Funcionamiento de la API

En este documento vamos a tratar cómo funcionan los distintos Endpoints en nuestra API de Aether.

Contenido

- Registro de usuarios

Este Endpoint espera recibir una petición POST por parte del cliente. Si recibe otro tipo de petición, éste devolverá un error. Acto seguido comprueba si los parámetros que ha recibido en la petición POST son los correctos. Si no lo son, lanza un "echo" diciendo que el nombre de alguno de los parámetros es erróneo. Seguidamente comprueba que el correo no exista en la BBDD, si existe salta una alerta y debes comenzar el proceso desde el principio. Si no existe, registra al usuario finalmente, a no ser que haya habido un error, con lo que te saltaría una alerta de que algo ha ido mal con el servidor, y habría que empezar de nuevo.

- Registro de usuarios APP

Este Endpoint espera recibir una petición POST por parte del cliente. Si recibe otro tipo de petición, éste devolverá un error. Acto seguido comprueba si los parámetros que ha recibido en la petición POST son los correctos. Si no lo son, lanza un "echo" diciendo que el nombre de alguno de los parámetros es erróneo. Seguidamente comprueba que el correo no exista en la BBDD, si existe salta una alerta y debes comenzar el proceso desde el principio. Si no existe, registra al usuario finalmente, a no ser que haya habido un error, con lo que te saltaría una alerta de que algo ha ido mal con el servidor, y habría que empezar de nuevo.

- Recuperar contraseña

Este Endpoint espera recibir una petición POST por parte del cliente. Si recibe otro tipo de petición, éste devolverá un error. Acto seguido comprueba que exista la combinación correo-código en la BBDD. Si existe se cambia la contraseña por la nueva otorgada.

- Obtener media aire

Este Endpoint espera recibir una petición GET por parte del cliente. Si recibe otro tipo de petición, éste devolverá un error. Acto seguido comprueba que los parámetros enviados son los correctos y necesarios. Si no lo son, devuelve un mensaje de error. Pero si son los correctos se obtiene la media de calidad de aire y se devuelve.

- Gestión Mediciones

Este Endpoint espera recibir una petición POST o GET por parte del cliente:

Peticion POST:

Comprueba que la petición sea POST. Acto seguido comprueba si los parámetros que ha recibido en la petición POST son los correctos. Si no lo son, lanza un "echo" diciendo que el nombre de alguno de los parámetros es erróneo. Si son los correctos, guarda la medición concreta en la BBDD.

Peticion GET:

Este Endpoint espera recibir una petición GET por parte del cliente. Acto seguido comprueba que los parámetros enviados son los correctos y necesarios. Si no lo son, devuelve un mensaje de error. Pero si son los correctos se obtiene la ultima medición si el parámetro "datos" vale "u", o todas las mediciones si vale "t".

- Enviar Correo Registro Web

Este Endpoint espera recibir una petición POST por parte del cliente. Si recibe otro tipo de petición, éste devolverá un error. Acto seguido envía un email a la dirección de correo electrónico dada por los parámetros del POST ("nombre", "correo" y "contrasenya"). Si el envío falla se delvolverá el codigo de error del envio de emails.

- disponeSensor

Este endpoint maneja una solicitud HTTP GET. Si la solicitud es del tipo GET, verifica si se ha proporcionado un parámetro llamado "correo" en la solicitud. Si es así, llama a otra función llamada "disponeSensor" y le pasa el valor del parámetro "correo". Si "disponeSensor" devuelve true, la función devuelve true y muestra "true" en pantalla. Si "disponeSensor" devuelve false, la función devuelve false y muestra "false" en pantalla. Si el parámetro "correo" no se proporciona en la solicitud, la función muestra un mensaje de error y devuelve null. Si la solicitud no es del tipo GET, la función muestra un mensaje de error y establece el código de respuesta HTTP en 400 (Solicitud incorrecta).

- datosUsuario

Este endpoint maneja una solicitud HTTP GET. Si la solicitud es del tipo GET, verifica si se ha proporcionado un parámetro llamado "correo" en la solicitud. Si es así, llama a otra función llamada "obtenerDatosUsuario" y le pasa el valor del parámetro "correo". Si "obtenerDatosUsuario" devuelve null, significa que no se han encontrado datos del usuario con el correo especificado y muestra un mensaje de error en pantalla. Si "obtenerDatosUsuario" devuelve un resultado distinto de null, almacena el resultado en una variable llamada "datosUsuario" y luego muestra el contenido de esta variable en pantalla. Si el parámetro "correo" no se proporciona en la solicitud, la función no hace nada. Si la solicitud no es del tipo GET, la función muestra un mensaje de error y establece el código de respuesta HTTP en 400 (Solicitud incorrecta).

- comprobarCredencialesWeb

Este Endpoint maneja una solicitud HTTP POST. Si la solicitud es del tipo POST, verifica si se han proporcionado los parámetros "correo" y "contrasenya" en la solicitud. Si es así, almacena los valores de estos parámetros en las variables "$correo" y "$contrasenya", respectivamente. Luego, encripta la contraseña utilizando el algoritmo SHA-512 y almacena el resultado en la variable "$contrasenya_encriptada". Después, llama a una función llamada "comprobarCredenciales" y le pasa los valores de "$correo" y "$contrasenya_encriptada" como parámetros. Si alguno de los parámetros "correo" o "contrasenya" no se proporciona en la solicitud, la función muestra un mensaje de error. Si la solicitud no es del tipo POST, la función muestra un mensaje de error y establece el código de respuesta HTTP en 400 (Solicitud incorrecta).

