

ENVIRONMENT VARIABLES IN NODE.JS

WE WILL COVER THE FOLLOWING TOPICS:

- process.env
- Why should we use environment variables?
- How can process.env help us with private data?
- The dotenv npm package
- Updating .gitignore

Environment variables are variables that are "global" to the environment where the code is running

Being scoped to the environment means they are available anywhere in the code

It does not need to be imported, nor do you need to create a reference to it

PROCESS.ENV

In **node.js**, we have access to a global variable called `process.env`

```
console.log(process.env);
```

Yes, it's that easy!

`process.env` is a global variable which is set at runtime. Our app can read from this variable.

Many cloud hosts like Heroku, Azure, AWS, now.sh, etc. use node environment variables

Task: Try running this code in a node environment

```
console.log(process.env);
```

What do you see?

You will have seen a list of variables dumped onto the page
This is because `process.env` is an **object**
And it already comes with a set of predefined keys / values

WHY SHOULD WE USE ENVIRONMENT VARIABLES?

This is not good! 😬

APP.JS

```
const data = fetch('https://somewebsite.com/getData?  
secret_api_key=1234567xxx');
```

Question: Why is this not good?

Answer:

- A lot of work to change these values if we use them in multiple places
- Also, we do not want to store private data, secret codes, API keys, passwords in our git repository

HOW CAN PROCESS.ENV HELP US WITH PRIVATE DATA?

We can add our own keys and values to process .env, just like we do with regular objects

If we can add values to this process .env variable, we might have a way to keep values "secret" from our git repository

APP.JS

```
process.env.SECRET_API_KEY = "1234567xxx";  
console.log(process.env.SECRET_API_KEY); // "1234567xxx"  
  
const data = fetch(`https://somewebsite.com/getData?  
secret_api_key=${process.env.SECRET_API_KEY}`);
```

- We need a better solution
- We will store the secret key in a separate file
- `process .env` will help us (but it is not the final solution)

THE DOTENV NPM PACKAGE

Developers saw this problem, and being developers, developed a solution

The most popular of these is dot env

```
npm install dotenv
```

dotenv allows us to store our secret keys in a separate .env file
It takes our keys and attaches them to the global process .env
variable

.ENV

```
SECRET_API_KEY = 1234567xxx;
```

APP.JS

```
const dotenv = require('dotenv');  
  
dotenv.config();  
  
const data = fetch(`https://somewebsite.com/getData?  
secret_api_key=${process.env.SECRET_API_KEY}`);
```



UPDATING .GITIGNORE

If we commit the `.env` file into our git repository, we're back to the same problem we had before!

We must tell git to ignore this file

.GITIGNORE



```
.env
```

If you accidentally commit private data into your git repository, it's quite hard to remove it later
So don't do it!