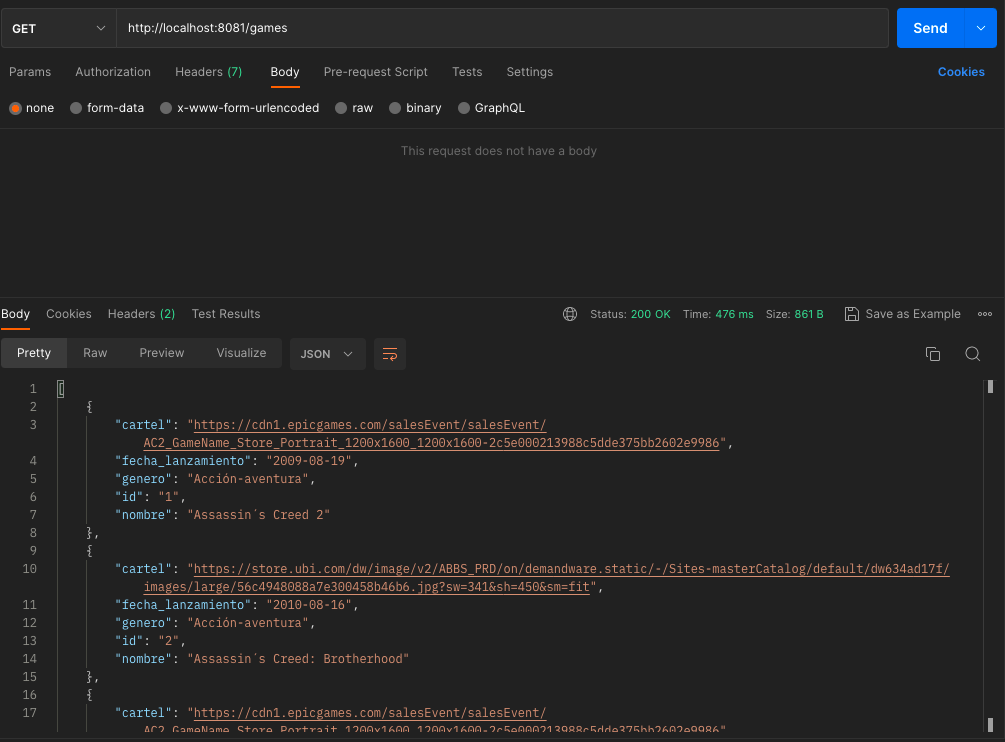
**MÓDULO VIDEOJUEGOS-QUARKUS:**

Este módulo es el encargado de obtener los datos de los videojuegos de la base de datos. Contiene los siguientes paquetes:

* Paquete daos: En este paquete se incluyen la interfaz con las funciones necesarias para hacer las peticiones a la base de datos de los videojuegos y su clase q implementa dichas funciones en un subpaquete impl (implementación)
* PostgresDataBaseDao: Contiene las funciones para seleccionar el listado (posteriormente con un GET), seleccionar un videojuego (posteriormente con otro GET), actualizar los valores de un videojuego (posteriormente con un PUT), insertar un videojuego (posteriormente con un POST) y eliminar un videojuego (posteriormente con un DELETE).
* Paquete repository: Aquí se meten la interfaz que incluye las funciones con las que ésta se comunica con el dao y su clase que las implementa en un subpaquete impl:
* repositoryImpl
* Paquete services: Lleva a cabo con una inerfaz y una clase que implementa sus funciones en un subpaquete impl, el mismo procedimiento que repository con dao pero, ahora con repository:
* serviceImpl
* Paquete resources: En este paquete se incluye la clase con los recursos http para hacer las peticiones. Se comunica con serviceImpl para obtener los datos del repository que solicita al dao.

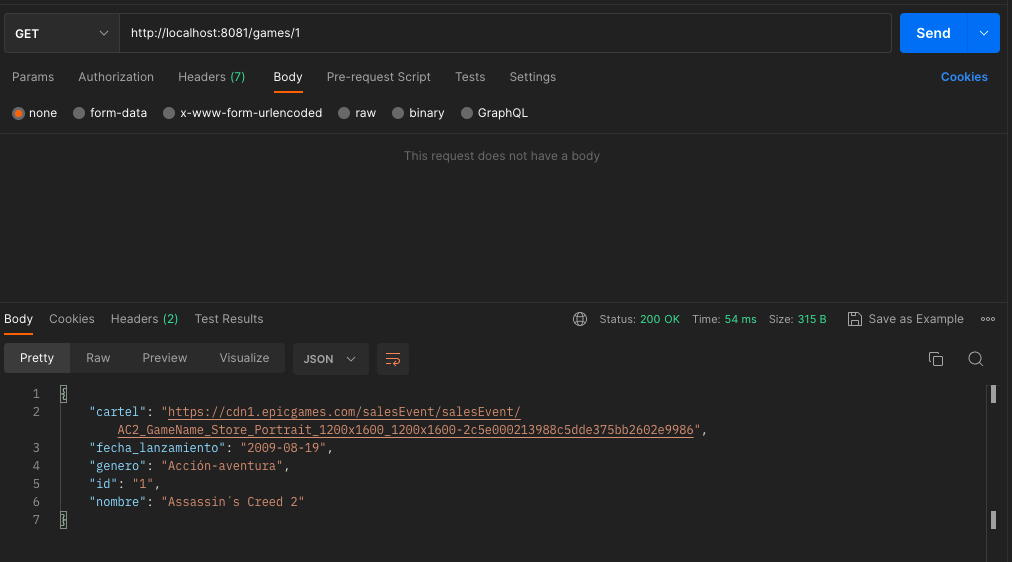
1. Función getVideojuegos():

