

Informe de Pruebas de la API de Reqres

Alberto van Oldenbarneveld

20 de septiembre de 2024

Índice

1. Introducción	2
2. Detalles de las Llamadas a la API	2
2.1. 1. Obtener la lista de usuarios (Página 2)	2
2.2. 2. Obtener un usuario específico (ID: 2)	2
2.3. 3. Obtener un usuario que no existe (ID: 23)	3
2.4. 4. Obtener la lista de recursos	3
2.5. 5. Obtener un recurso específico (ID: 2)	4
2.6. 6. Obtener un recurso que no existe (ID: 23)	5
2.7. 7. Crear un usuario	5
2.8. 8. Actualizar un usuario completo (PUT) (ID: 2)	6
2.9. 9. Actualizar parcialmente un usuario (PATCH) (ID: 2)	7
2.10. 10. Eliminar un usuario (DELETE) (ID: 2)	7
2.11. 11. Registro exitoso de un usuario	8
2.12. 12. Registro no exitoso de un usuario	8
2.13. 13. Inicio de sesión exitoso	9
2.14. 14. Inicio de sesión no exitoso	9
2.15. 15. Obtener usuarios con respuesta retrasada (3 segundos)	10
3. Conclusiones	11

1. Introducción

Este informe documenta las pruebas realizadas sobre la API de Reqres. Se han ejecutado 15 llamadas diferentes utilizando la herramienta cURL para verificar el funcionamiento de diversas operaciones, incluyendo la creación, obtención, actualización y eliminación de recursos, así como registros e inicios de sesión exitosos y no exitosos.

2. Detalles de las Llamadas a la API

2.1. 1. Obtener la lista de usuarios (Página 2)

Comando:

```
curl -X GET "https://reqres.in/api/users?page=2"
```

Comentario:

Esta solicitud obtiene la segunda página de la lista de usuarios. La respuesta incluye información sobre la paginación y los detalles de los usuarios en esa página.

Respuesta:

```
{
  "page":2,
  "per_page":6,
  "total":12,
  "total_pages":2,
  "data":[
    {"id":7,"email":"michael.lawson@reqres.in","first_name":"Michael","last_name":"Lawson","avatar":
    ↪ "https://reqres.in/img/faces/7-image.jpg"},
    {"id":8,"email":"lindsay.ferguson@reqres.in","first_name":"Lindsay","last_name":"Ferguson","avatar":
    ↪ "https://reqres.in/img/faces/8-image.jpg"},
    {"id":9,"email":"tobias.funke@reqres.in","first_name":"Tobias","last_name":"Funke","avatar":
    ↪ "https://reqres.in/img/faces/9-image.jpg"},
    {"id":10,"email":"byron.fields@reqres.in","first_name":"Byron","last_name":"Fields","avatar":
    ↪ "https://reqres.in/img/faces/10-image.jpg"},
    {"id":11,"email":"george.edwards@reqres.in","first_name":"George","last_name":"Edwards","avatar":
    ↪ "https://reqres.in/img/faces/11-image.jpg"},
    {"id":12,"email":"rachel.howell@reqres.in","first_name":"Rachel","last_name":"Howell","avatar":
    ↪ "https://reqres.in/img/faces/12-image.jpg"}
  ],
  "support":{
    "url":"https://reqres.in/#support-heading",
    "text":"To keep ReqRes free, contributions towards server costs are appreciated!"
  }
}
```

2.2. 2. Obtener un usuario específico (ID: 2)

Comando:

```
curl -X GET "https://reqres.in/api/users/2"
```

Comentario:

Esta solicitud obtiene los detalles del usuario con ID 2. La respuesta incluye la información personal del usuario solicitado.

Respuesta:

```
{
  "data":{
    "id":2,
    "email":"janet.weaver@reqres.in",
    "first_name":"Janet",
    "last_name":"Weaver",
    "avatar":"https://reqres.in/img/faces/2-image.jpg"
  },
  "support":{
    "url":"https://reqres.in/#support-heading",
    "text":"To keep ReqRes free, contributions towards server costs are appreciated!"
  }
}
```

2.3. 3. Obtener un usuario que no existe (ID: 23)

Comando:

```
curl -X GET "https://reqres.in/api/users/23"
```

Comentario:

Esta solicitud intenta obtener los detalles de un usuario con ID 23, el cual no existe en la base de datos de Reqres. Se espera una respuesta vacía.

