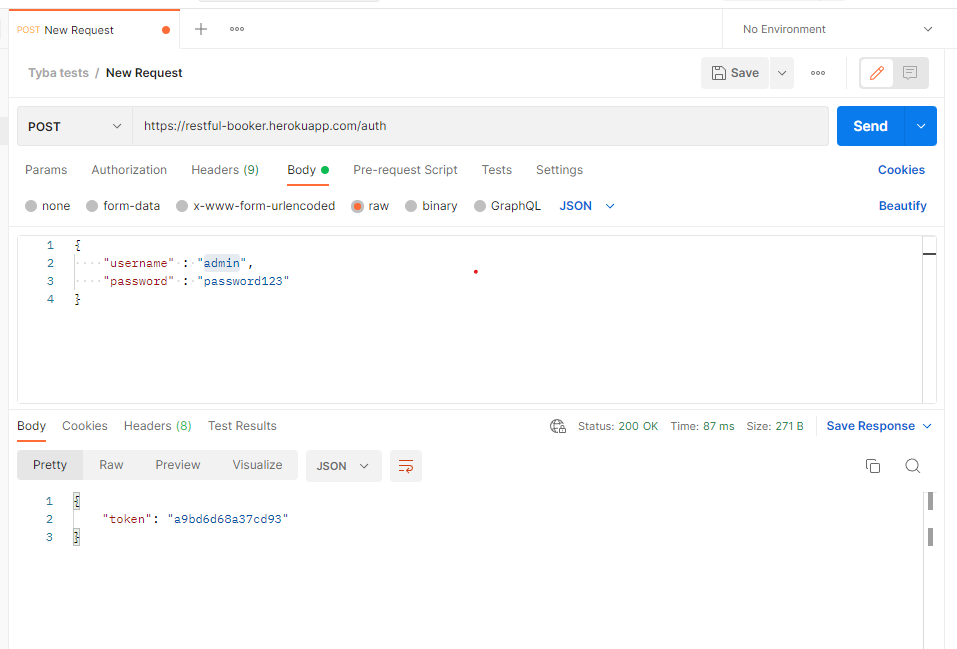
**Prueba de servicios Tyba**

1. Generar token

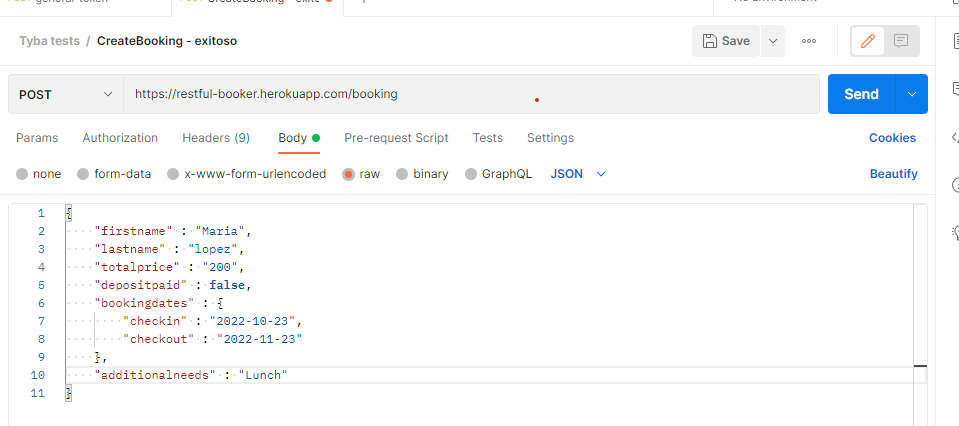


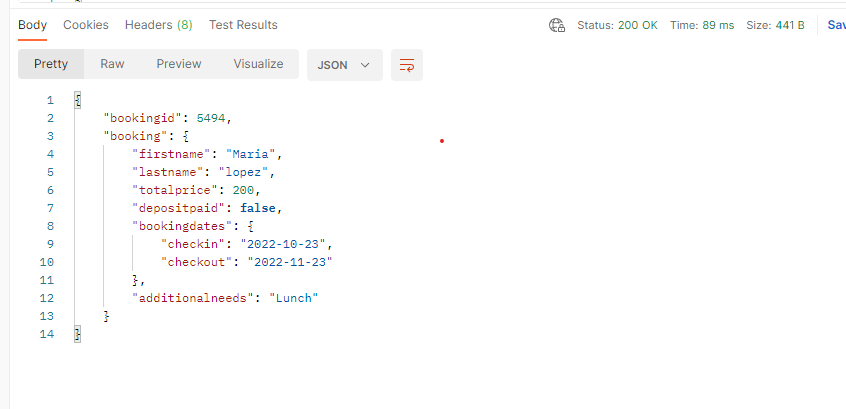
1. Crear reserva exitosamente

Dado que el usuario quiere realizar una reserva

Cuando consume el servicio con los datos: firstname, lastname, totalprice, depositpaid, checkin, checkout y aditionalneeds

Entonces la reserva se debe crear exitosamente. **Exitoso**

**Request** ****

**Response **

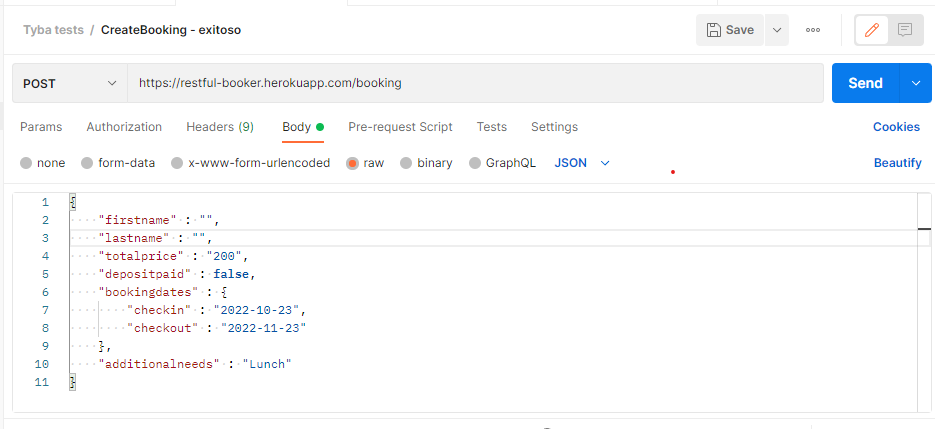
1. Crear reserva con los campos nombre ni apellido vacíos

