

API Rest para app frontend SmartCloth

/api/user/login

Accede a sesión de usuario con nombre de usuario y contraseña.

Método: POST

Parámetros:

- **username:** Nombre de usuario
- **password:** Contraseña
- **token:** Token de seguridad de la aplicación para evitar intentos de login externos

Devuelve:

Objeto JSON con los siguientes datos:

- **success:** true o false. Valor booleano que define si ha tenido éxito la operación.
- **session_id:** token de sesión del usuario (si ha tenido éxito el login)
- **user:** objeto con los datos del usuario (si ha tenido éxito el login)
- **error:** Mensaje de error solo en caso de que no haya tenido éxito la operación.

Puede tener los siguientes valores:

- *user_not_found.* (Nombre de usuario no registrado)
- *wrong_password.* (Contraseña incorrecta)

/api/user/register

Registra una cuenta de usuario nueva.

Método: POST

Parámetros:

- **username:** Nombre de usuario
- **email:** Email de usuario
- **age:** Edad del usuario (número entero)
- **password:** Contraseña
- **sex:** 'Male' o 'Female' (cadena que indica el sexo del usuario)
- **image:** (subida de fichero de imagen al servidor desde el ordenador del usuario)
- **token:** Token de seguridad de la aplicación para evitar intentos de registro externos

Devuelve:

Objeto JSON con los siguientes datos:

- **success:** true o false. Valor booleano que define si ha tenido éxito la operación.
- **error:** Mensaje de error solo en caso de que no haya tenido éxito la operación.

Puede tener los siguientes valores:

- *username_repeat*
- *email_repeat*

Nota:

Indícame cuando puedas las restricciones que tendrá en el backend la contraseña (longitud, tipos de caracteres, etc) para que valide en frontend antes de enviarla

Pregunta:

¿No sería más práctico poner la fecha de nacimiento para que el valor de la edad se fuera actualizando solo?

/api/user/resetpassword

Asigna una contraseña aleatoria nueva al usuario y se la envía por email.

Método: POST

Parámetros:

- **email:** Email de usuario
- **token:** Token de seguridad de la aplicación para evitar intentos externos

Devuelve:

Objeto JSON con los siguientes datos:

- **success:** true o false. Valor booleano que define si ha tenido éxito la operación.
- **error:** Mensaje de error solo en caso de que no haya tenido éxito la operación con el siguiente valor:
 - *email_not_found*

Nota:

Tendrías que enviar un email al usuario con un enlace con caducidad para resetear su contraseña

/api/user

Solicita los datos del usuario

Método: GET

Parámetros:

- **token:** Token de sesión de usuario. Realizar la operación solo si el token es correcto.

Devuelve:

Objeto JSON con los siguientes datos:

- **success:** true o false. Valor booleano que define si ha tenido éxito la operación.
- **user:** objeto con los datos del usuario (si ha tenido éxito la operación)
- **error:** Mensaje de error solo en caso de que no haya tenido éxito la operación.

Puede tener el siguiente valor:

- *not_logged_in* (El token de sesión es incorrecto)
-

/api/user/edit/username

Edita el nombre del usuario que está logueado.

Método: PATCH

Parámetros:

- **username:** Nuevo nombre de usuario
- **token:** Token de sesión de usuario. Realizar la operación solo si el token es correcto.

Devuelve:

Objeto JSON con los siguientes datos:

- **success:** true o false. Valor booleano que define si ha tenido éxito la operación.
- **error:** Mensaje de error solo en caso de que no haya tenido éxito la operación.

Puede tener el siguiente valor:

- *username_repeat* (Nombre de usuario repetido. Nota: No puede repetirse con respecto a otros usuarios, sí puede poner el mismo nombre que ya tenía antes él mismo)
- *not_logged_in* (El token de sesión es incorrecto)

/api/user/edit/email

Edita el email del usuario que está logueado.

Método: PATCH

Parámetros:

- **email:** Nuevo email de usuario
- **token:** Token de sesión de usuario. Realizar la operación solo si el token es correcto.

