Interview Questions

Modules and Commands

Q1. What Is NodeJS Module? **(UPLERS)**

Answer:

- Modules in Node.js are collections of files or folders that contain various chunks of code encapsulated into them.
- Modules enhance reusability and help to develop complex code with ease.
- Modules are simple or complex functions organised into single or multiple **.js** files that can be used reused within a Node.js Application.
- Modules can be considered as JavaScript Libraries.
- Modules are a bunch of functions that a developer can reuse in an application without starting from scratch.
- Modules help in breaking down complex codes into smaller segments which helps in debugging.
- Modules help in structuring the code base, and we can combine various modules into our application to get desired output.

Q2. What are Modules? Explain its types with examples. **(ARKA SOFTWARES)**

Answer:

- When a **function/ code, simple or complex,** is written down into a **.js** file, many such files combine into a folder to enhance the reusability of the code and increase its readability. Such files/folders, when grouped, make a **module**.

- A Module can be the same as JavaScript Libraries. Modules help break down large chunks of reusable code into smaller fragments, files, and folders that can be used across multiple segments in the application.

- There are three types of Modules, i.e. Core/ Built-in Modules, Local Modules, External/ Third-Party Modules.

- **Core Modules** have basic functionality and come along with Node.js, and there is no need to download them. E.g. **http, url, path, os, fs.**

- Local Modules are coded locally in a Node.js Application. Unlike Built-in/ Core modules or External/ Third-Party modules, these modules are built locally, specifically for the project. These modules include the various functionalities required in the project or various chunks of needed reusable code in multiple files and sections of the code.

- External modules are available online and installed using the Node Package Manager **(npm)**. These modules are read to use and coded by the developer community of Node.js and make the development process hassle-free and quick. E.g. **express, react, nodemon, mongoose, angular.**

Q3. What are Local Modules? How to create, export and import these modules? **(UBER)**

Answer:

- **Local Modules** are coded locally in a Node.js Application.

- Unlike Built-in/ Core modules or External/ Third-Party modules, these modules are built locally, specifically for the project.

- These modules include the various functionalities required in the project or various chunks of needed reusable code in multiple files and sections of the code.

- These modules can be packaged and published on npm for **Node.js, and the npm community** can use them.

- **Creating Local Modules:**
    - Create a new **.js** file and Code the Desired Functionality in the form of a function expression.
    - The Functionality can be as simple as a function that returns the factorial of a number.
    - To use this chunk of code outside this module/file, we'll have to export the functionality and then import the functionality wherever needed.
    - **module.exports** is the keyword used to let the compiler/system know that these functionalities can be outside that particular module/file.
    - We can simply use ~~module.~~**exports** as 'module' is now redundant with the latest updates.
    - To export a particular module or functionality, we need to replace the function's name in the function expression with **module.exports.func_name** or simply **exports.func_name**.
    - To import a module, we need to use the **require** keyword and collect the module into a variable. Generally, it is collected into a constant.
    - **const var_name = require('path')**