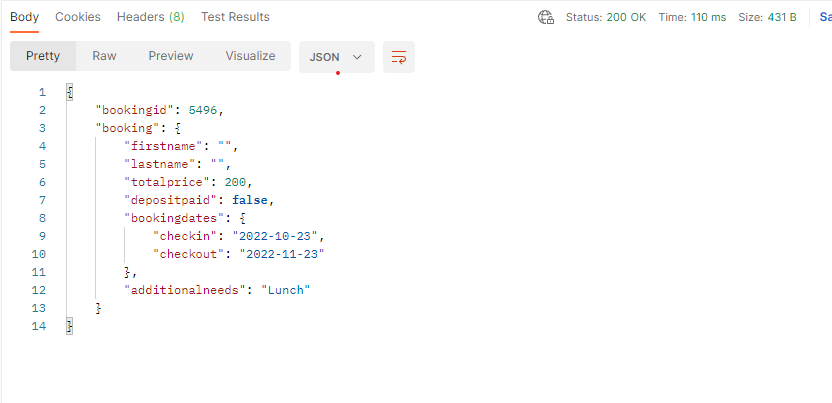
Dado que el usuario quiere realizar una reserva

Cuando consume el servicio con los datos: totalprice, depositpaid, checkin, checkout y aditionalneeds

Pero los campos nombre ni el apellido se envían vacíos

Entonces la reserva no se debería crear porque el nombre y apellido deben ser campos obligatorios. **Fallido**

**Request** 

**Response** 

1. Crear reserva sin precio total

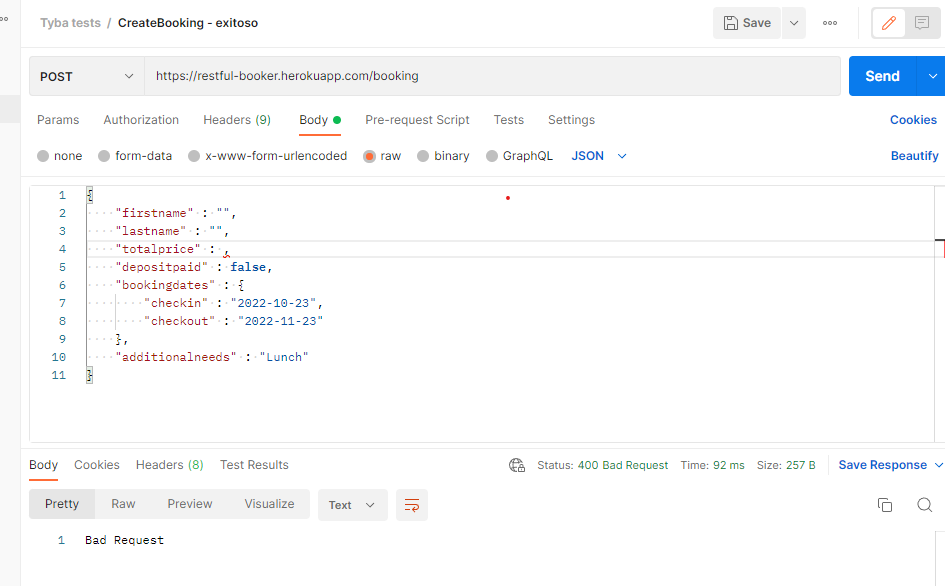
Dado que el usuario quiere realizar una reserva

Cuando consume el servicio con los datos: firstname, lastname, depositpaid, checkin, checkout y aditionalneeds

Pero no se ingresa el precio total

Entonces la reserva no se debería crear porque el precio debe ser un campo obligatorio. **Éxito**

**Request**  **y Response**

****

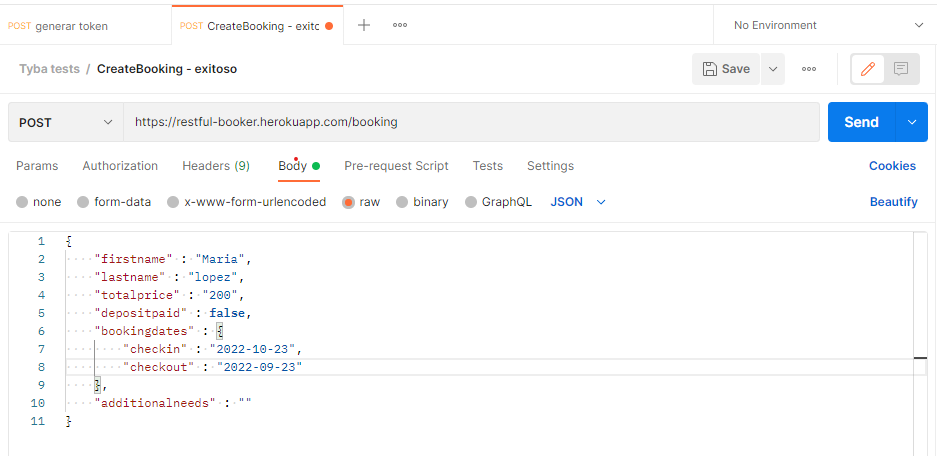
1. Crear reserva con fecha de checkout menor a fecha de checkin

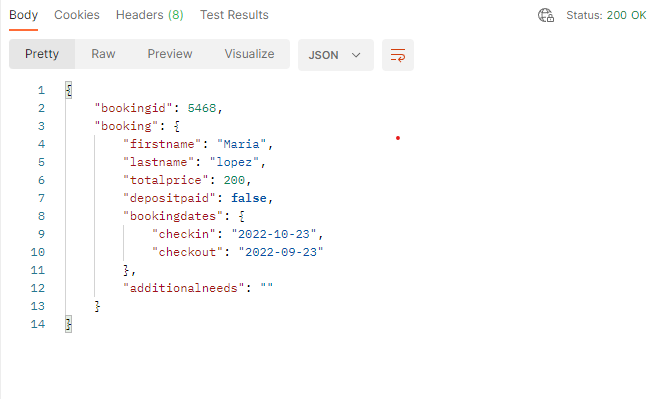
Dado que el usuario quiere realizar una reserva

Cuando consume el servicio con los datos: firstname, lastname, totalprecio, depositpaid, checkin, checkout, additionalneeds.

