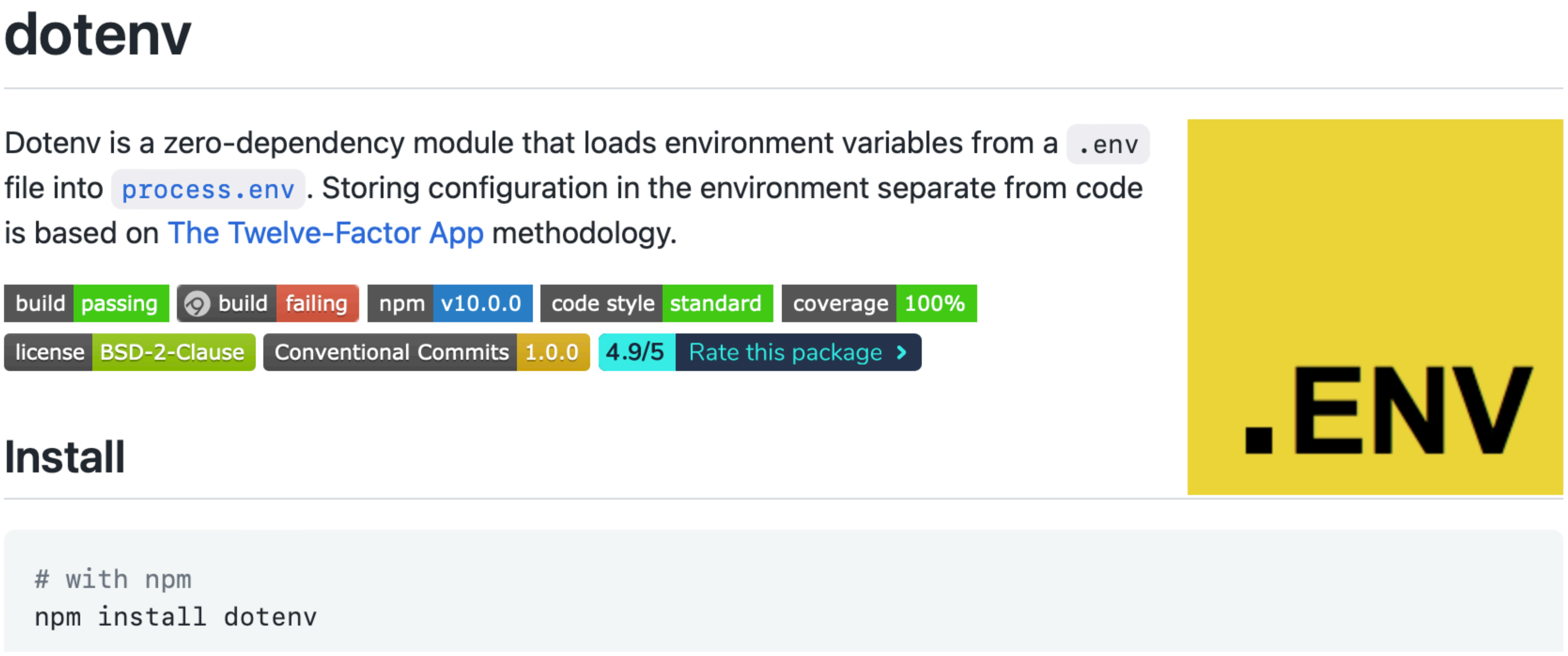
Environment Variables

236705

Full Stack Web Development

https://github.com/motdotla/dotenv



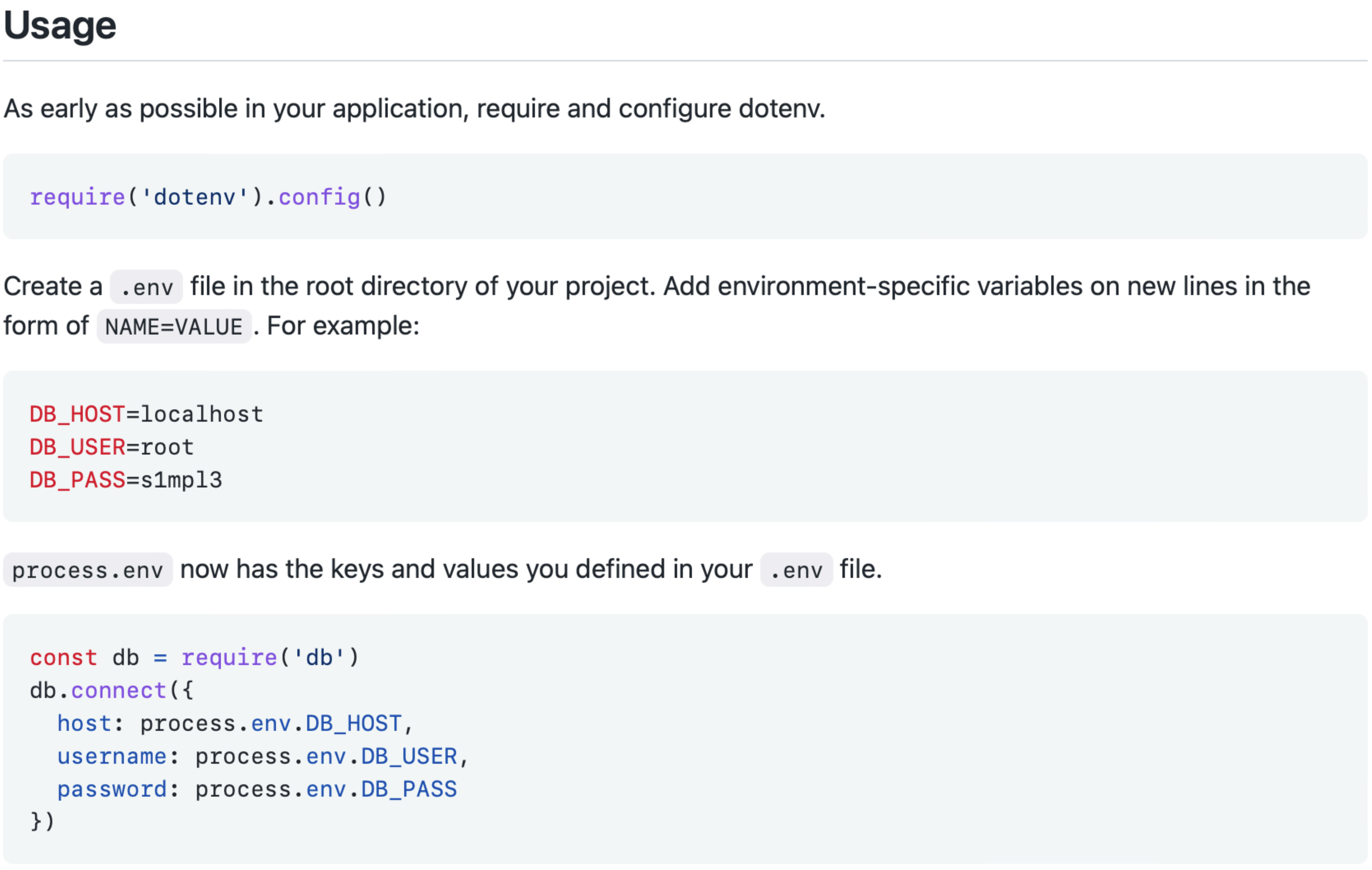
dotenv

Dotenv is a zero-dependency module that loads environment variables from a `.env` file into `process.env`. Storing configuration in the environment separate from code is based on The Twelve-Factor App methodology.

build passing | build failing | npm v10.0.0 | code style standard | coverage 100% | license