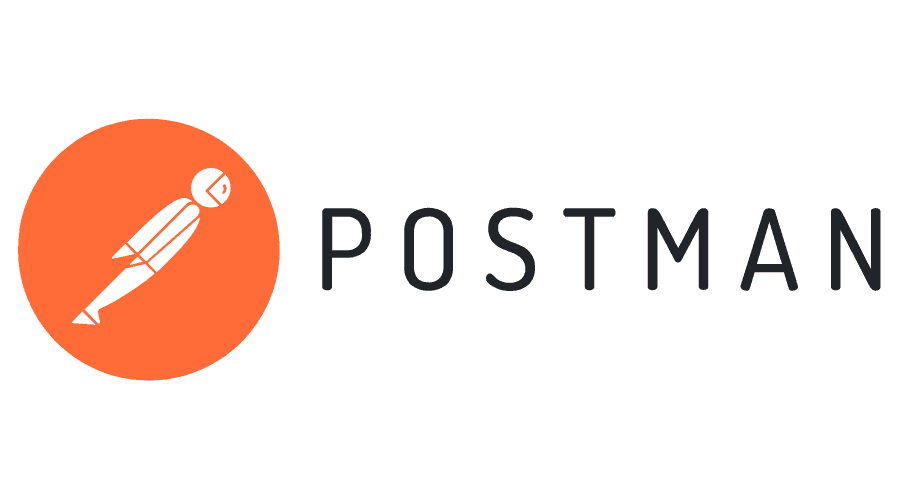
Documentación Postman



Javier Meléndez Contreras

Carlos García Acevedo

Andrés Masset Ruiz

**1º GET:** Vemos que la petición enviada es de tipo GET y que accede a la siguiente dirección de la API: https://itunes.apple.com/lookup?upc=720642462928&entity=song

La primera parte, https://itunes.apple.com/, es una variable de entorno que guarda la URL verdadera de la API. Desde esta vista, aparte de ver las variables de entorno existentes, se puede modificar el nombre del entorno, crear nuevas variables, modificarlas y borrarlas.

**Authorization** sirve para introducir el tipo de autorización que usa la API, si hiciera uso de alguna. No es nuestro caso, por lo que no entraremos en más detalles.

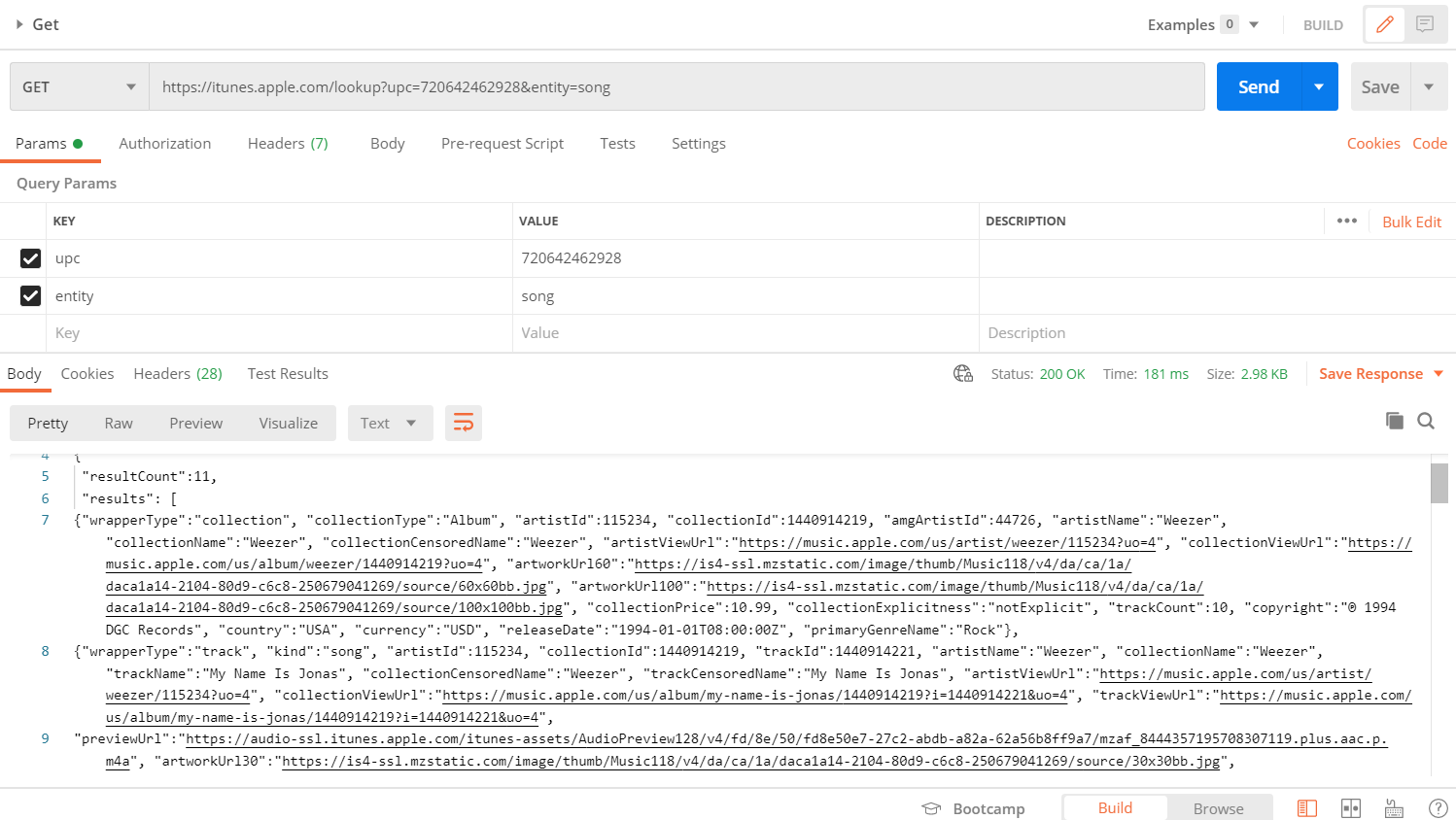
**Headers**, donde se deben introducir las cabeceras de la petición. Para este ejemplo usamos las 7 cabeceras que podemos observar