Pero se ingresa la fecha de checkin mayor a la fecha de checkout

Entonces la reserva no se debe poder crear porque la fecha de checkout no puede ser menor a la fecha de checkin. **Fallido**

**Request** ****

**Response** ****

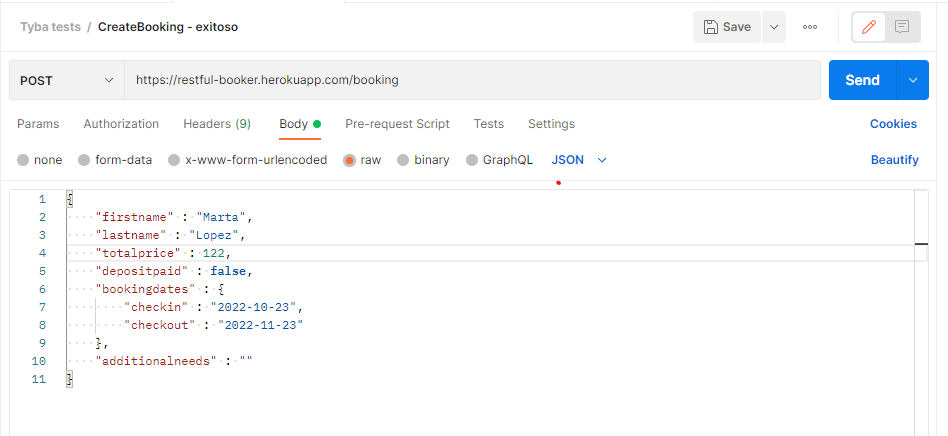
1. Crear reserva sin necesidades adicionales

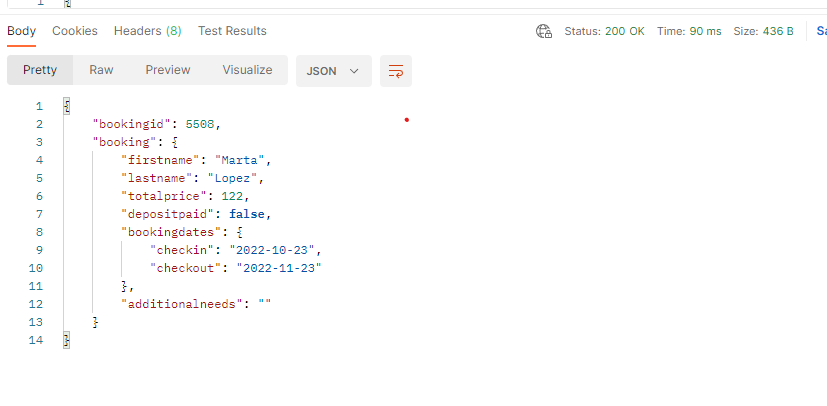
Dado que el usuario quiere realizar una reserva

Cuando consume el servicio con los datos: firstname, lastname, totalprecio, depositpaid, checkin, checkout.

Pero no se ingresan necesidades adicionales

Entonces la reserva se debe crear exitosamente porque el campo necesidades adicionales no es obligatorio. **Éxito**

**Request** 

**Response **

1. Crear reserva con fechas de checkin y checkout inválidas

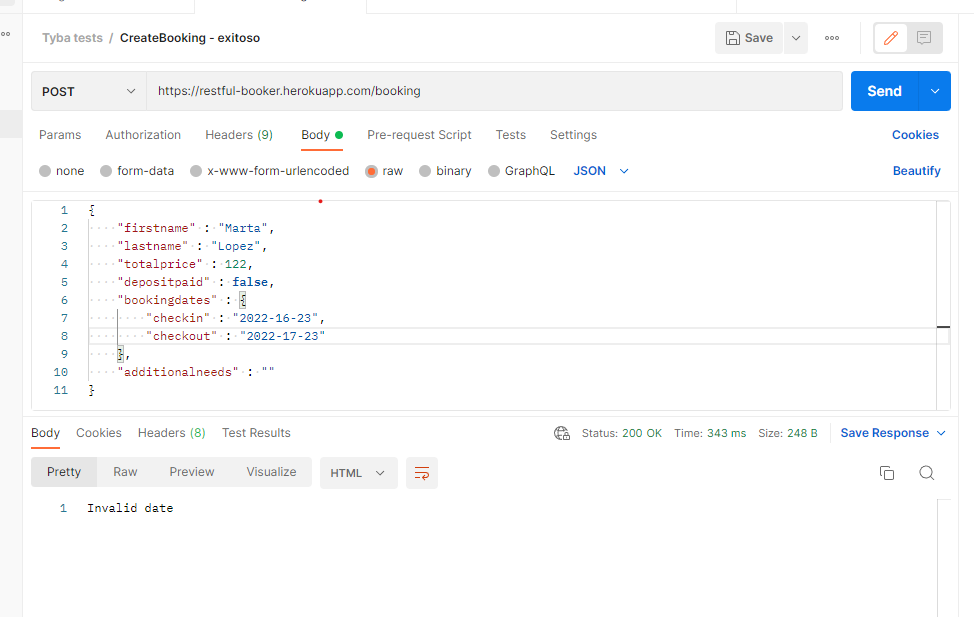
Dado que el usuario quiere realizar una reserva

Cuando consume el servicio con los datos: firstname, lastname, totalprecio, depositpaid, checkin, checkout.

Pero se ingresan fechas de checkin y checkout inválidas

Entonces la reserva no se debe crear. **Éxito**

**Request y Response**

****

1. Crear reserva con fechas de checkin y checkout que ya pasaron

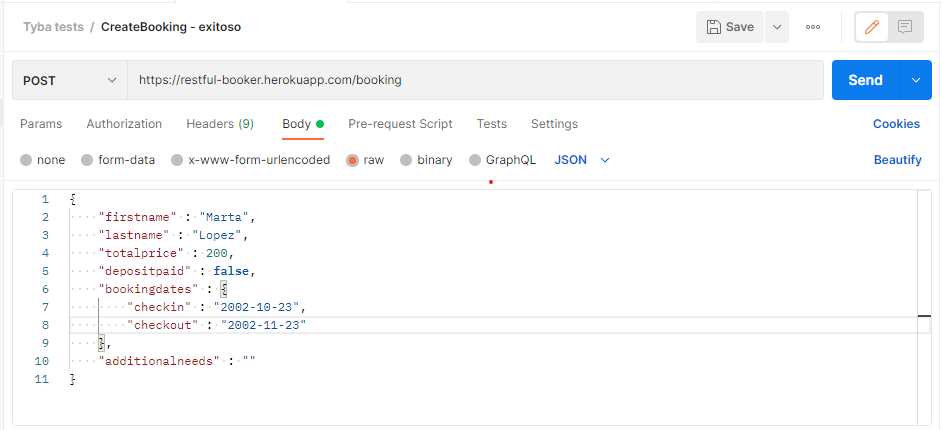
Dado que el usuario quiere realizar una reserva