BSD-2-Clause | Conventional Commits 1.0.0 | 4.9/5 Rate this package ›

Install

```
# with npm
npm install dotenv
```



Usage

As early as possible in your application, require and configure dotenv.

```
require('dotenv').config()
```

Create a `.env` file in the root directory of your project. Add environment-specific variables on new lines in the form of `NAME=VALUE`. For example:

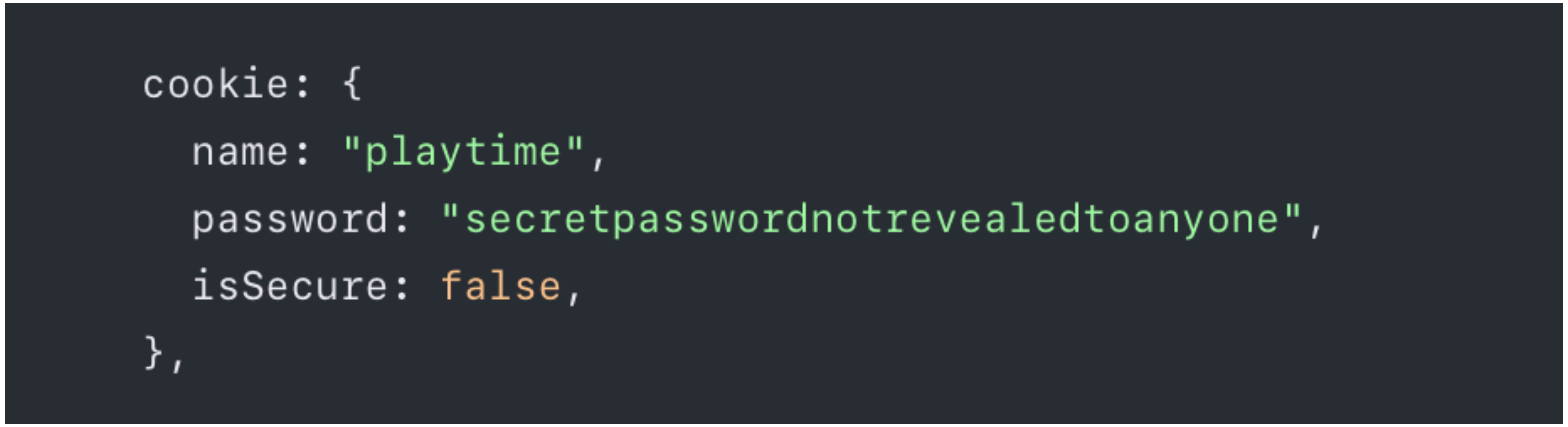
```
DB_HOST=localhost
DB_USER=root
DB_PASS=s1mpl3
```