**Body** muestra los datos que se mandan con la petición. En este caso, no nos devuelve ningún dato.

**Pre-request Script**, al recibir una respuesta vamos a querer hacer al menos 2 cosas. La primera, hacer una comprobación de que la respuesta es la esperada y, la segunda, guardar el resultado de la respuesta para ser usado en nuevas peticiones.

**Tests**. En ella, podemos hacer operaciones con los resultados de la respuesta. Se muestran las comprobaciones realizadas y el resultado de las mismas que, en este caso, todas han sido las esperadas.

**Setting** intenta minimizar la cantidad de configuraciones que tiene que cambiar, por lo que algunos valores predeterminados se establecen automáticamente. Puede personalizar su configuración según su caso de uso.



**2º GET:** Vemos que la petición enviada es de tipo GET y que accede a la siguiente dirección de la API: https://api.lyrics.ovh/v1/Coldplay/Adventure of a Lifetime

La primera parte, https://api.lyrics.ovh, es una variable de entorno que guarda la URL verdadera de la API. Desde esta vista, aparte de ver las variables de entorno existentes, se puede modificar el nombre del entorno, crear nuevas variables, modificarlas y borrarlas.

**Authorization** sirve para introducir el tipo de autorización que usa la API, si hiciera uso de alguna. No es nuestro caso, por lo que no entraremos en más detalles.

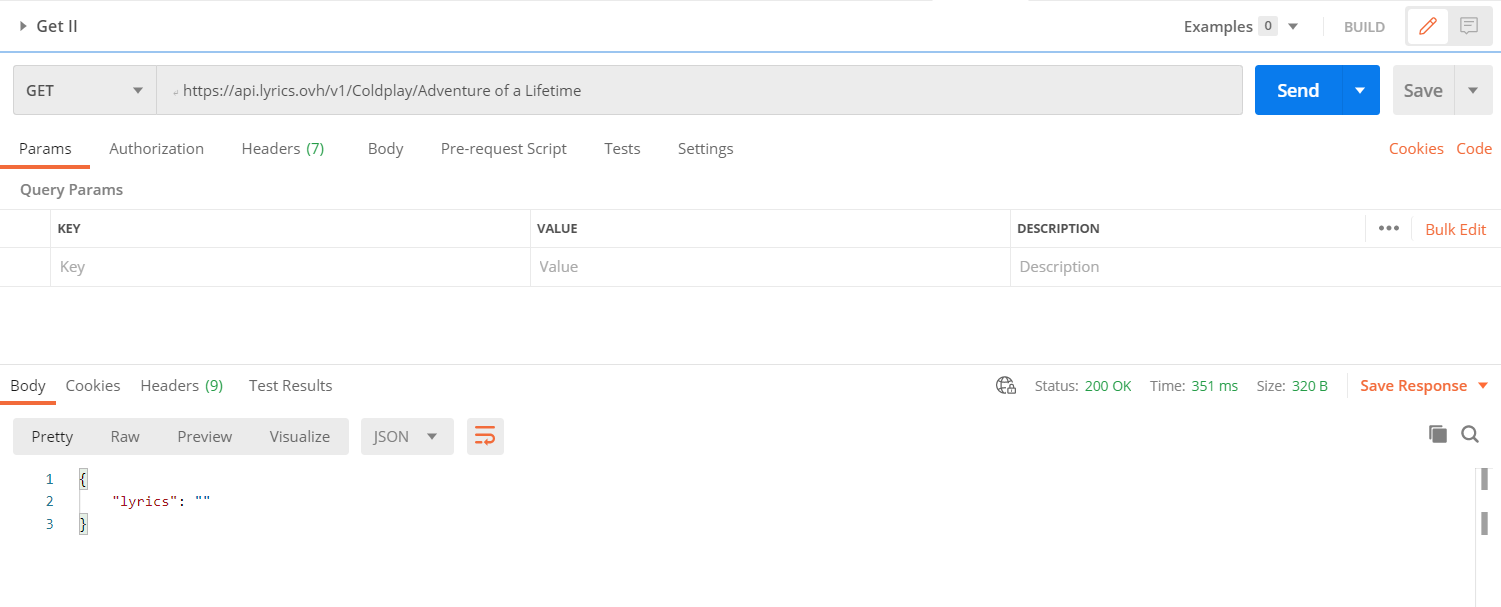
**Headers**, donde se deben introducir las cabeceras de la petición. Para este ejemplo usamos las 7 cabeceras que podemos observar

**Body** muestra los datos que se mandan con la petición. En este caso, no nos devuelve ningún dato.

**Pre-request Script**, al recibir una respuesta vamos a querer hacer al menos 2 cosas. La primera, hacer una comprobación de que la respuesta es la esperada y, la segunda, guardar el resultado de la respuesta para ser usado en nuevas peticiones.