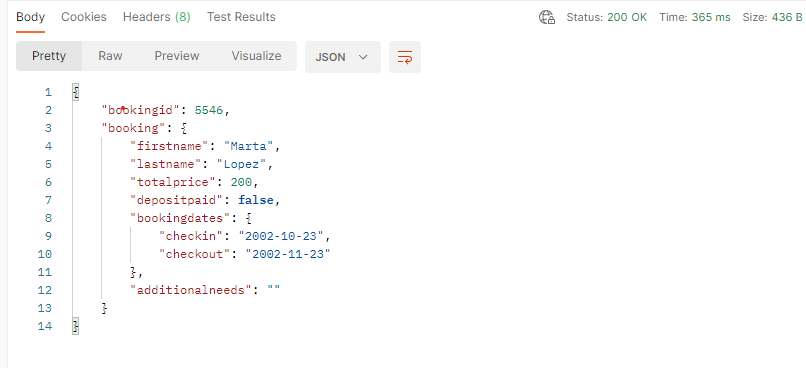
Cuando consume el servicio con los datos: firstname, lastname, totalprecio, depositpaid, checkin, checkout.

Pero se ingresan fechas de checkin y checkout que ya pasaron

Entonces la reserva no se debe crear. **Fallido**

**Request**

****

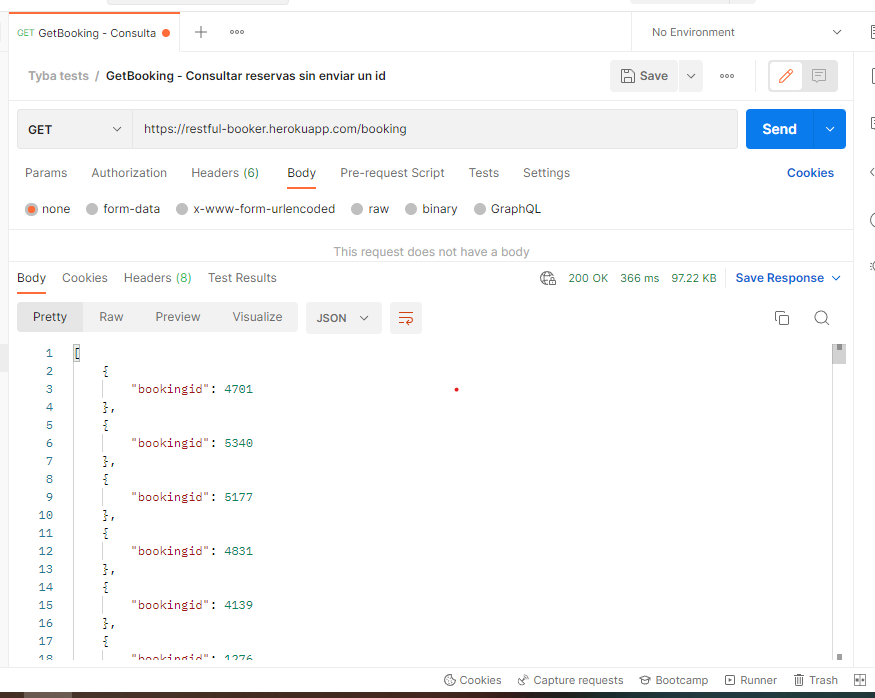
**Response** ****

1. Consultar reserva si ingresar id

Dado que el usuario quiere consultar las reservas existentes

Cuando consume el servicio sin un id específico

Entonces se deben obtener todas las reservas. **Fallido (**Porque está obteniendo solo el id de las reservas y no su detalle**)**

