

Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

 cogilvy-ga / env-walkthrough

<> Code

! Issues

🔗 Pull requests

📁 Projects

📖 Wiki

📈 Insights

🔗 master ▼

...

env-walkthrough / README.md



cogilvy-ga Update README.md

🕒 History

👤 1 contributor

Raw

Blame



73 lines (44 sloc) 1.86 KB

How to Use an Environment File (.env) to Hide Important Info in An Express App

We will be using a npm package called `dotenv` for our environment variables.

Start by reviewing the [dotenv docs](#), then follow the steps below:

Step 1

First cd into your project directory, then install the npm package called `dotenv` , and lastly create a file called `.env` . The `.env` file should live in the top level of your project directory.

For this example, we will assume our project directory is called `express-todos` .

```
cd express-todos
npm i dotenv
touch .env
```

Step 2

In your `server.js` file, add the following code above `var app = express();`

```
//load the env
require('dotenv').config();
```

Step 3

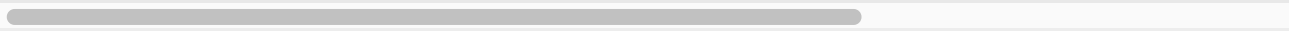
In your `.env` file, you can add variables using the following syntax:

```
VARIABLE_NAME=value
```

We are going to add our connection string to the `.env` file.

You will need to update `<username>` , `<password>` , and `<dbName>` to use your info:

```
DATABASE_URL=mongodb+srv://<username>:<password>@sei.draft.azure.mongodb.net/
```



Step 4

Any variables you add to your `.env` file will now be accessible through the `process.env` object.

The `process.env` global variable is injected by the Node at runtime for your application to use, and it represents the state of the system environment your application is in when it starts.

Here is how you would use that connection string in your express app to connect to your database:

In database.js:

```
const mongoose = require('mongoose');

mongoose.connect(process.env.DATABASE_URL, {
  useNewUrlParser: true,
  useUnifiedTopology: true,
  useCreateIndex: true
});

const db = mongoose.connection;

// database connection event
db.on('connected', function () {
  console.log(`Mongoose connected to: ${db.host}:${db.port}`);
});
```