**Tests**. En ella, podemos hacer operaciones con los resultados de la respuesta. Se muestran las comprobaciones realizadas y el resultado de las mismas que, en este caso, todas han sido las esperadas.

**Setting** intenta minimizar la cantidad de configuraciones que tiene que cambiar, por lo que algunos valores predeterminados se establecen automáticamente. Puede personalizar su configuración según su caso de uso.



**1º POST:** Vemos que la petición enviada es de tipo POST y que accede a la siguiente dirección de la API:

https://itunes.apple.com/lookup?amgAlbumId=15175,15176,15177,15178,15183,15184,15187,1519,15191,15195,15197,15198

La primera parte, https://itunes.apple.com, es una variable de entorno que guarda la URL verdadera de la API. Desde esta vista, aparte de ver las variables de entorno existentes, se puede modificar el nombre del entorno, crear nuevas variables, modificarlas y borrarlas.

Luego haríamos una petición de "lookup" para que se muestre lo que queremos a continuación, con "amgAlbumId " pedimos la ID del grupo o artista, con los siguientes números 15175,15176,15177,15178,15183,15184, queremos que nos muestre wrapperType":"collection", "collectionType":"Album", "artistId":2541832, "collectionId":1459409129, "amgArtistId":5141, "artistName":"Pianosaurus".

Para los siguientes 6 números 15187,1519,15191,15195,15197,15198, nos muestra otro tipo de colección, de albúm, de la Id de artista y de colección, y por último, otro nombre de artista.

En el Request Params tenemos 1 "key" con las que identificamos todo lo visto anteriormente.

**Authorization** sirve para introducir el tipo de autorización que usa la API, si hiciera uso de alguna. No es nuestro caso, por lo que no entraremos en más detalles.

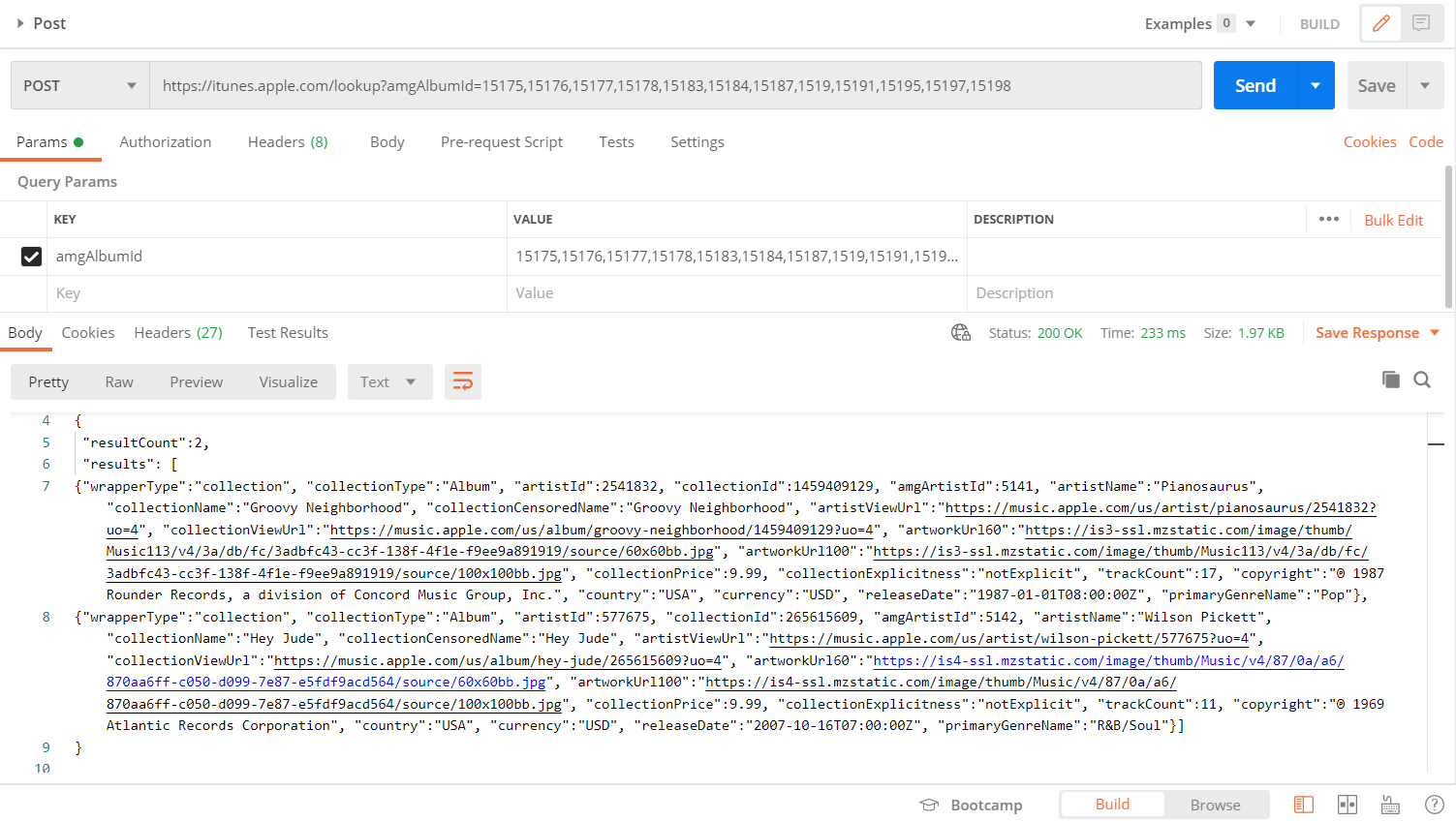
**Headers**, donde se deben introducir las cabeceras de la petición. Para este ejemplo usamos las 8 cabeceras que podemos observar.

