**Estructura de los mensajes para el manejo de respuestas del backend:**

En nuestro proyecto, hemos establecido una estructura estándar para los mensajes que serán utilizados para comunicar el estado de las respuestas del backend. Esta estructura sigue el siguiente formato:

{

"status": "success",

"message": "",

"code": 200,

"results": null

}

* **status:** Indica el estado de la respuesta. Puede ser "success" (éxito), "error" (error) o "info" (información).
* **message:** Contiene un mensaje descriptivo que proporciona detalles sobre el resultado de la operación.
* **code:** Representa el código de estado HTTP asociado con la respuesta.
* **results:** Contiene datos adicionales relevantes para la respuesta, como en el caso de respuestas exitosas.

**Ejemplos de mensajes:**

**Mensajes de error:**

{

"status": "error",

"message": "La solicitud es incorrecta. Falta el parámetro 'nombre'.",

"code": 400,

"results": null

}

{

"status": "error",

"message": "Recurso no encontrado. El usuario solicitado no existe.",

"code": 404,

"results": null

}

{

"status": "error",

"message": "Error interno del servidor. Inténtalo de nuevo más tarde.",

"code": 500,

"results": null

}

**Mensajes de éxito:**

{

"status": "success",

"message": "El recurso fue eliminado exitosamente.",

"code": 204,

"results": null

}

{

"status": "success",

"message": "Usuario creado exitosamente.",

"code": 201,

"results": { "user\_id": 12345 } }

{

"status": "success",

"message": "Los datos del usuario se actualizaron correctamente.",

"code": 200,

"results": null

}

**Mensajes de información (info):**

{

"status": "info",

"message": "La solicitud se recibió correctamente. Espere mientras se procesa.",

"code": 200,

"results": null

}

{ "status": "info",

"message": "Se enviaron correos electrónicos de confirmación a los usuarios.",

"code": 200,

"results": null

}

Uso en el proyecto:

Esta estructura estandarizada nos permitirá comunicar de manera coherente y clara el estado de las respuestas del backend a las aplicaciones cliente. Al seguir esta estructura, aseguramos una fácil interpretación de los mensajes por parte de los desarrolladores y un manejo consistente de las diferentes situaciones que puedan surgir durante la interacción con nuestra API. Esto facilitará el proceso de desarrollo, depuración y mantenimiento de nuestro proyecto.