

# **BACKEND**

- 1) [.ENV \(Environment Variables\)](#)
- 2) [Serve Files From Server](#)

# ENV (Environment Variables)

The .env file is a simple text file that stores environment variables for an application. It's commonly used to store sensitive configuration settings, such as API keys, database credentials, and other values that might vary across different environments (development, staging, production). By keeping these settings in a separate file, they can be managed more easily and securely, and the code itself remains cleaner.

1. **Key-Value Pairs:** The file contains key-value pairs, where the key is the name of the environment variable, and the value is its corresponding value.
2. **Security:** .env files are typically excluded from version control systems (e.g., Git) to prevent accidental exposure of sensitive information.
3. **Flexibility:** They allow for easy adaptation to different environments by simply modifying the values in the .env file.

Ex :

```
1. DB_HOST="localhost"
2. DB_USER="myuser"
3. DB_PASSWORD="mypassword"
4. API Keys: API_KEY="your_api_key"
```

## Nodejs Implementation :

1. npm i dotenv
2. in the main server.js file (where the port is running), import dotenv package `require('dotenv').config();` This command will help access the env variables in all other files because all files are run through server.js
3. Now wherever you want to use the variables defined in the .env file, just use `process.env.VARIABLE`

Note : Sometimes, the file doesn't recognize what process.env.VARIABLE is, for that import the dotenv package in that specific file again

## What is process ?

The process object is a Node.js object. It provides information about, and control over, the current Node.js process. It is a global object, meaning it is always available in Node.js applications without requiring an import statement. While Node.js uses JavaScript as its language, the process object itself is not a standard JavaScript object available in browser environments. It's specific to the Node.js runtime environment.

more about process object : [https://github.com/69JINX/FrontEnd/blob/main/Notes/Notes\\_Pic/dotenv-and-global-access.png](https://github.com/69JINX/FrontEnd/blob/main/Notes/Notes_Pic/dotenv-and-global-access.png)

# Serve Files (Assets) From Server

[https://github.com/69JINX/FrontEnd/blob/main/Notes/Notes\\_Pic/serve-files-from-server.png](https://github.com/69JINX/FrontEnd/blob/main/Notes/Notes_Pic/serve-files-from-server.png)

By default, no one can access files on a server just by using a link to the server. To allow public access to files stored on the server, we can use middleware provided by Express that serves static files via their file paths.

(Note: In real-world applications, files are often stored on cloud platforms like AWS S3 instead of directly on the server.)

Imagine the following file structure on the server:

```
└─ Public/
    └─ image.jpg
```

To give public access to `image.jpg` via a URL, we can use the following Express middleware:

```
1. app.use('/files', express.static(path.join(process.cwd(), 'Public')))
```

Now, image.jpg will be accessible through this URL: <https://your-domain.com/files/image.jpg>