Respuesta:

```
{}
```

2.4. 4. Obtener la lista de recursos

Comando:

```
curl -X GET "https://reqres.in/api/unknown"
```

Comentario:

Esta solicitud obtiene una lista de recursos disponibles. La respuesta incluye detalles sobre cada recurso, como nombre, año, color y valor Pantone.

Respuesta:

```
{
  "page":1,
  "per_page":6,
  "total":12,
  "total_pages":2,
  "data":[
    {"id":1,"name":"cerulean","year":2000,"color":"#98B2D1","pantone_value":"15-4020"},
    {"id":2,"name":"fuchsia rose","year":2001,"color":"#C74375","pantone_value":"17-2031"},
    {"id":3,"name":"true red","year":2002,"color":"#BF1932","pantone_value":"19-1664"},
    {"id":4,"name":"aqua sky","year":2003,"color":"#7BC4C4","pantone_value":"14-4811"},
    {"id":5,"name":"tigerlily","year":2004,"color":"#E2583E","pantone_value":"17-1456"},
    {"id":6,"name":"blue turquoise","year":2005,"color":"#53B0AE","pantone_value":"15-5217"}
  ],
  "support":{
    "url":"https://reqres.in/#support-heading",
    "text":"To keep ReqRes free, contributions towards server costs are appreciated!"
  }
}
```

2.5. 5. Obtener un recurso específico (ID: 2)

Comando:

```
curl -X GET "https://reqres.in/api/unknown/2"
```

Comentario:

Esta solicitud obtiene los detalles del recurso con ID 2. La respuesta proporciona información específica sobre dicho recurso.

Respuesta:

```
{
  "data":{
    "id":2,
    "name":"fuchsia rose",
    "year":2001,
    "color":"#C74375",
    "pantone_value":"17-2031"
  },
  "support":{
    "url":"https://reqres.in/#support-heading",
    "text":"To keep ReqRes free, contributions towards server costs are appreciated!"
  }
}
```

2.6. 6. Obtener un recurso que no existe (ID: 23)

Comando:

```
curl -X GET "https://reqres.in/api/unknown/23"
```

Comentario:

Esta solicitud intenta obtener los detalles de un recurso con ID 23, el cual no existe. Se espera una respuesta vacía.

Respuesta:

```
{}
```

2.7. 7. Crear un usuario

Comando:

```
curl -X POST "https://reqres.in/api/users" \
  -H "Content-Type: application/json" \
  -d '{
    "name": "morpheus",
    "job": "leader"
  }'
```

Comentario:

Esta solicitud crea un nuevo usuario con el nombre “morpheus” y el puesto "leader". La respuesta confirma la creación del usuario con un ID y la fecha de creación.

Respuesta:

```
{
  "name": "John Doe",
  "job": "Software Developer",
  "id": "736",
  "createdAt": "2024-09-20T11:23:58.289Z"
}
```

2.8. 8. Actualizar un usuario completo (PUT) (ID: 2)

Comando:

```
curl -X PUT "https://reqres.in/api/users/2" \
  -H "Content-Type: application/json" \
  -d '{
    "name": "morpheus",
    "job": "zion resident"
  }'
```

Comentario:

Esta solicitud actualiza completamente el usuario con ID 2, cambiando su nombre a "morpheus" y su puesto a "zion resident". La respuesta confirma la actualización con la fecha correspondiente.

Respuesta:

```
{
  "name": "morpheus",
  "job": "zion resident",
  "updatedAt": "2024-09-20T11:24:02.788Z"
}
```

2.9. 9. Actualizar parcialmente un usuario (PATCH) (ID: 2)

Comando:

```
curl -X PATCH "https://reqres.in/api/users/2" \
  -H "Content-Type: application/json" \
  -d '{
    "job": "Senior Project Manager"
  }'
```

Comentario:

Request

/api/users/2

```
{
  "name": "morpheus",
  "job": "Senior Project Manager"
}
```

Response

200

```
{
  "name": "morpheus",
  "job": "Senior Project Manager",
  "updatedAt": "2024-09-20T17:57:39.059Z"
}
```

2.10. 10. Eliminar un usuario (DELETE) (ID: 2)

Comando:

```
curl -X DELETE "https://reqres.in/api/users/2"
```

Comentario:

Esta solicitud elimina el usuario con ID 2. La respuesta no contiene contenido, lo que indica que la eliminación fue exitosa.

Respuesta: (No hay contenido en la respuesta)

2.11. 11. Registro exitoso de un usuario

Comando:

```
curl -X POST "https://reqres.in/api/register" \
-H "Content-Type: application/json" \
-d '{
  "email": "eve.holt@reqres.in",
  "password": "pistol"
}'
```

Comentario:

Esta solicitud registra exitosamente un usuario proporcionando el correo electrónico y la contraseña. La respuesta incluye un ID y un token de autenticación.