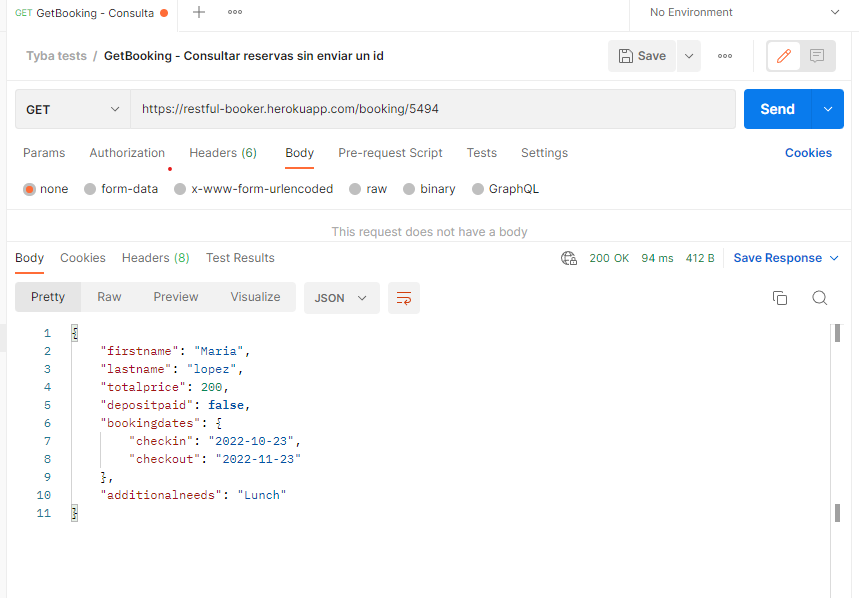
**Response** ****

1. Consultar reserva ingresando un id

Dado que el usuario creo previamente una reserva  
Y quiere consultarla

Cuando consume el servicio con el id específico

Entonces debe obtener la información de dicha reserva. **Éxito**

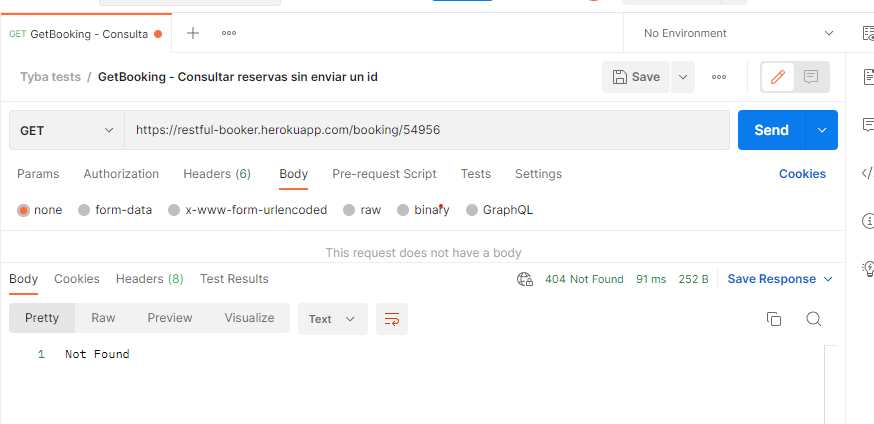
**Response** ****

1. Consultar reserva ingresando un id

Dado que el usuario creo previamente una reserva  
Y quiere consultarla

Cuando consume el servicio con el id específico

Entonces debe obtener la información de dicha reserva. **Éxito**

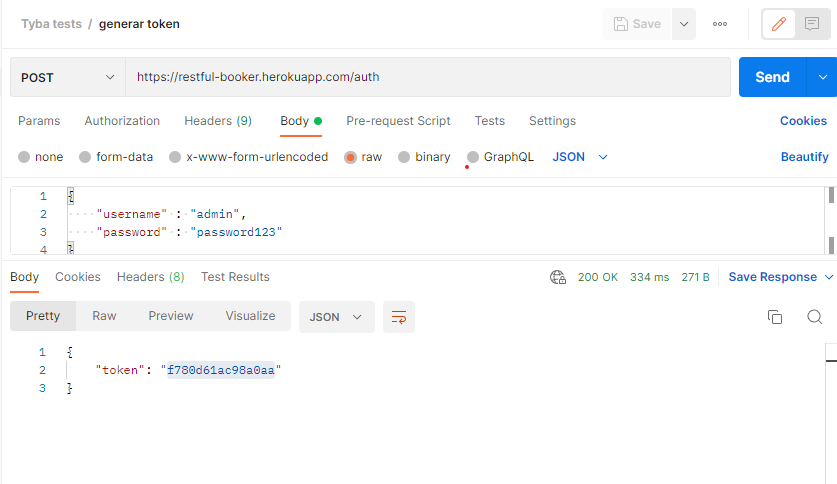
**Response** ****

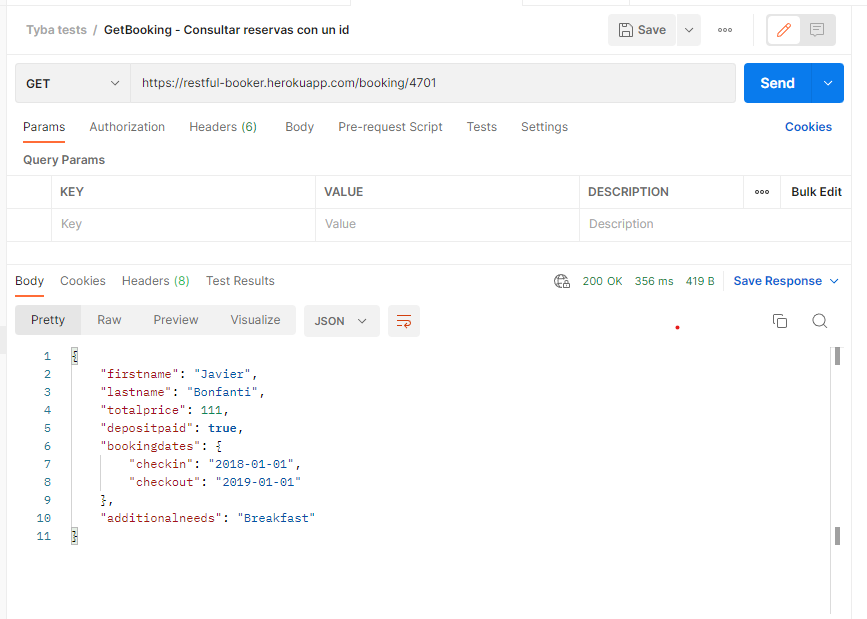
1. Actualizar reserva exitosamente

Dado que el usuario creo previamente una reserva  
Y quiere actualizarla

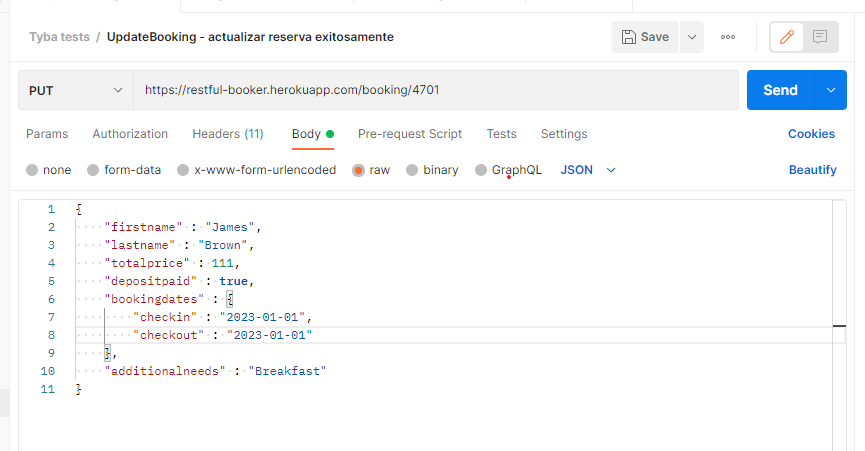
Cuando consume el servicio UpdateBooking