`process.env` now has the keys and values you defined in your `.env` file.

```
const db = require('db')
db.connect({
  host: process.env.DB_HOST,
  username: process.env.DB_USER,
  password: process.env.DB_PASS
})
```

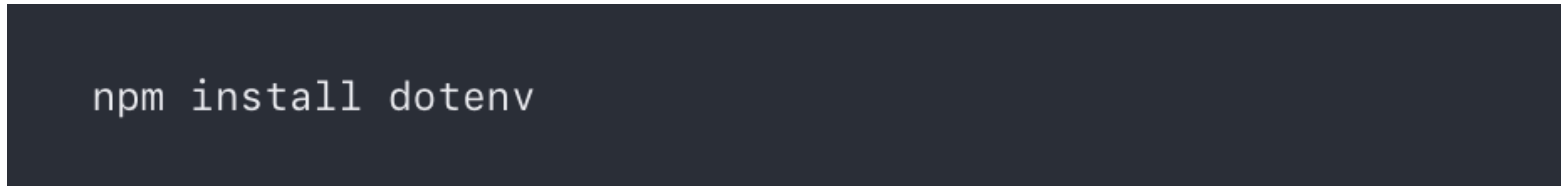
- Currently cooke name + password are hard coded into the server.js

- If application source is compromised, then these secrets become available



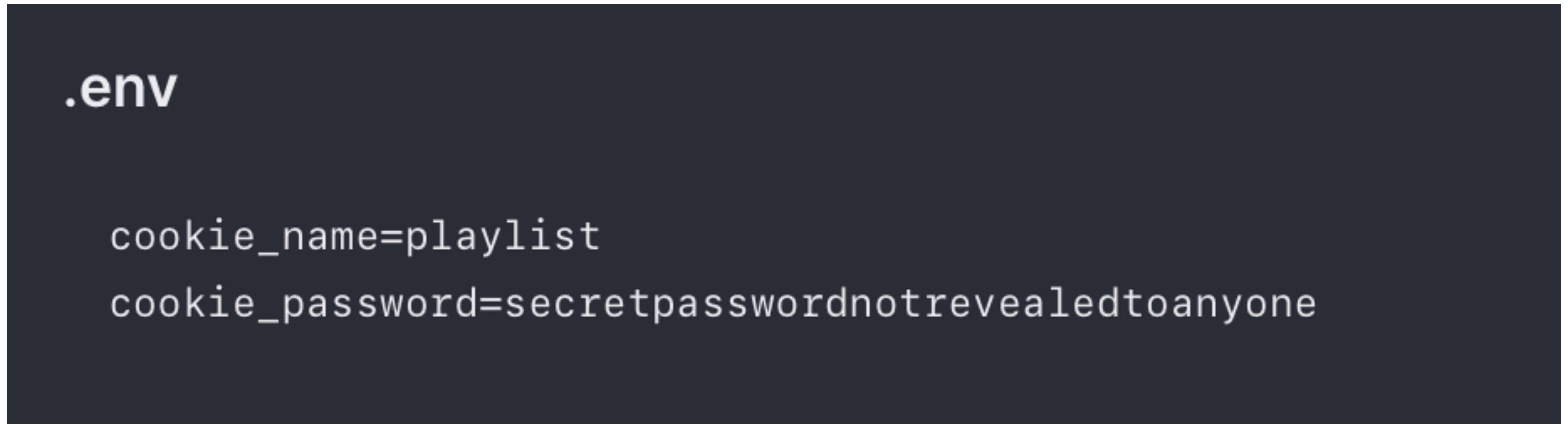
```
cookie: {
  name: "playtime",
  password: "secretpasswordnotrevealedtoanyone",
  isSecure: false,
},
```