- comprobarCredenciales

Este Endpoint está dedicado a la APP y maneja una solicitud HTTP POST. Si la solicitud es del tipo POST, verifica si se han proporcionado los parámetros "correo" y "contrasenya" en la solicitud. Si es así, almacena los valores de estos parámetros en las variables "$correo" y "$contrasenya", respectivamente. Luego, encripta la contraseña y almacena el resultado en la variable "$contrasenya_encriptada". Después, llama a una función llamada "comprobarCredenciales" y le pasa los valores de "$correo" y "$contrasenya_encriptada" como parámetros. Si "comprobarCredenciales" devuelve true, inicia una sesión y establece la variable de sesión "usuario" con el valor de "$correo". Luego, redirige al usuario a la página "bienvenida.php". Si alguno de los parámetros "correo" o "contrasenya" no se proporciona en la solicitud o si "comprobarCredenciales" devuelve false, muestra un mensaje de error y redirige al usuario a la página "index.php". Si la solicitud no es del tipo POST, la función muestra un mensaje de error y establece el código de respuesta HTTP en 400 (Solicitud incorrecta).

- cambiarNombreUsuario

Este Endpoint maneja una solicitud HTTP POST. Si la solicitud es del tipo POST, verifica si se han proporcionado los parámetros "nombre" y "correo" en la solicitud. Si es así, almacena los valores de estos parámetros en las variables "$nombre" y "$correo", respectivamente. Luego, elimina los espacios en blanco del nombre y almacena el resultado en la variable "$nombreSinEspacios". Después, llama a una función llamada "cambiarNombreUsuario" y le pasa los valores de "$correo" y "$nombreSinEspacios" como parámetros. Si "cambiarNombreUsuario" devuelve true, muestra un mensaje indicando que se ha cambiado el nombre del usuario. Si "cambiarNombreUsuario" devuelve false, muestra un mensaje de error. Si alguno de los parámetros "nombre" o "correo" no se proporciona en la solicitud, la función muestra un mensaje de error. Si la solicitud no es del tipo POST, la función muestra un mensaje de error y establece el código de respuesta HTTP en 400 (Solicitud incorrecta).

- cambiarContrasenya

En este Endpoint, si la solicitud es del tipo POST, se verifica si se han proporcionado los parámetros "contrasenya", "correo" y "nuevaContrasenya" en la solicitud. Si es así, se almacenan los valores de estos parámetros en las variables "$contrasenya", "$correo" y "$nuevaContrasenya", respectivamente. Luego, se encripta las contraseñas y almacena los resultados en las variables "$contrasenyaEncriptada" y "$nuevaContrasenyaEncriptada", respectivamente. Después, se llama a una función llamada "cambiarContrasenya" y se le pasa los valores de "$correo", "$contrasenyaEncriptada" y "$nuevaContrasenyaEncriptada" como parámetros. Si "cambiarContrasenya" devuelve true, se muestra un mensaje indicando que se ha cambiado la contraseña del usuario y se redirige al usuario a la página "index.php". Si alguno de los parámetros "contrasenya", "correo" o "nuevaContrasenya" no se proporciona en la solicitud, la función muestra un mensaje de error. Si la solicitud no es del tipo POST, la función muestra un mensaje de error y establece el código de respuesta HTTP en 400 (Solicitud incorrecta).

- borrarUsuario

Este endpoint maneja una solicitud HTTP POST. Si la solicitud es del tipo POST, verifica si se han proporcionado los parámetros "contrasenya" y "correo" en la solicitud. Si es así, almacena los valores de estos parámetros en las variables "$correo" y "$contrasenya", respectivamente. Luego, encripta la contraseña y almacena el resultado en la variable "$contrasenya_encriptada". Después, llama a una función llamada "borrarUsuario" y le pasa los valores de "$correo" y "$contrasenya_encriptada" como parámetros. Si "borrarUsuario" devuelve true, muestra un mensaje indicando que se ha borrado el usuario. Si "borrarUsuario" devuelve false, muestra un mensaje de error. Si alguno de los parámetros "contrasenya" o "correo" no se proporciona en la solicitud, la función muestra un mensaje de error. Si la solicitud no es del tipo POST, la función muestra un mensaje de error y establece el código de respuesta HTTP en 400 (Solicitud incorrecta).

- asignarSensorUsuario

Este Endpoint maneja una solicitud HTTP POST. Si la solicitud es del tipo POST, verifica si se han proporcionado los parámetros "idSensor" y "correo" en la solicitud. Si es así, almacena los valores de estos parámetros en las variables "$correo" y "$idSensor", respectivamente. Después, llama a una función llamada "asignarSensorUsuario" y le pasa los valores de "$correo" y "$idSensor" como parámetros. Si "asignarSensorUsuario" devuelve true, muestra un mensaje indicando que se ha asignado el sensor al usuario. Si "asignarSensorUsuario" devuelve false, muestra un mensaje de error. Si alguno de los parámetros "idSensor" o "correo" no se proporciona en la solicitud, la función muestra un mensaje de error. Si la solicitud no es del tipo POST, la función muestra un mensaje de error y establece el código de respuesta HTTP en 400 (Solicitud incorrecta).