**Body** muestra los datos que se mandan con la petición. En este caso, no nos devuelve ningún dato.

**Pre-request Script**, al recibir una respuesta vamos a querer hacer al menos 2 cosas. La primera, hacer una comprobación de que la respuesta es la esperada y, la segunda, guardar el resultado de la respuesta para ser usado en nuevas peticiones.

**Tests**. En ella, podemos hacer operaciones con los resultados de la respuesta. Se muestran las comprobaciones realizadas y el resultado de las mismas que, en este caso, todas han sido las esperadas.

**Setting** intenta minimizar la cantidad de configuraciones que tiene que cambiar, por lo que algunos valores predeterminados se establecen automáticamente. Puede personalizar su configuración según su caso de uso.



**2º POST:** Vemos que la petición enviada es de tipo POST y que accede a la siguiente dirección de la API:

https://itunes.apple.com/lookup?amgArtistId=468749,5723&entity=album&limit=5

La primera parte, https://itunes.apple.com, es una variable de entorno que guarda la URL verdadera de la API. Desde esta vista, aparte de ver las variables de entorno existentes, se puede modificar el nombre del entorno, crear nuevas variables, modificarlas y borrarlas.

Luego haríamos una petición de "lookup" para que se muestre lo que queremos a continuación, con "amgArtistId" pedimos la ID del grupo o artista, con "entity" lo que queremos es que nos muestre sus álbumes, pero no todos, porque al poner un "limit" a 5 queremos que solo muestre los 5 primeros álbumes.

En el Request Params tenemos 3 "key" con las que identificamos el ID del grupo o artista, el álbum y que nos muestre solo 5 de esos álbumes.

**Authorization** sirve para introducir el tipo de autorización que usa la API, si hiciera uso de alguna. No es nuestro caso, por lo que no entraremos en más detalles.

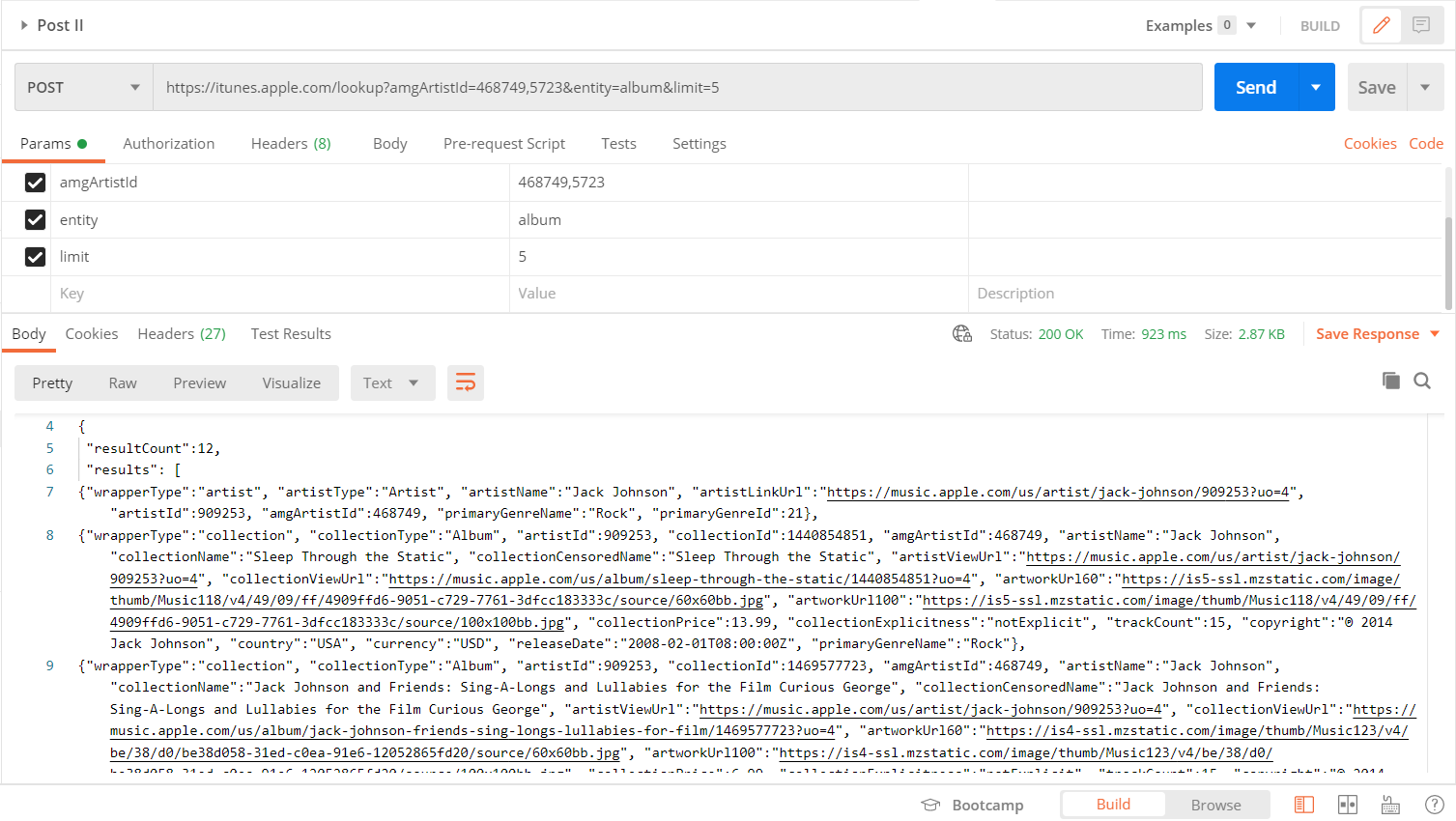
**Headers**, donde se deben introducir las cabeceras de la petición. Para este ejemplo usamos las 7 cabeceras que podemos observar.