- Install dotenv module



```
npm install dotenv
```

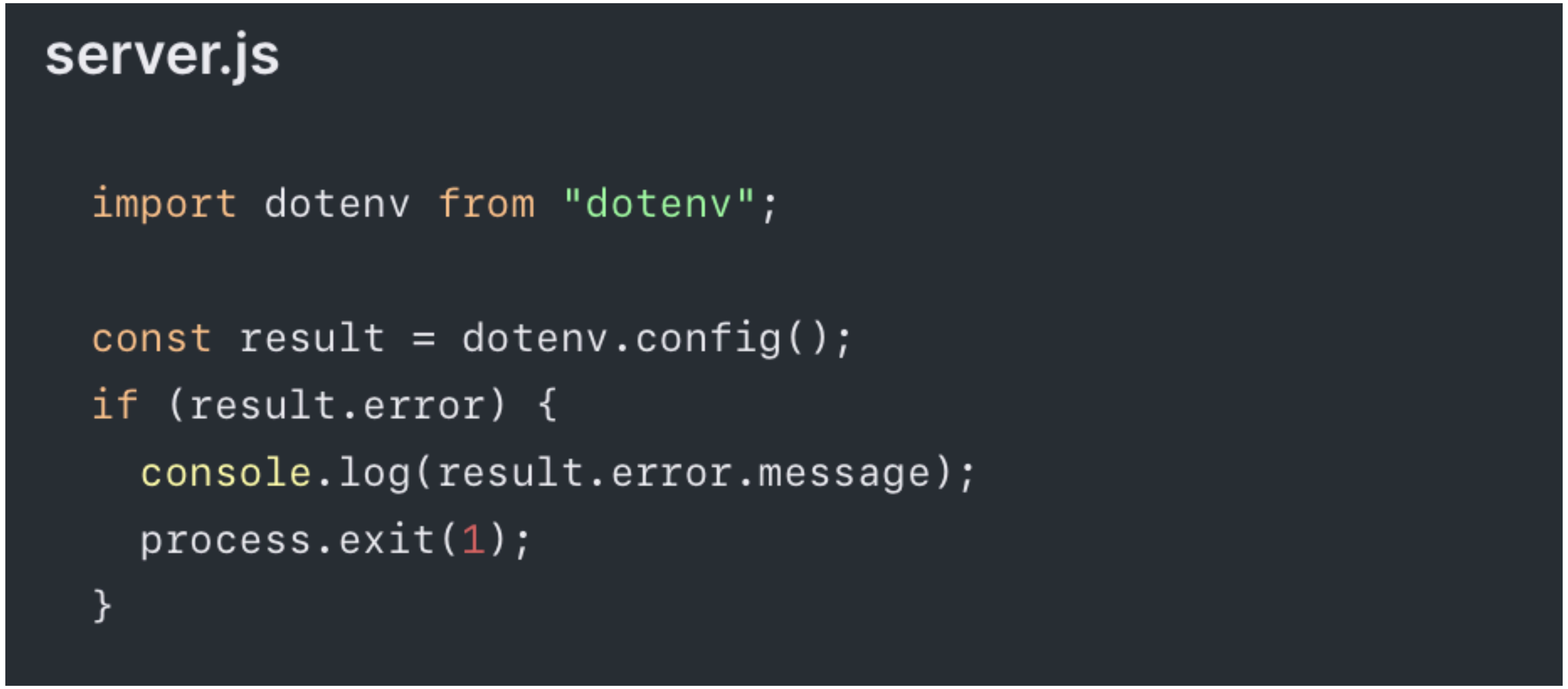
- Introduce text file containing these secrets