Entonces la reserva se debe actualizar exitosamente. **Éxito**

Generar token token 

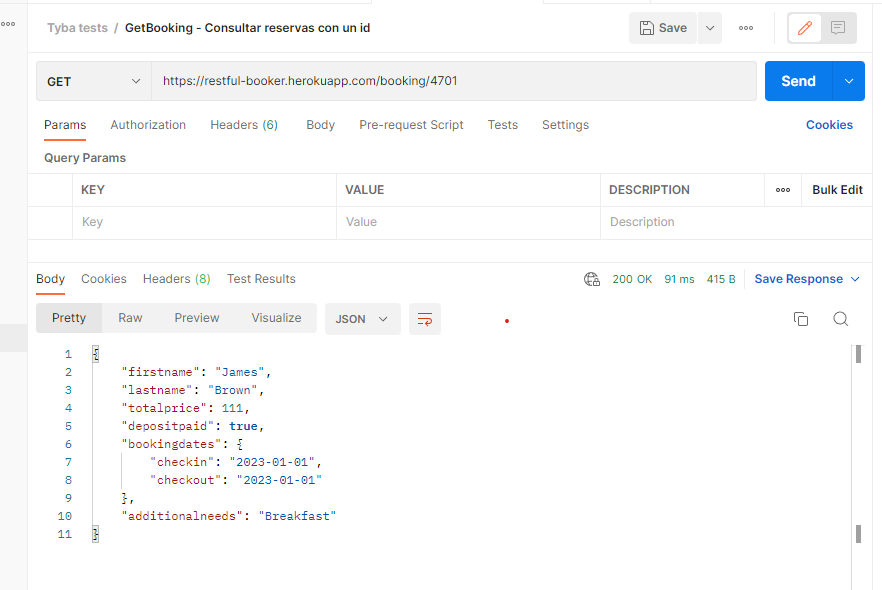
**Reserva original** ****

**Actualización de la reserva**

**Request** 

**Response** 

**Consultar reserva después de la modificación**

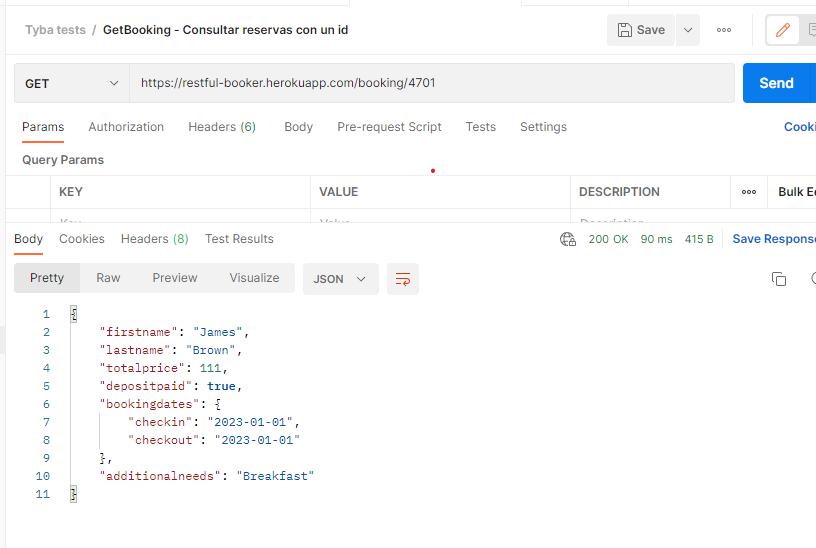
****

1. Intentar actualizar reserva con fecha pasada

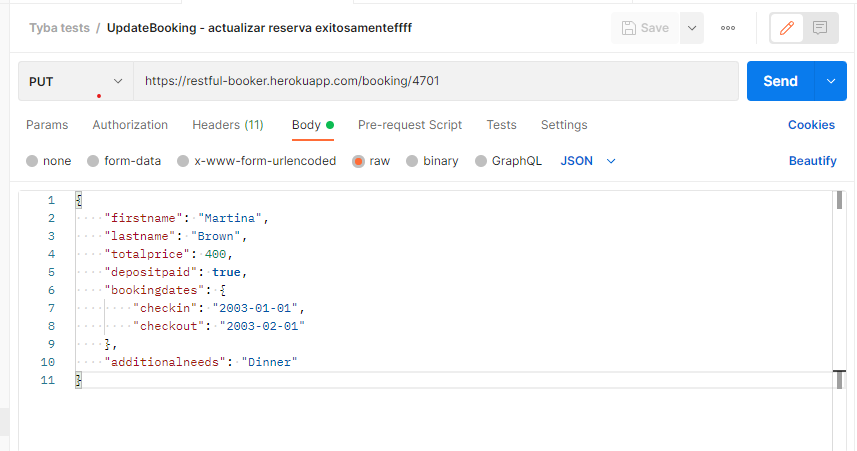
Dado que el usuario creo previamente una reserva  
Y quiere actualizarla

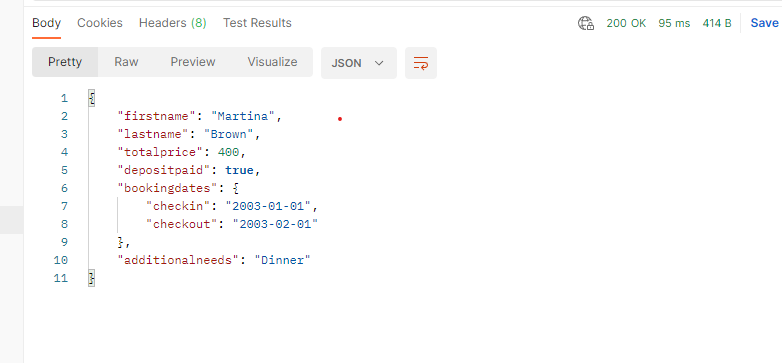
Cuando consume el servicio UpdateBooking con fecha de checkin y checkout pasadas

Entonces la reserva NO debe ser actualizada y debe retornar un código de error. **Fallido**