Respuesta:

```
{
  "id": 4,
  "token": "QpwL5tke4Pnpja7X4"
}
```

2.12. 12. Registro no exitoso de un usuario

Comando:


```
curl -X POST "https://reqres.in/api/register" \
  -H "Content-Type: application/json" \
  -d '{
    "email": "sydney@fife"
  },'
```

Comentario:

Esta solicitud intenta registrar un usuario sin proporcionar la contraseña, lo que resulta en un error. La respuesta indica que falta la contraseña.

Respuesta:

```
{
  "error": "Missing password"
}
```

2.13. 13. Inicio de sesión exitoso

Comando:

```
curl -X POST "https://reqres.in/api/login" \
  -H "Content-Type: application/json" \
  -d '{
    "email": "eve.holt@reqres.in",
    "password": "cityslicka"
  },'
```

Comentario:

Esta solicitud inicia sesión exitosamente proporcionando el correo electrónico y la contraseña correctos. La respuesta incluye un token de autenticación.

Respuesta:

```
{
  "token": "QpwL5tke4Pnpja7X4"
}
```

2.14. 14. Inicio de sesión no exitoso

Comando:

```
curl -X POST "https://reqres.in/api/login" \
  -H "Content-Type: application/json" \
  -d '{
    "email": "peter@klaven"
  },'
```

Comentario:

Esta solicitud intenta iniciar sesión sin proporcionar la contraseña, lo que resulta en un error. La respuesta indica que falta la contraseña.

Respuesta:

```
{
  "error": "Missing password"
}
```

2.15. 15. Obtener usuarios con respuesta retrasada (3 segundos)

Comando:

```
curl -X GET "https://reqres.in/api/users?delay=3"
```

Comentario:

Esta solicitud obtiene la lista de usuarios con una demora de respuesta de 3 segundos, simulando una respuesta lenta del servidor. La respuesta incluye la lista de usuarios como en las solicitudes anteriores.

Respuesta:

```
{
  "page": 1,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 1,
      "email": "george.bluth@reqres.in",
      "first_name": "George",
      "last_name": "Bluth",
      "avatar": "https://reqres.in/img/faces/1-image.jpg"
    },
    {
      "id": 2,
      "email": "janet.weaver@reqres.in",
      "first_name": "Janet",
      "last_name": "Weaver",
      "avatar": "https://reqres.in/img/faces/2-image.jpg"
    },
    {
      "id": 3,
      "email": "emma.wong@reqres.in",
      "first_name": "Emma",
      "last_name": "Wong",
      "avatar": "https://reqres.in/img/faces/3-image.jpg"
    },
    {
      "id": 4,
      "email": "eve.holt@reqres.in",
      "first_name": "Eve",
      "last_name": "Holt",
      "avatar": "https://reqres.in/img/faces/4-image.jpg"
    },
    {
      "id": 5,
      "email": "charles.morris@reqres.in",
      "first_name": "Charles",
      "last_name": "Morris",
      "avatar": "https://reqres.in/img/faces/5-image.jpg"
    },
    {
      "id": 6,
      "email": "tracey.ramos@reqres.in",
      "first_name": "Tracey",
      "last_name": "Ramos",
      "avatar": "https://reqres.in/img/faces/6-image.jpg"
    }
  ],
  "support": {
    "url": "https://reqres.in/#support-heading",
    "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
  }
}
```

3. Conclusiones

Después de realizar las 15 pruebas a la API de Reqres, se ha comprobado que funciona de manera consistente y según lo esperado. Las solicitudes para obtener usuarios y recursos existentes devolvieron los datos correctos, mientras que las pruebas con IDs inexistentes manejaron adecuadamente los errores.

Las operaciones de crear, actualizar y eliminar usuarios también respondieron correctamente, proporcionando confirmaciones claras de cada acción. Además, las pruebas de registro e inicio de sesión demostraron que la API gestiona bien tanto los casos exitosos como los errores cuando faltan datos necesarios, como las contraseñas.

La prueba con una respuesta retrasada de 3 segundos fue útil para observar cómo la API maneja situaciones de latencia, lo cual es importante para entender su rendimiento en diferentes condiciones.

En resumen, la API de Reqres es una herramienta útil para realizar pruebas y desarrollar aplicaciones que requieran interacción con APIs. Su comportamiento predecible y las respuestas claras facilitan su integración y uso en futuros proyectos.