

USANDO EL OBJETO PROCESS

AGREGAR DOTENV

Formato: link a un repositorio en Github con el proyecto cargado.

Sugerencia: no incluir los node_modules

Desafío
entregable



>> Consigna:

Sobre el proyecto del último desafío entregable, mover todas las claves y credenciales utilizadas a un archivo **.env**, y cargarlo mediante la librería **dotenv**.

La única configuración que no va a ser manejada con esta librería va a ser el puerto de escucha del servidor. Éste deberá ser leído de los argumento pasados por línea de comando, usando alguna librería (minimist o yargs). En el caso de no pasar este parámetro por línea de comandos, conectar por defecto al puerto 8080.

Observación: por el momento se puede dejar la elección de sesión y de persistencia explicitada en el código mismo. Más adelante haremos también parametrizable esta configuración.

CODER HOUSE

USANDO EL OBJETO PROCESS

Formato: link a un repositorio en Github con el proyecto cargado.

Sugerencia: no incluir los node_modules

Desafío
entregable



>> Consigna:

Agregar una ruta '/info' que presente en una vista sencilla los siguientes datos:

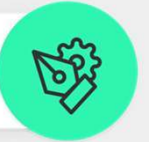
- Argumentos de entrada
- Path de ejecución
- Nombre de la plataforma (sistema operativo)
- Process id
- Versión de node.js
- Carpeta del proyecto
- Memoria total reservada (rss)

USANDO EL OBJETO PROCESS

Formato: link a un repositorio en Github con el proyecto cargado.

Sugerencia: no incluir los node_modules

Desafío
entregable



>> Consigna:

Agregar otra ruta '/api/randoms' que permita calcular un cantidad de números aleatorios en el rango del 1 al 1000 especificada por parámetros de consulta (query). Por ej: /randoms?cant=20000.

Si dicho parámetro no se ingresa, calcular 100.000.000 números.

El dato devuelto al frontend será un objeto que contendrá como claves los números random generados junto a la cantidad de veces que salió cada uno. Esta ruta no será bloqueante (utilizar el método fork de child process). Comprobar el no bloqueo con una cantidad de 500.000.000 de randoms.

Observación: utilizar routers y apis separadas para esta funcionalidad.

CODER HOUSE