
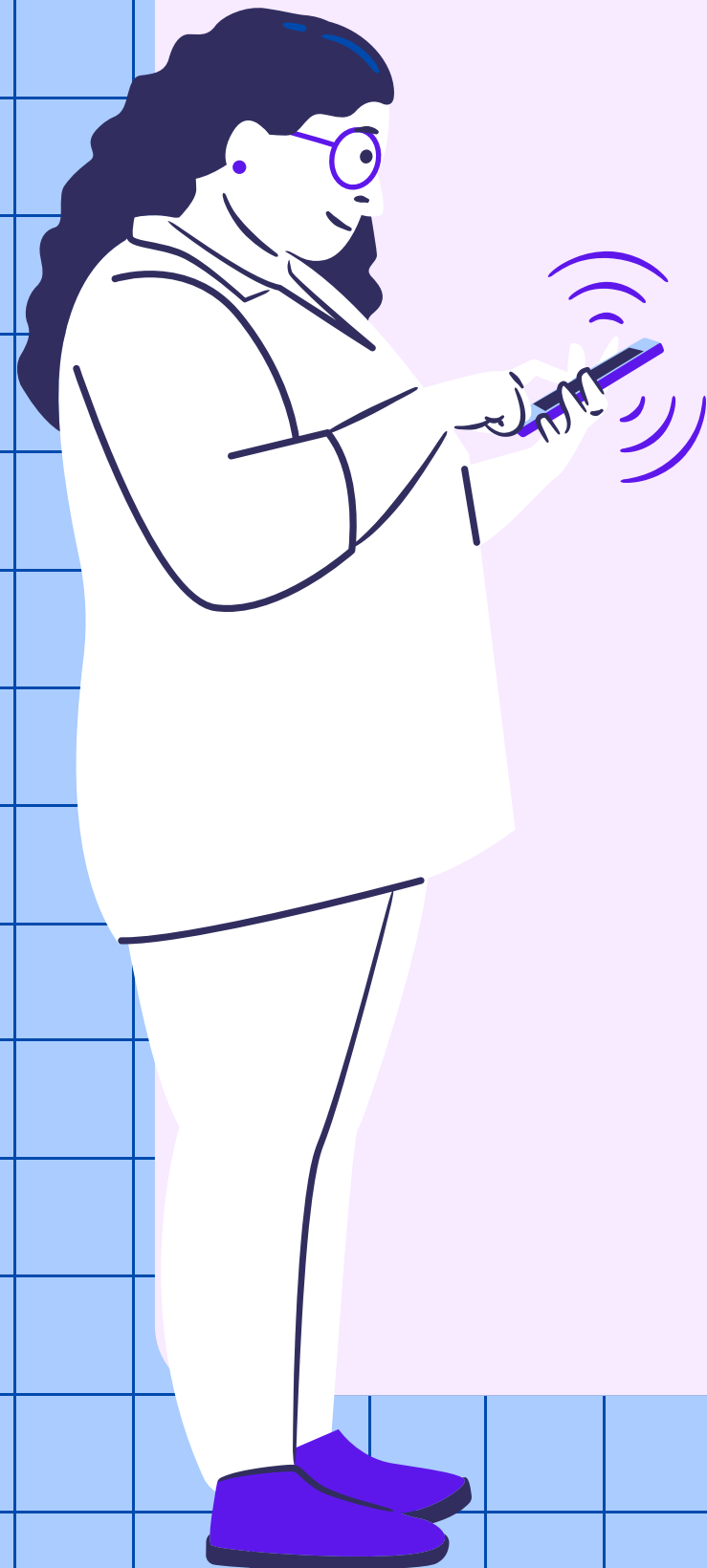


Mejores prácticas "Verbos HTTP"



ARANXA GUADALUPE MARTINEZ OJEDA





Los verbos HTTP mas usados son POST, GET, PUT, PATCH y DELETE

- **POST**- Guarda una nueva representación de un recurso
- **GET** - Solicita el estado de un recurso especificado. No es usado para guardar o actualizar datos.
- **PUT** - Actualiza el estado de un recurso existente
- **PATCH** - Actualiza parcialmente el estado de un recurso existente. Esto es de ayuda con se trata con datos complejos.
- **DELETE** - Elimina un recurso especificado.

RUTAS IDEALES

GET

/personas/{usuario}/roles

Regresa una lista de roles para una persona dado un usuario único

Una buena práctica a seguir es considerar qué forma sería la más clara para el usuario de la API. Una buena regla es usar la relación "pertenece a"

En el ejemplo anterior se puede entender que los roles "pertenecen a" una persona. Un usuario potencial del API podría mirar la URL y, sin saber nada acerca de la estructura general de la API, podría descubrir que es lo que hace el URL anterior.

MÁS EJEMPLOS

GET

`/personas/{usuario}/roles/{rolename}`

Regresa un rol para un solo usuario dado el usuario y el nombre del rol

POST

`/personas/{usuario}/roles`

Agrega un nuevo rol a un usuario dado el usuario

PATCH

`/personas/{usuario}`

Actualiza parcialmente la información dada para un nombre de usuario

GET

`/personas/{usuario}/roles/{rolename}`

Elimina un solo rol dado el nombre del rol para un usuario dado el nombre del usuario

NOMBRES DE COLECCIONES EN PLURAL

Puedes pensar en los datos de tu API como una colección de diferentes recursos de tus clientes.

Si tienes un endpoint como `https://mysite.com/post/123` podría estar bien usar DELETE para borrar un post o actualizar un posr con PUT o PATCH, pero esto no indica al usuario que pueden haber más posts en la colección.

Por esta razón tus colecciones deben de hacer uso de sustantivos en plural.

En lugar de `https://mysite.com/post/123` debe de ser:
`https://mysite.com/posts/123`

CODIGOS DE ESTADO DE RESPUESTA HTTP

Esto ayuda a los usuarios saber que es lo que esta pasando, si su solicitud fue exitosa o ha fallado.

- **100 - 199 Respuestas informativas**

Por ejemplo, 102 indica que el recurso esta siendo procesado

- **200 - 299 Respuestas satisfactorias**

Por ejemplo, 202 indica que la solicitud se ha recibido, pero aún no se ha actuado

- **300 - 399 Redirecciones**

Por ejemplo, 301 indica que la URI ha sido cambiado

- **400 - 499 Errores de los clientes**

Por ejemplo, 404 significa que el recurso no ha sido encontrado

- **500 - 599 Errores de los servidores**

Por ejemplo, 500 significa error interno del servidor