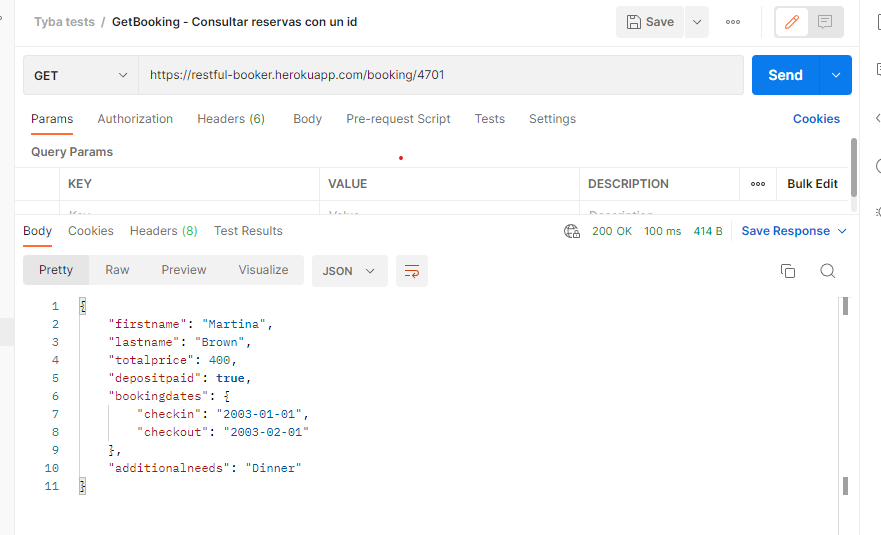
**Reserva** ****

**Actualización de la reserva**

**Request** 

**Response** 

**Consultar reserva después de la modificación**

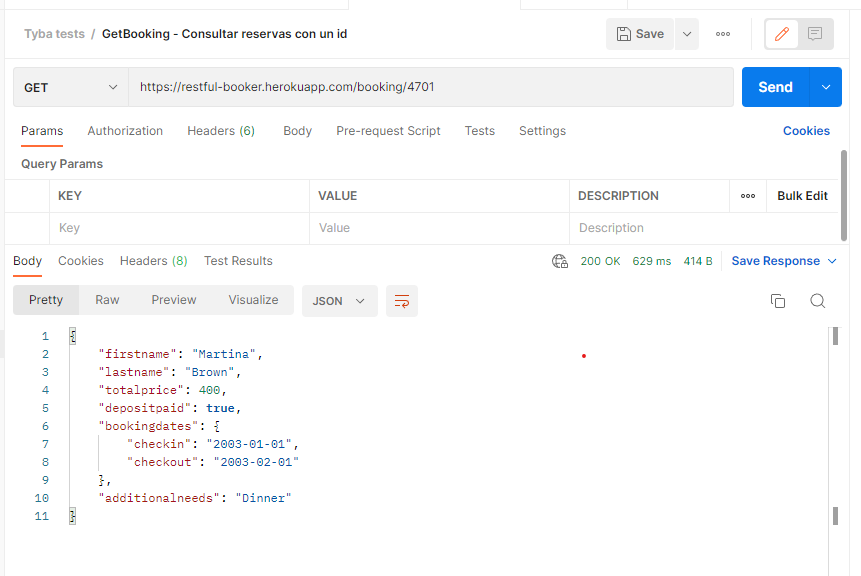


1. Intentar actualizar reserva con fecha de checkin invalida

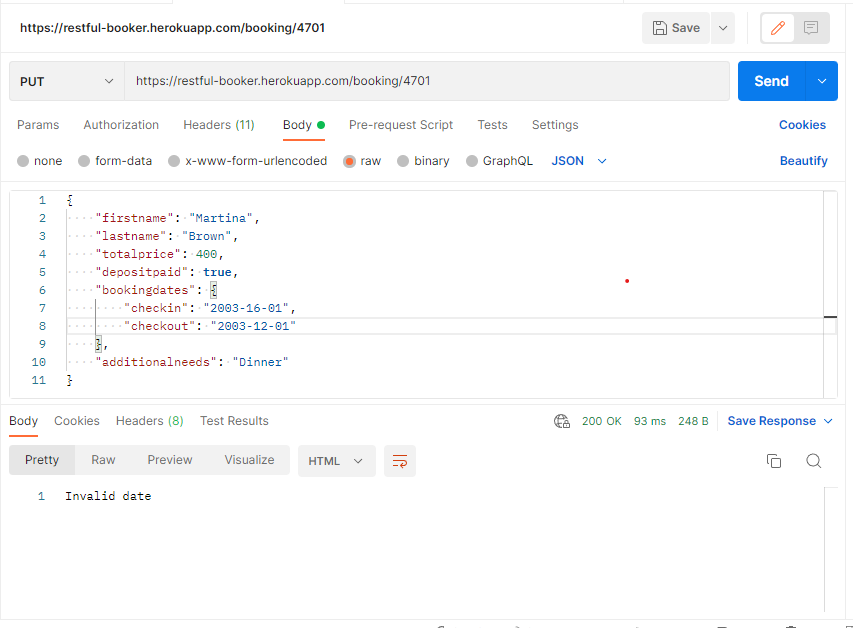
Dado que el usuario creo previamente una reserva  
Y quiere actualizarla

Cuando consume el servicio UpdateBooking con fecha de checkin inválida

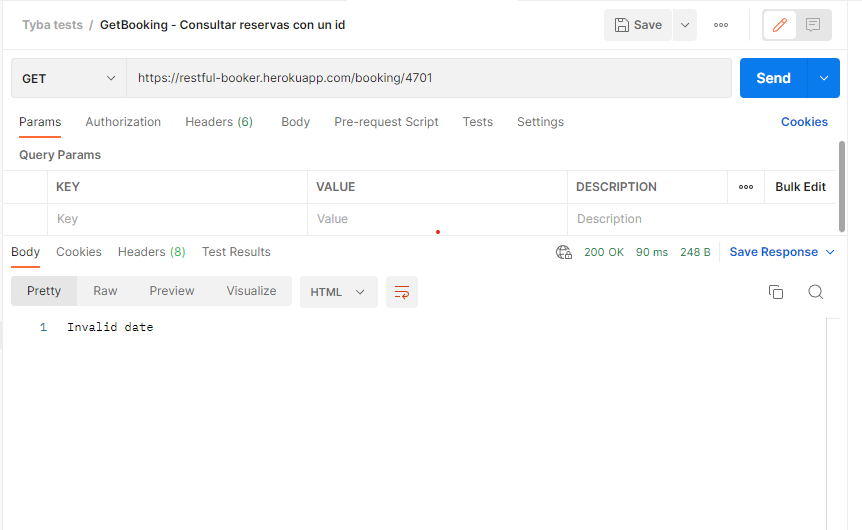
Entonces la reserva NO debe ser actualizada y debe retornar un código de error. **Fallido (Se actualizó la reserva por el mensaje Invalid date)**