.env

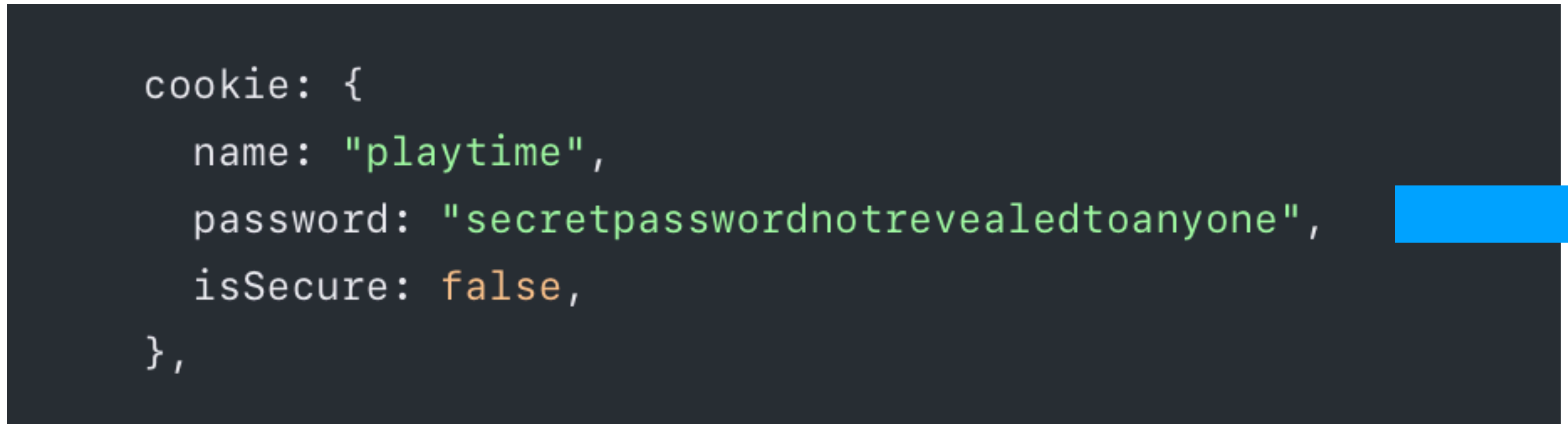
```
cookie_name=playlist
cookie_password=secretpasswordnotrevealedtoanyone
```

- Import dotenv
- Fail to launch application if dot env not found, or incorrectly formatted

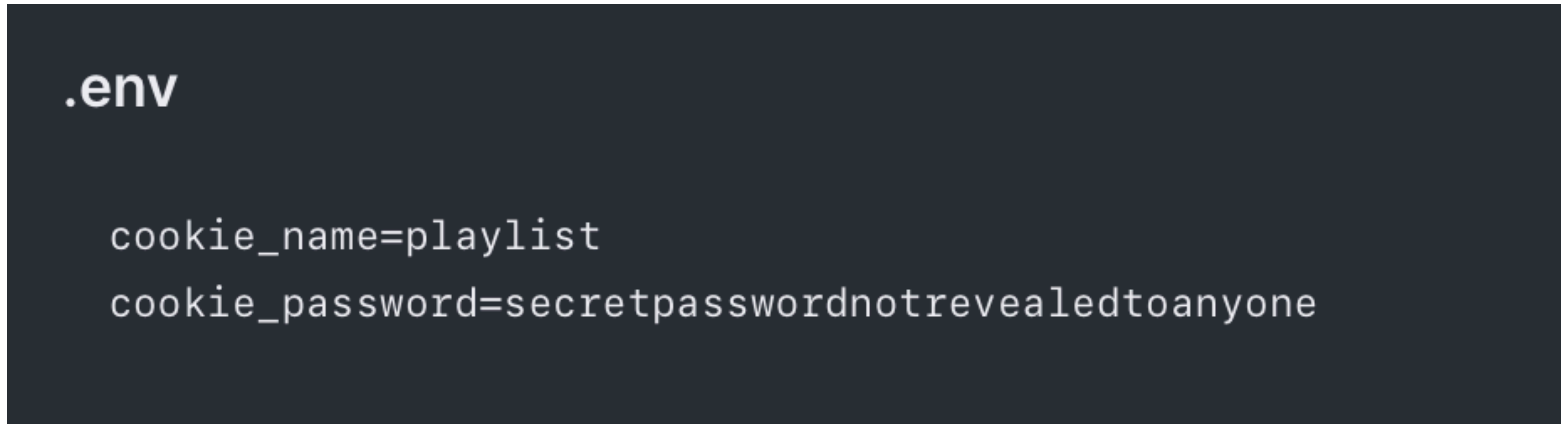


server.js

```
import dotenv from "dotenv";

const result = dotenv.config();
if (result.error) {
  console.log(result.error.message);
  process.exit(1);
}
```