Devuelve:

Objeto JSON con los siguientes datos:

- **success:** true o false. Valor booleano que define si ha tenido éxito la operación.
- **error:** Mensaje de error solo en caso de que no haya tenido éxito la operación.
Puede tener el siguiente valor:
 - *username_repeat* (Email de usuario repetido. Nota: No puede repetirse con respecto a otros usuarios, sí puede poner el mismo email que ya tenía antes él mismo)
 - *not_logged_in* (El token de sesión es incorrecto)

/api/user/edit/age

Edita la edad del usuario que está logueado.

Método: PATCH

Parámetros:

- **age:** Nueva edad del usuario (número entero)
- **token:** Token de sesión de usuario. Realizar la operación solo si el token es correcto.

Devuelve:

Objeto JSON con los siguientes datos:

- **success:** true o false. Valor booleano que define si ha tenido éxito la operación.
- **error:** Mensaje de error solo en caso de que no haya tenido éxito la operación.
Puede tener el siguiente valor:
 - *not_logged_in* (El token de sesión es incorrecto)

/api/user/edit/sex

Edita la sexo del usuario que está logueado.

Método: PATCH

Parámetros:

- **sex:** 'Male' o 'Female' (cadena que indica el sexo del usuario)
- **token:** Token de sesión de usuario. Realizar la operación solo si el token es correcto.

Devuelve:

Objeto JSON con los siguientes datos:

- **success:** true o false. Valor booleano que define si ha tenido éxito la operación.
- **error:** Mensaje de error solo en caso de que no haya tenido éxito la operación. Puede tener el siguiente valor:
 - *not_logged_in* (El token de sesión es incorrecto)

/api/user/edit/password

Edita la contraseña del usuario que está logueado.

Método: PATCH

Parámetros:

- **current_password:** Contraseña actual (Debe comprobarse que sea correcta)
- **password:** Contraseña nueva
- **token:** Token de sesión de usuario. Realizar la operación solo si el token es correcto.

Devuelve:

Objeto JSON con los siguientes datos:

- **success:** true o false. Valor booleano que define si ha tenido éxito la operación.
- **error:** Mensaje de error solo en caso de que no haya tenido éxito la operación. Puede tener el siguiente valor:
 - *wrong_password* . Contraseña actual introducida es incorrecta.
 - *not_logged_in* (El token de sesión es incorrecto)

/api/data

Solicita datos de los valores nutricionales de un intervalo de fechas y hora: Gramos y raciones de grasas, carbohidratos, proteínas; calorías totales consumidas; dosis introducidas de glucemia e insulina en ese periodo.

Método: GET

Parámetros:

- **start:** Fecha y hora, formato sql "Y-m-d H:i:s" (Ejemplo: "2023-12-21 13:30:00")
- **end:** Fecha y hora, formato sql "Y-m-d H:i:s" (Ejemplo: "2023-12-21 13:30:00")
- **token:** Token de sesión de usuario. Realizar la operación solo si el token es correcto.

Devuelve:

Objeto JSON con los siguientes datos:

- **success:** true o false. Valor booleano que define si ha tenido éxito la operación.
- **data:** En el caso de que la operación haya tenido éxito. Objeto JSON con los valores indicados en la descripción para el intervalo solicitado: Gramos y raciones de grasas, carbohidratos, proteínas; calorías totales consumidas; dosis introducidas de glucemia e insulina en ese periodo.
- **error:** Mensaje de error solo en caso de que no haya tenido éxito la operación. Puede tener el siguiente valor:
 - *not_logged_in* (El token de sesión es incorrecto)

/api/data/add

El usuario introduce nuevos datos de glucemia e insulina

Método: POST

Parámetros:

- **datetime:** Fecha y hora, formato sql "Y-m-d H:i:s" (Ejemplo: "2023-12-21 13:30:00")
- **glycemia:** Valor numérico de glucemia
- **insulin:** Valor numérico de insulina
- **insulin_type:** Tipo de insulina
- **token:** Token de sesión de usuario. Realizar la operación solo si el token es correcto.

Devuelve:

Objeto JSON con los siguientes datos:

- **success:** true o false. Valor booleano que define si ha tenido éxito la operación.
- **error:** Mensaje de error solo en caso de que no haya tenido éxito la operación.

Puede tener el siguiente valor:

- *not_logged_in* (El token de sesión es incorrecto)