**Body** muestra los datos que se mandan con la petición. En este caso, no nos devuelve ningún dato.

**Pre-request Script**, al recibir una respuesta vamos a querer hacer al menos 2 cosas. La primera, hacer una comprobación de que la respuesta es la esperada y, la segunda, guardar el resultado de la respuesta para ser usado en nuevas peticiones.

**Tests**. En ella, podemos hacer operaciones con los resultados de la respuesta. Se muestran las comprobaciones realizadas y el resultado de las mismas que, en este caso, todas han sido las esperadas.

**Setting** intenta minimizar la cantidad de configuraciones que tiene que cambiar, por lo que algunos valores predeterminados se establecen automáticamente. Puede personalizar su configuración según su caso de uso.



**1º DEL:** Vemos que la petición enviada es de tipo DEL y que accede a la siguiente dirección de la API: https://graph.facebook.com/v2.11/10219439247352479

La primera parte, https://graph.facebook.com es una variable de entorno que guarda la URL verdadera de la API. Desde esta vista, aparte de ver las variables de entorno existentes, se puede modificar el nombre del entorno, crear nuevas variables, modificarlas y borrarlas. Luego "v2.11 " para buscar en ese repositorio una publicacion de un usuario " con la ID " 10219439247352479 ".

**Authorization** sirve para introducir el tipo de autorización que usa la API, si hiciera uso de alguna. En nuestro caso da error, porque no tenemos los permisos de dicho usuario para poder borrar la publicación.

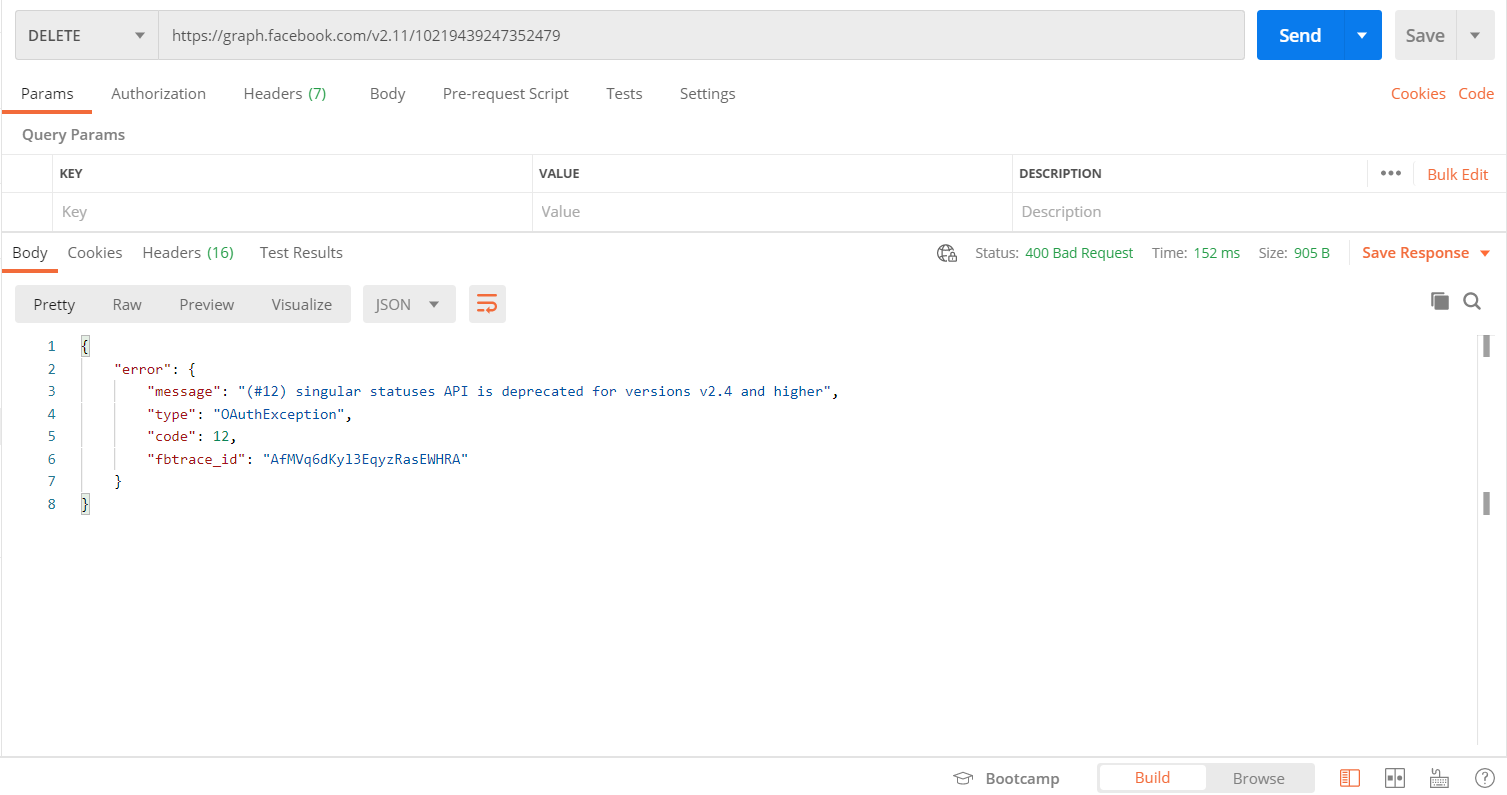
**Headers**, donde se deben introducir las cabeceras de la petición. Para este ejemplo usamos las 7 cabeceras que podemos observar.