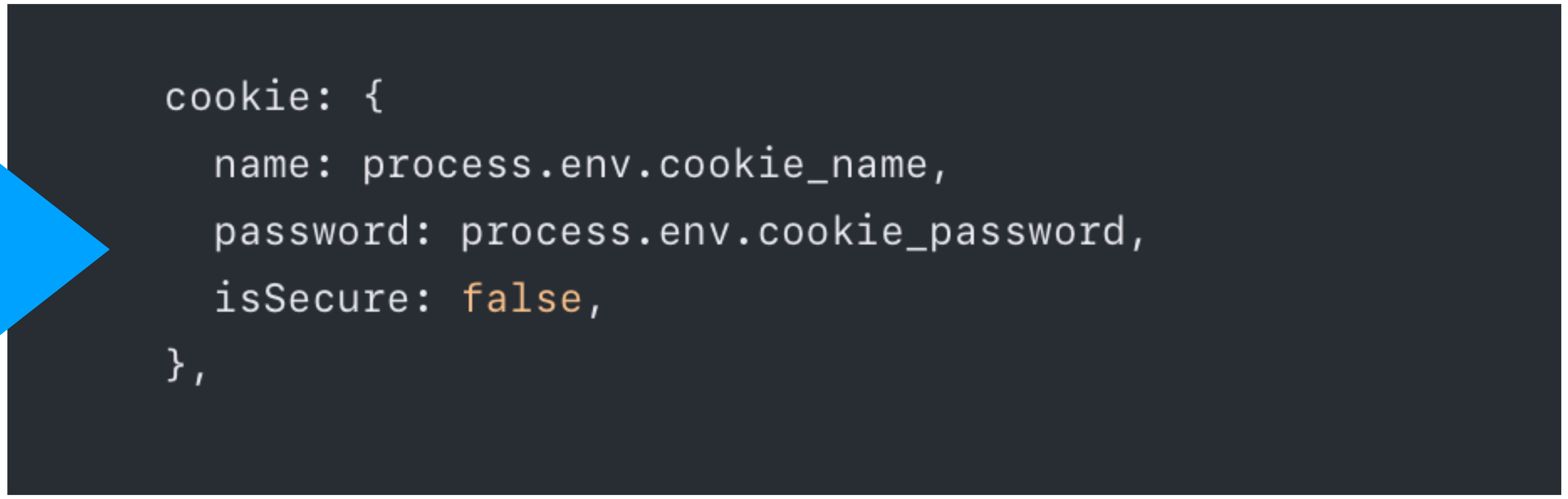


```
cookie: {
  name: "playtime",
  password: "secretpasswordnotrevealedtoanyone",
  isSecure: false,
},
```



.env

```
cookie_name=playlist
cookie_password=secretpasswordnotrevealedtoanyone
```

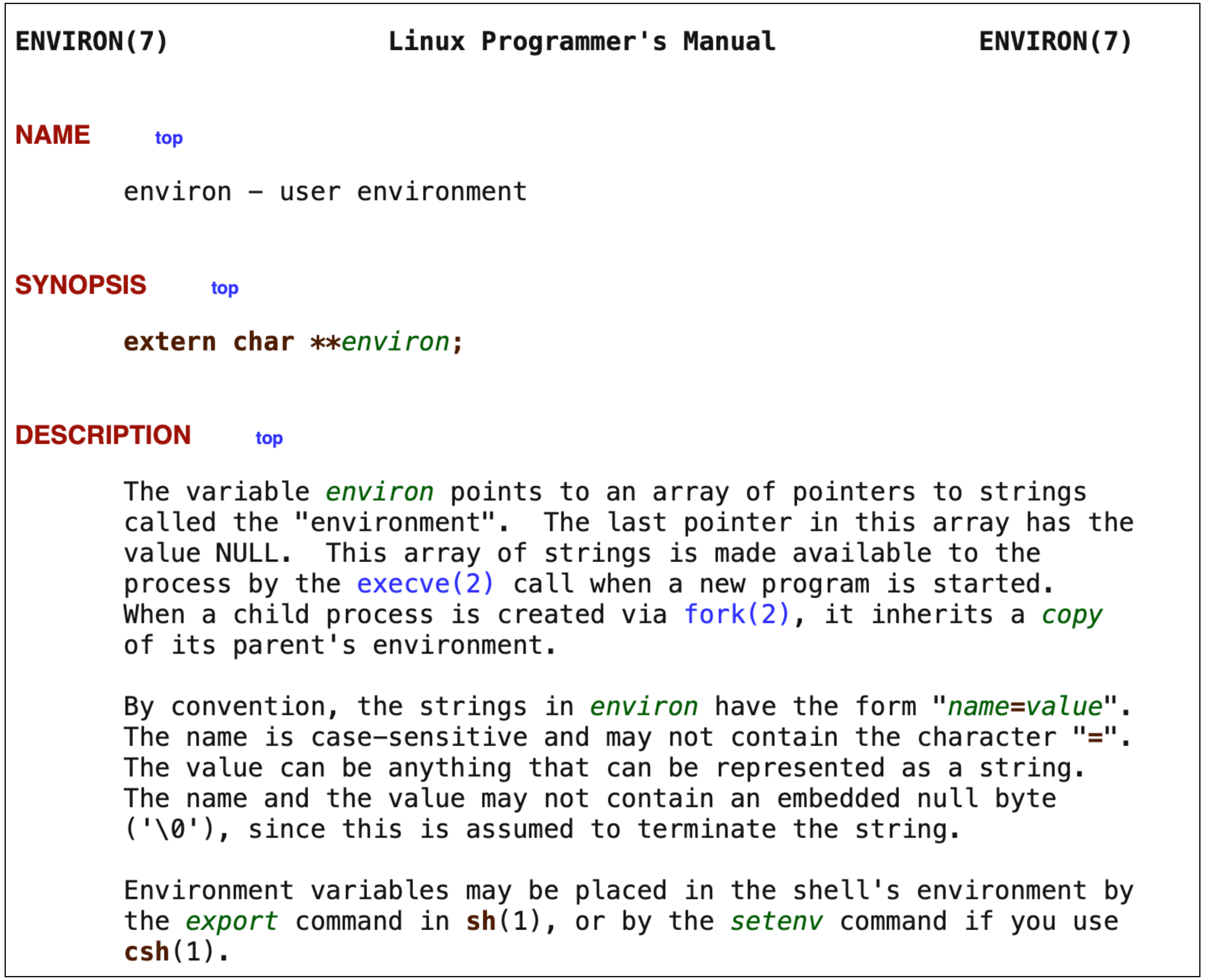


```
cookie: {
  name: process.env.cookie_name,
  password: process.env.cookie_password,
  isSecure: false,
},
```

