Informe: Análisis de las APIs expuestas de “https://regres.in/

En este informe, analicé las 15 llamadas expuestas en la API de “https://reqres.in/” utilizando la herramienta cURL para ejecutar las solicitudes y revisar las respuestas obtenidas. A continuación, se detalla cada petición, especificando su tipo (GET, POST, PUT, PATCH, DELETE) y un resumen de la información obtenida.

Para las llamadas GET se sobre entiende que en el terminal se realiza la petición con curl + la dirección correspondiente, para el resto de llamadas se indicará el comando correspondiente. Si no se especifica código de respuesta es: 200 OK.

- https://reqres.in/api/users?page=2

  Llamada GET.
  Solicita una lista de usuarios paginada. Incluye el parámetro page=2 por lo que estamos solicitando la segunda página de usuarios.
  La respuesta es un formato JSON que devuelve la información de la página 2 de usuarios, donde hay 6 usuarios en total con diferentes campos.

- https://reqres.in/api/users/2

  Llamada GET.
  Solicita información sobre un usuario en específico, el usuario con ID = 2.
  La respuesta es un formato JSON, devuelve toda la información del usuario 2 , Janet Weaver,.

- https://reqres.in/api/users/23

  Llamada GET.
  Solicita información de un usuario en específico, el usuario con ID = 23.
  La respuesta son unos corchetes vacíos ({}), indican que no hay datos para el usuario solicitado. Código 404 – not found.

- https://reqres.in/api/unknown

  Llamada GET.
  Solicita una lista de recursos.
  Devuelve un formato JSON que incluye una lista de colores con detalles de cada uno.

- https://reqres.in/api/unknown/2

  Llamada GET.
  Solicita una lista de recursos los datos del ID = 2.
  Devuelve Datos en JSON del ID = 2.

- https://reqres.in/api/unknown/23

  Llamada GET.
  Solicita una lista de recursos inexistente.
  Devuelve corchetes vacíos ({}). Código 404 – not found

- https://reqres.in/api/users name=morpheus job=leader

  Llamada POST.
  Solicita incluir un nuevo usuario.
  Realizo petición: curl -X POST https://reqres.in/api/users -d "name=morpheus&job=leader".
  La respuesta ha sido un formato JSON con los datos que le facilito, añade un ID único (ID = 137) y me indica la fecha y hora en la que ha sido creado.

- https://reqres.in/api/users/2 name=morpheus job="zion_resident"

  Llamada PUT.
  Modifica completamente el usuario con ID = 2 con los nuevos datos indicados.
  Realizo petición: curl -X PUT https://reqres.in/api/users/2 -d "name=morpheus&job=zion_resident".
  La respuesta es un formato JSON con los nuevos datos que confirma la actualización. Añade fecha y hora de la misma.

- https://reqres.in/api/users/2 name=morpheus job="zion resident"

  Llamada PATCH.
  Actualiza de forma parcial los datos del usuario con ID 2.
  Realizo petición curl -X PATCH https://reqres.in/api/users/2 -d "name=morpheus&job=zion resident"
  La respuesta es tipo JSON, me indica la actualización del usuario ID 2, así como la fecha de la actualización.

- https://reqres.in/api/users/2

  Llamada DELETE.
  Elimina al usuario con ID 2.
  Realizo petición curl -X DELETE https://reqres.in/api/users/2.
  Como respuesta aparece código 204 No Content, lo que indica que la eliminación fue exitosa.

- https://reqres.in/api/register email=eve.holt@reqres.in password=pistol

  Llamada POST.
  Registra un usuario con email y contraseña proporcionados.
  Realizo petición: curl -X POST https://reqres.in/api/register -d "email=eve.holt@reqres.in&password=pistol"
  Respuesta en JSON con un ID único y un token para futuras autentificaciones.

- https://reqres.in/api/register email=sydney@fife

  Llamada POST.
  Registra un usuario con un email sin contraseña, debería dar error.
  Realizo petición: curl -X POST https://reqres.in/api/register -d "email=sydney@fife"
  La respuesta indica que falta un campo necesario: la contraseña. En formato JSON. Código 400 Bad Request.

- https://reqres.in/api/login email=eve.holt@reqres.in password=citylicka

  Llamada POST.
  Inicia sesión con usuario y contraseña especificados.
  Realizo petición: curl -X POST https://reqres.in/api/login -d "email=eve.holt@reqres.in&password=citylicka"
  La respuesta contiene un token de autenticacion para futuras solicitudes del usuario.

- https://reqres.in/api/login email=peter@klaven

  Llamada POST.
  Realizo petición: curl -X POST https://reqres.in/api/register -d "email=peter@klaven"
  Inicia sesión con el correo proporcionado, no se proporcionó contraseña por lo tanto debería de responder con un error.
  La respuesta indica un error: missing password. Estado de la respuesta: 400 Bad Request.

- https://reqres.in/api/users?delay=3

  Llamada GET.
  Solicita una lista de usuarios. Se agrega un parámetro, delay=3, que simula un retraso en la respuesta del servidor de 3 segundos.
  Tras un retraso de 3 segundos la respuesta contiene un objeto JSON que incluye información sobre la paginación y una lista de usuarios.

Tras realizar estas llamadas podemos concluir:

- **Métodos GET:** Se observaron respuestas exitosas al obtener usuarios específicos y listas de usuarios, lo que confirma la correcta funcionalidad de la API. En algunos casos, como la llamada a un usuario inexistente, se recibió una respuesta vacía, lo que sugiere que la API no siempre proporciona un error 404, sino que puede devolver un objeto vacío.
- **Métodos POST y PUT:** Las solicitudes para crear y actualizar usuarios proporcionaron respuestas satisfactorias, incluyendo la confirmación de la creación y la actualización de datos, evidenciando la operativa eficiente de la API para manipular recursos.
- **Método PATCH:** Se logró actualizar parcialmente la información de un usuario con éxito, indicando que la API admite cambios en campos específicos sin necesidad de enviar todos los datos del recurso.
- **Método DELETE:** La eliminación de un usuario fue confirmada sin errores, lo que demuestra que la API puede gestionar la eliminación de recursos de manera efectiva.
- **Simulación de latencia:** Al añadir el parámetro de retraso, se pudo comprobar cómo la API se comporta bajo condiciones de latencia, lo que es crucial para entender el rendimiento de la aplicación en entornos de red variables.
- **Manejo de errores:** Se evidenció que algunas solicitudes, como las de registro y login sin los parámetros necesarios, generan errores que son manejados adecuadamente por la API, ofreciendo retroalimentación sobre lo que falta.

La API de ReqRes es una herramienta útil y eficiente. Las respuestas obtenidas en cada llamada demuestran su robustez y capacidad para manejar diversas situaciones. La experiencia adquirida a través de estas pruebas me ha proporcionado conocimientos para entender cómo interactuar con APIs en el desarrollo de aplicaciones.