**Body** muestra los datos que se mandan con la petición. En este caso, no nos devuelve ningún dato.

**Pre-request Script**, al recibir una respuesta vamos a querer hacer al menos 2 cosas. La primera, hacer una comprobación de que la respuesta es la esperada y, la segunda, guardar el resultado de la respuesta para ser usado en nuevas peticiones.

**Tests**. En ella, podemos hacer operaciones con los resultados de la respuesta. Se muestran las comprobaciones realizadas y el resultado de las mismas que, en este caso, todas han sido las esperadas.

**Setting** intenta minimizar la cantidad de configuraciones que tiene que cambiar, por lo que algunos valores predeterminados se establecen automáticamente. Puede personalizar su configuración según su caso de uso.

**2º DEL:** Vemos que la petición enviada es de tipo DEL y que accede a la siguiente dirección de la API: https://gorest.co.in/public-api/users/1158

La primera parte, https://gorest.co.in, es una variable de entorno que guarda la URL verdadera de la API. Desde esta vista, aparte de ver las variables de entorno existentes, se puede modificar el nombre del entorno, crear nuevas variables, modificarlas y borrarlas. Luego "public-api" para buscar en ese repositorio un usuario en "users" con la ID "1158".

**Authorization** sirve para introducir el tipo de autorización que usa la API, si hiciera uso de alguna. En nuestro caso da error, porque debemos iniciar sesión con dicho usuario para poder borrar el contenido.

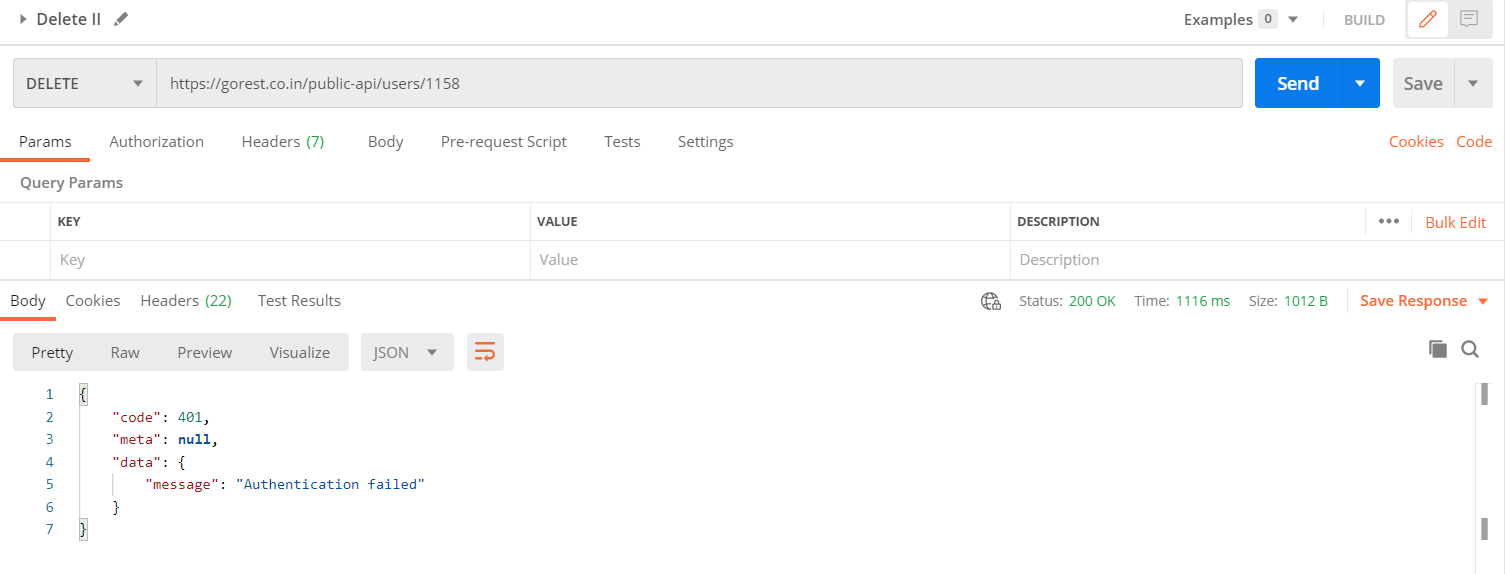
**Headers**, donde se deben introducir las cabeceras de la petición. Para este ejemplo usamos las 7 cabeceras que podemos observar

**Body** muestra los datos que se mandan con la petición. En este caso, no nos devuelve ningún dato.

**Pre-request Script**, al recibir una respuesta vamos a querer hacer al menos 2 cosas. La primera, hacer una comprobación de que la respuesta es la esperada y, la segunda, guardar el resultado de la respuesta para ser usado en nuevas peticiones.