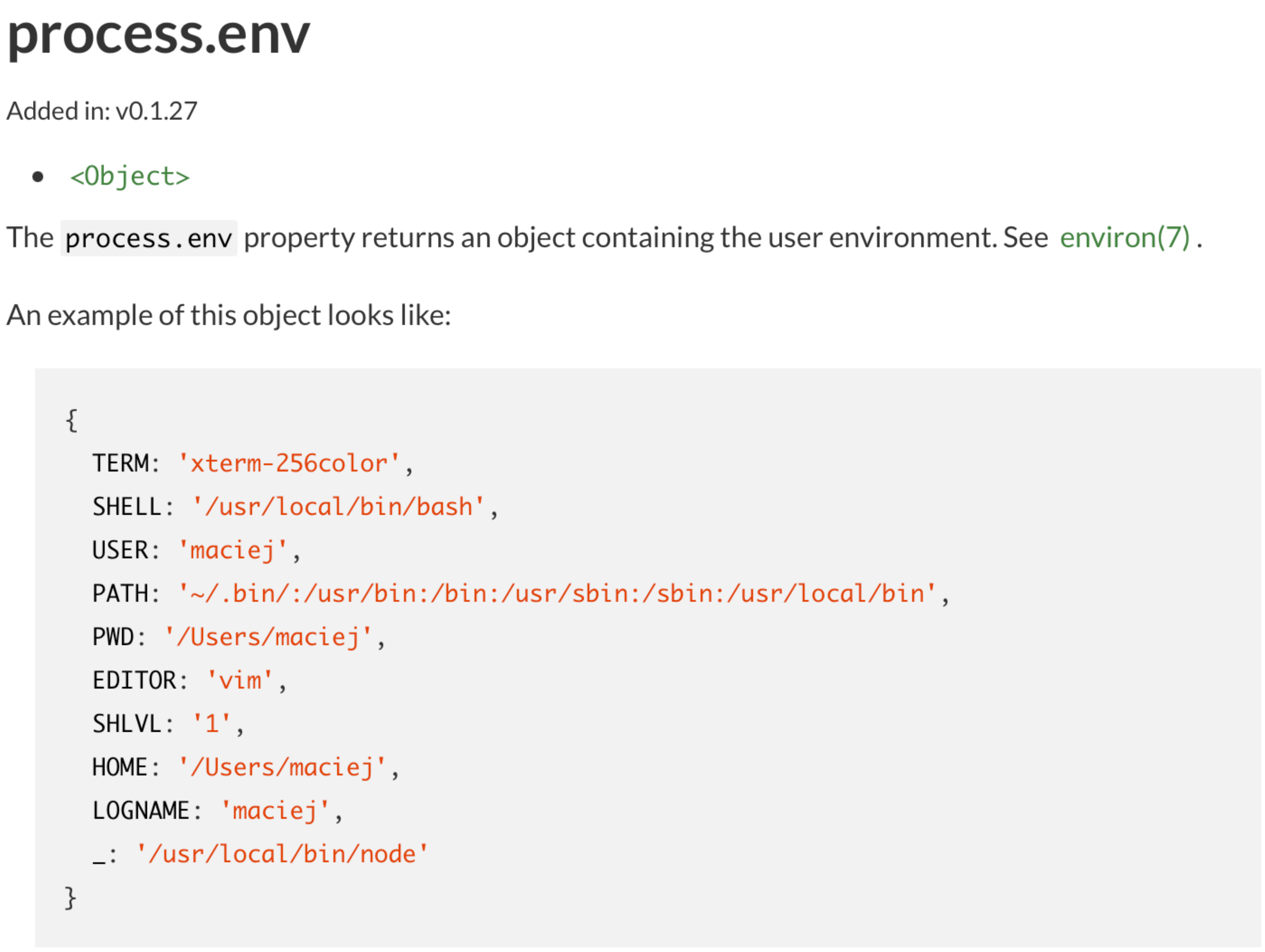
- .env values inject into **process.env**

