



Developing Back-End Apps with Node.js and Express

Module 1 Glossary: Introduction to Server-Side JavaScript

| Term | Definition |
|---------------------|--|
| Anonymous Function | A function that is not named. An anonymous function is often passed into another function as a parameter. |
| Application Server | Transforms data into dynamic content and runs the business logic, which is the data storage and transfer rules. |
| Asynchronous | A process that runs independently of other processes. |
| Callback Function | A function passed into another function as a parameter, which is then invoked inside the outer function to complete an action. Instead of blocking on asynchronous I/O operations, callback functions are used to handle results when the operations complete. |
| Database Server | A server dedicated to providing database services. |
| Dependencies | Code, usually in the form of libraries and packages, that are called from other modules and reused in a program. |
| Event-Driven | Where the flow of a program is determined by particular events such as user input. |
| Express.js | A highly configurable web framework for building applications on Node.js. |
| Framework | Generates code that cannot be altered to perform common tasks. Examples include Django, Ruby on Rails, and Express.js. |
| HTTP Server | A type of software-based server that understands URLs and hypertext transfer protocol. |
| Load | Refers to the number of concurrent users, the number of transactions, and the amount of data transferred back and forth between the clients and servers. |
| Module | Files containing related, encapsulated JavaScript code that serve a specific purpose. |
| Multi-Threaded | Where multiple tasks are executed simultaneously. |
| Node.js | A JavaScript runtime environment that runs on Google Chrome's V8 engine. |
| Non-Blocking | Failure of a given thread does not cause failure in another, and the execution of a task is not blocked until execution of another task is completed. |
| Npm | Stands for node package manager. It is the default package manager for the Node.js runtime environment. |
| Package | A directory with one or more modules bundled together. |
| Package.json | Contains metadata information about the project, including dependencies and scripts. |
| Payload | The data transmitted between client and server. |
| Runtime Environment | Behaves similarly to a mini operating system that provides the resources necessary for an application to run. It is the infrastructure that supports the execution of a codebase. It is the hardware and software environment in which an application gets executed. Node.js is an example of a backend runtime environment. |
| Scalability | The application's ability to dynamically handle the load as is or shrinks without it affecting the application's performance. |
| Server.js | A file that contains the code that handles server creation. |
| Single-Threaded | Where only one command is processed at a given point of time. |
| Web Server | Ensures client requests are responded to, often using HTTP. |
| Web Service | A type of web API that communicates using HTTP requests. It is the web service in the programming interface that sends and receives requests using HTTP among web servers and the client. |

Changelog

| Date | Version | Changed by | Change Description |
|------------|---------|-----------------|--|
| 28-10-2022 | 1.0 | Sapthashree K S | Initial version created |
| 25-11-2022 | 1.1 | K Sundararajan | IDSN logo added based on Beta testing feedback |
| 29-11-2022 | 1.2 | K Sundararajan | Title updated based on Beta testing feedback |