**Tests**. En ella, podemos hacer operaciones con los resultados de la respuesta. Se muestran las comprobaciones realizadas y el resultado de las mismas que, en este caso, todas han sido las esperadas.

**Setting** intenta minimizar la cantidad de configuraciones que tiene que cambiar, por lo que algunos valores predeterminados se establecen automáticamente. Puede personalizar su configuración según su caso de uso.



**1º PUT:** Vemos que la petición enviada es de tipo PUT y que accede a la siguiente dirección de la API: https://www.twitter.com/users/123

La primera parte, https://www.twitter.com, es una variable de entorno que guarda la URL verdadera de la API. Desde esta vista, aparte de ver las variables de entorno existentes, se puede modificar el nombre del entorno, crear nuevas variables, modificarlas y borrarlas. Luego "public-api" para buscar en ese repositorio un usuario en "users" con la ID "123".

**Authorization** sirve para introducir el tipo de autorización que usa la API, si hiciera uso de alguna. En nuestro caso no da error, porque vemos como nos devuelve la ruta especificada por el usuario.

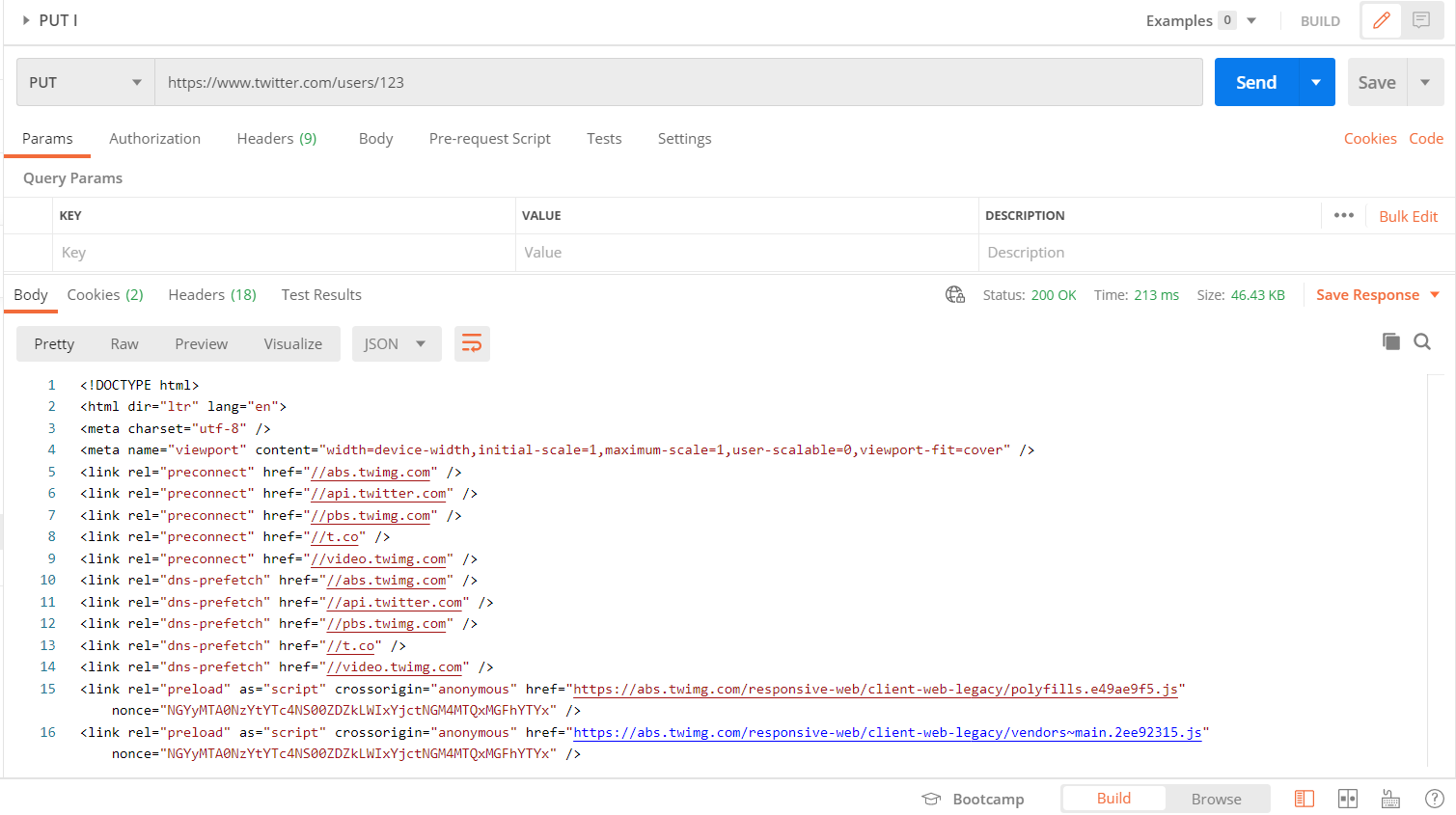
**Headers**, donde se deben introducir las cabeceras de la petición. Para este ejemplo usamos las 9 cabeceras que podemos observar.