The Environment

- A Unix concept, the environment encapsulates a set of name/value pairs that define useful contextual information for a process



```
ENVIRON(7)              Linux Programmer's Manual              ENVIRON(7)


NAME       top

       environ - user environment


SYNOPSIS       top

       extern char **environ;


DESCRIPTION       top

       The variable environ points to an array of pointers to strings
       called the "environment".  The last pointer in this array has the
       value NULL.  This array of strings is made available to the
       process by the execve(2) call when a new program is started.
       When a child process is created via fork(2), it inherits a copy
       of its parent's environment.

       By convention, the strings in environ have the form "name=value".
       The name is case-sensitive and may not contain the character "=".
       The value can be anything that can be represented as a string.
       The name and the value may not contain an embedded null byte
       ('\0'), since this is assumed to terminate the string.

       Environment variables may be placed in the shell's environment by
       the export command in sh(1), or by the setenv command if you use
       csh(1).
```



process.env

Added in: v0.1.27

- <Object>

The `process.env` property returns an object containing the user environment. See environ(7) .

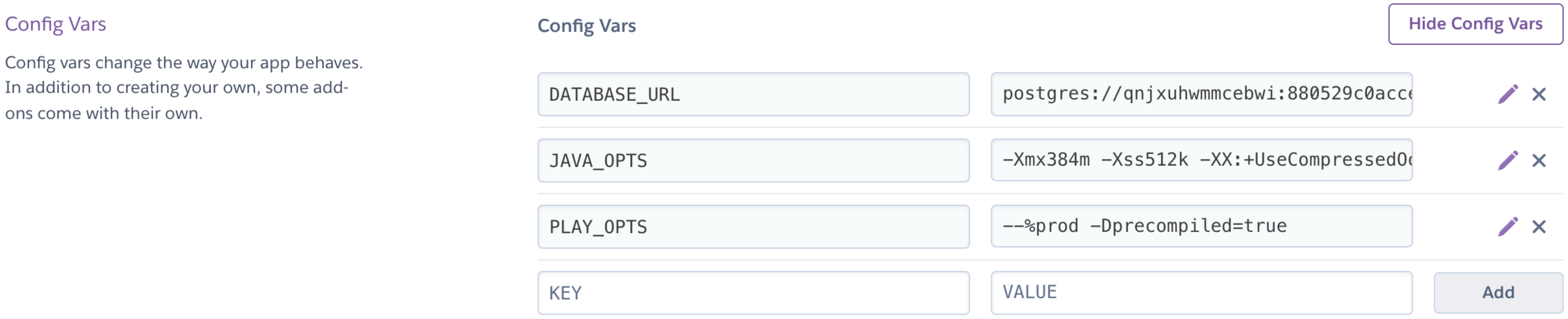
An example of this object looks like:

```
{
  TERM: 'xterm-256color',
  SHELL: '/usr/local/bin/bash',
  USER: 'maciej',
  PATH: '~/.bin/:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin',
  PWD: '/Users/maciej',
  EDITOR: 'vim',
  SHLVL: '1',
  HOME: '/Users/maciej',
  LOGNAME: 'maciej',
  _: '/usr/local/bin/node'
}
```

- Node makes the environment available via the process.env identifier.

- As well as .env files, variables are often also specified on the configuration dashboard for could services

- These should override values in .env file



Config Vars

Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

Config Vars

DATABASE_URL	postgres://qnjxuhwmmcebwi:880529c0acce
JAVA_OPTS	-Xmx384m -Xss512k -XX:+UseCompressedOo
PLAY_OPTS	--%prod -Dprecompiled=true
KEY	VALUE

Heroku Dashboard

Environment Variables

.ENV

Full Stack Web Development