**Reserva **

**Actualización de la reserva**

**Request** 

**Consultar reserva después de la modificación**

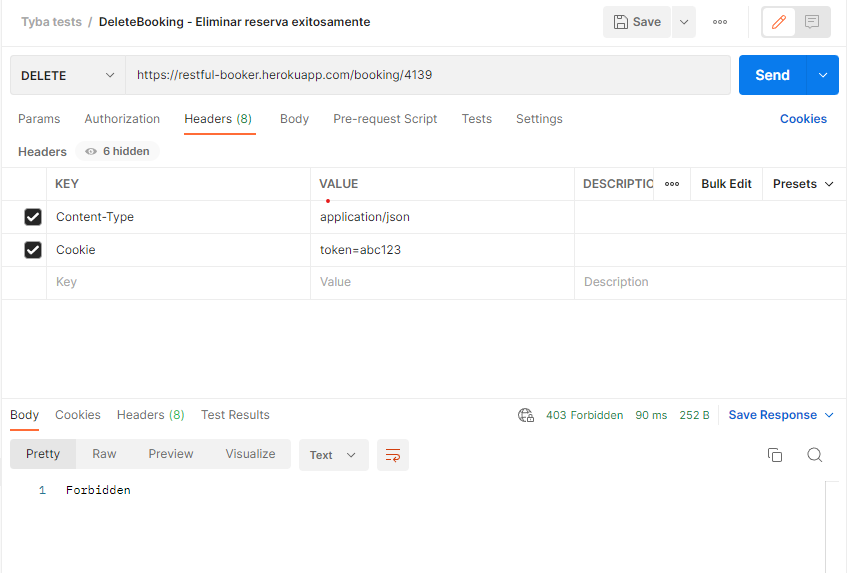


1. Intentar eliminar una reserva con un token vencido

Dado que el usuario creo previamente una reserva  
Y quiere eliminarla

Cuando consume el servicio DeleteBooking token inválida

Entonces la reserva NO debe poder eliminar. **Exito**

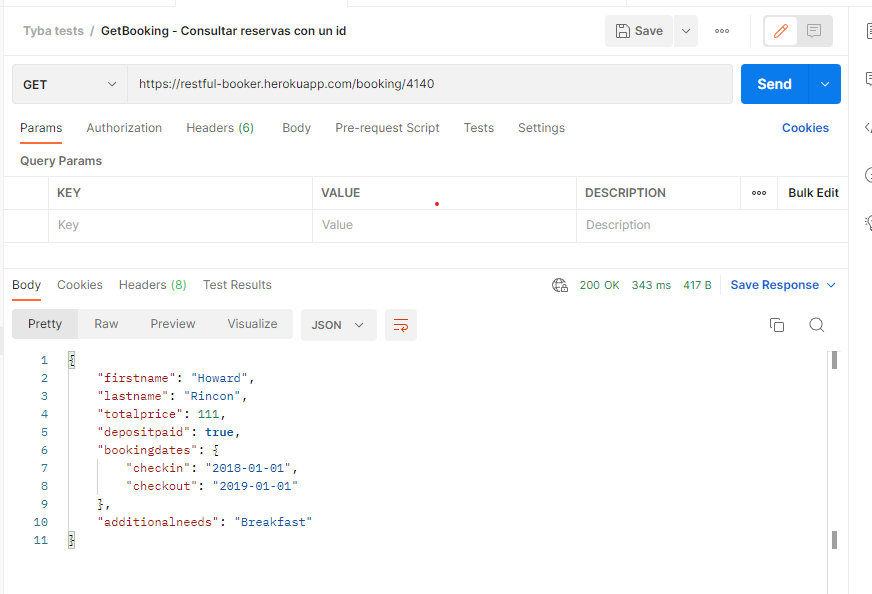


1. Eliminar una reserva exitosamente

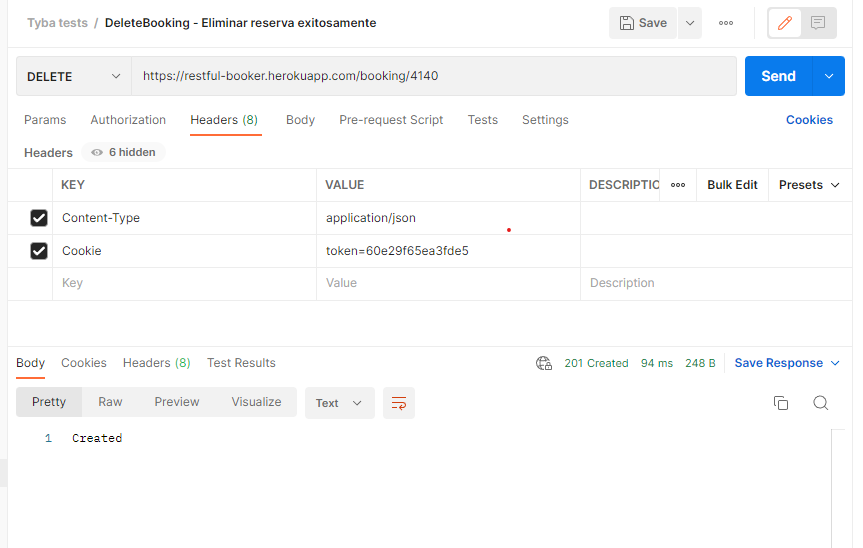
Dado que el usuario creo previamente una reserva  
Y quiere eliminarla

Cuando consume el servicio DeleteBooking

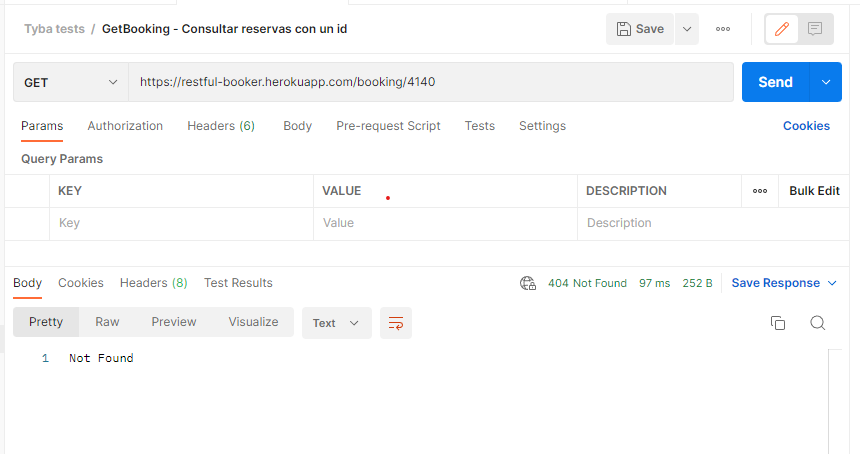
Entonces la reserva debe ser eliminada exitosamente. **Éxito (Pero el mensaje de respuesta debe ser diferente, en este momento es Created)**

**Reserva** ****

**Eliminar la reserva**

****

**Consultar la reserva después de ser eliminada**



1. Intentar eliminar reserva sin un id

Cuando se consume el servicio DeleteBooking sin un ID

Entonces se debe obtener un código de error y no se debe eliminar ninguna reserva 