**Body** muestra los datos que se mandan con la petición. En este caso, no nos devuelve ningún dato.

**Pre-request Script**, al recibir una respuesta vamos a querer hacer al menos 2 cosas. La primera, hacer una comprobación de que la respuesta es la esperada y, la segunda, guardar el resultado de la respuesta para ser usado en nuevas peticiones.

**Tests**. En ella, podemos hacer operaciones con los resultados de la respuesta. Se muestran las comprobaciones realizadas y el resultado de las mismas que, en este caso, todas han sido las esperadas.

**Setting** intenta minimizar la cantidad de configuraciones que tiene que cambiar, por lo que algunos valores predeterminados se establecen automáticamente. Puede personalizar su configuración según su caso de uso.



**2º PUT:** Vemos que la petición enviada es de tipo PUT y que accede a la siguiente dirección de la API: <https://gorest.co.in/public-api/users/1158>

La primera parte, https://gorest.co.in, es una variable de entorno que guarda la URL verdadera de la API. Desde esta vista, aparte de ver las variables de entorno existentes, se puede modificar el nombre del entorno, crear nuevas variables, modificarlas y borrarlas. Luego "public-api" para buscar en ese repositorio un usuario en "users" con la ID "1158".

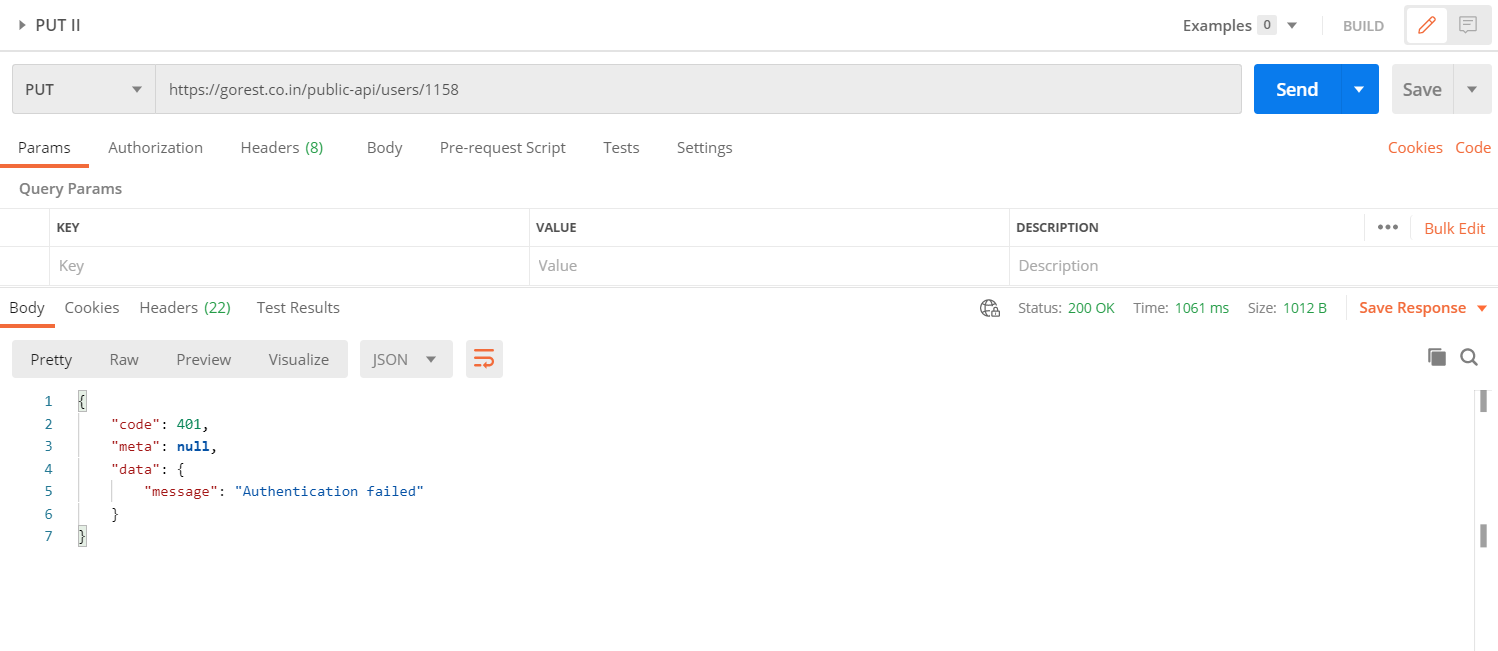
**Authorization** sirve para introducir el tipo de autorización que usa la API, si hiciera uso de alguna. En nuestro caso da error, porque debemos iniciar sesión con dicho usuario para poder borrar el contenido.

**Headers**, donde se deben introducir las cabeceras de la petición. Para este ejemplo usamos las 7 cabeceras que podemos observar

**Body** muestra los datos que se mandan con la petición. En este caso, no nos devuelve ningún dato.

**Pre-request Script**, al recibir una respuesta vamos a querer hacer al menos 2 cosas. La primera, hacer una comprobación de que la respuesta es la esperada y, la segunda, guardar el resultado de la respuesta para ser usado en nuevas peticiones.

**Tests**. En ella, podemos hacer operaciones con los resultados de la respuesta. Se muestran las comprobaciones realizadas y el resultado de las mismas que, en este caso, todas han sido las esperadas.

**Setting** intenta minimizar la cantidad de configuraciones que tiene que cambiar, por lo que algunos valores predeterminados se establecen automáticamente. Puede personalizar su configuración según su caso de uso.