Full Stack Developer

Diseño y desarrollo de APIs

Ejercicios resueltos

Ejercicio 1

Para comprobar el enrutamiento de peticiones HTTP en un proyecto Express y NodeJS, crea un proyecto en un directorio denominado ejercicio1; usando Visual Studio Code, deberás crear la siguiente estructura de archivos y directorio:

- **Un archivo app.js.**
- **Un directorio denominado routes.**
- **Un archivo alumns.js dentro del directorio routes.**
- **Un archivo teachers.js dentro del directorio routes.**
- **Las dependencias de NodeJS generadas al inicializar un proyecto incluyendo Express.**

A continuación, desarrolla los siguientes pasos:

- **Genera un proyecto NodeJS con la herramienta de línea de comandos.**
- **Instala como dependencia de desarrollo la herramienta nodemon.**
- **Crea un *script* en el archivo package.json para ejecutar el proyecto con nodemon.**
- **Crea un archivo .gitignore para que el directorio node_modules no se controle en las versiones.**
- **Crea en el archivo alumns.js un método "get" que devuelva un JSON con un ejemplo de datos de alumnos.**
- **Crea en el archivo teachers.js un método "get" que devuelva un JSON con un ejemplo de datos de profesores.**

- En el archivo app.js, implementa dos rutas /alumns y /teachers para las anteriores peticiones.

- Ejecuta el proyecto con el *script* y comprueba con Postman las peticiones para cada ruta.

Ejercicio 2

Para comprobar el uso de códigos de estado y el método "post" en un proyecto Express NodeJS, crea un proyecto en un directorio denominado ejercicio2 y, usando Visual Studio Code, deberás crear la siguiente estructura de archivos y directorio:

- Un archivo app.js.

- Un directorio denominado routes.

- Un archivo players.js dentro del directorio routes.

- Las dependencias de NodeJS generadas al inicializar un proyecto incluyendo Express.

A continuación, desarrolla los siguientes pasos:

- Genera un proyecto NodeJS con la herramienta de línea de comandos.

- Instala como dependencia de desarrollo la herramienta nodemon.

- Crea un *script* en el archivo package.json para ejecutar el proyecto con nodemon.

- Crea un archivo .gitignore para que el directorio node_modules no se controle en las versiones.

- Crea en el archivo players.js un método "get" que devuelva un JSON con un ejemplo de datos de jugadores.

- **En el mismo archivo players.js, crea un método “post” que valide todos los datos para añadir un nuevo jugador en el body de la petición.**

- **Para el método “post”, implementa la respuesta de error si no se cumple la validación.**

- **En el archivo app.js, implementa la ruta /players para las anteriores peticiones.**

- **Ejecuta el proyecto con el script y comprueba con Postman las peticiones.**

Ejercicio 3

Para comprobar las operaciones CRUD en un proyecto NodeJS y Express, crea un proyecto en un directorio denominado ejercicio3; usando Visual Studio Code, deberás crear la siguiente estructura de archivos y directorio:

- **Un archivo app.js.**
- **Un directorio denominado routes.**
- **Un archivo books.js dentro del directorio routes.**
- **Las dependencias de NodeJS generadas al inicializar un proyecto incluyendo Express.**

A continuación, desarrolla los siguientes pasos:

- **Genera un proyecto NodeJS con la herramienta de línea de comandos.**
- **Instala como dependencia de desarrollo la herramienta nodemon.**
- **Crea un *script* en el archivo package.json para ejecutar el proyecto con nodemon.**
- **Crea un archivo .gitignore para que el directorio node_modules no se controle en las versiones.**
- **Crea en el archivo books.js un método "get" que devolverá el *array* de libros, cuya estructura tendrá un campo de ID, un campo de título y un campo de autor.**
- **En el mismo archivo books.js, crea un método "post" para añadir un registro de libro con los campos anteriores.**
- **Para completar las operaciones CRUD, añade un método "put" y otro "delete" para, a partir del campo "id", actualizar o borrar respectivamente un registro de libro en el *array*.**

- **En el archivo app.js implementa la ruta /books para las anteriores peticiones.**

- **Ejecuta el proyecto con el *script* y comprueba con Postman las peticiones.**