to create a simple Hello World app that can deploy to the web and on mobile devices.