* Método http: GET
* Resource path: “/games”
* Obtiene el listado de videojuegos
* Status code 200 OK
* Media Type “application/json”
* Ejemplo:



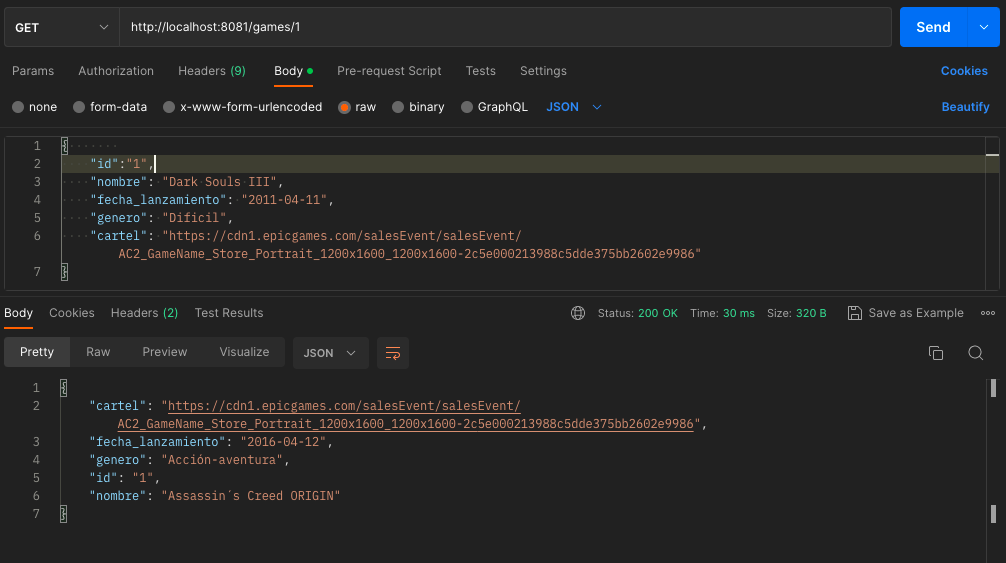
1. Función getVideojuegosById():

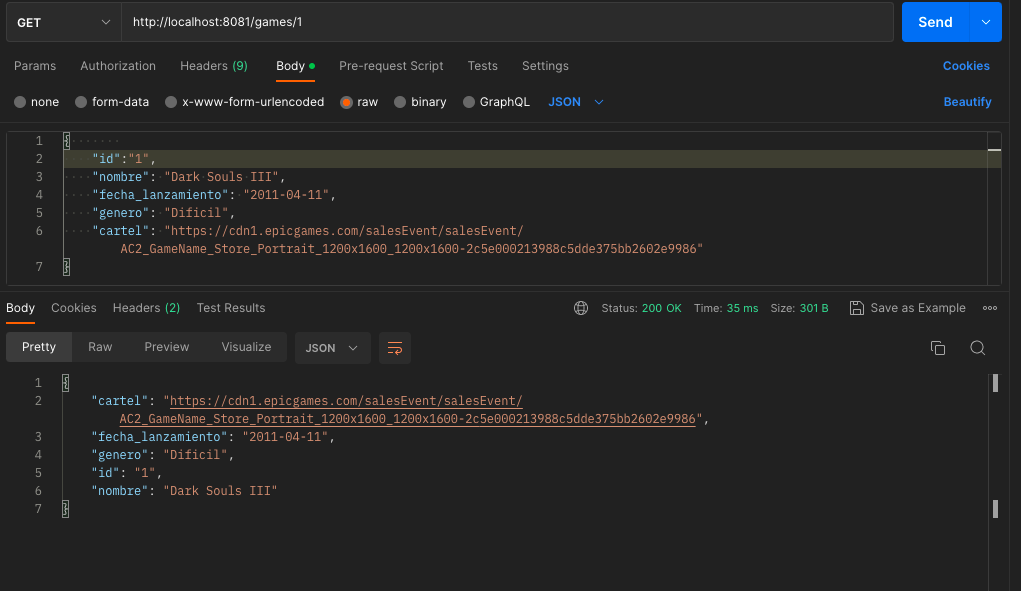
* Método http: GET
* Resource path: “/games/{id}”
* Obtiene del listado un videojuego concreto pasándole un id como parámetro.
* Incluye un pathparam id para identificar el videojuego.
* Status code 200 OK
* Media Type “application/json”
* Ejemplo:



1. Función updateVideojuegoById():

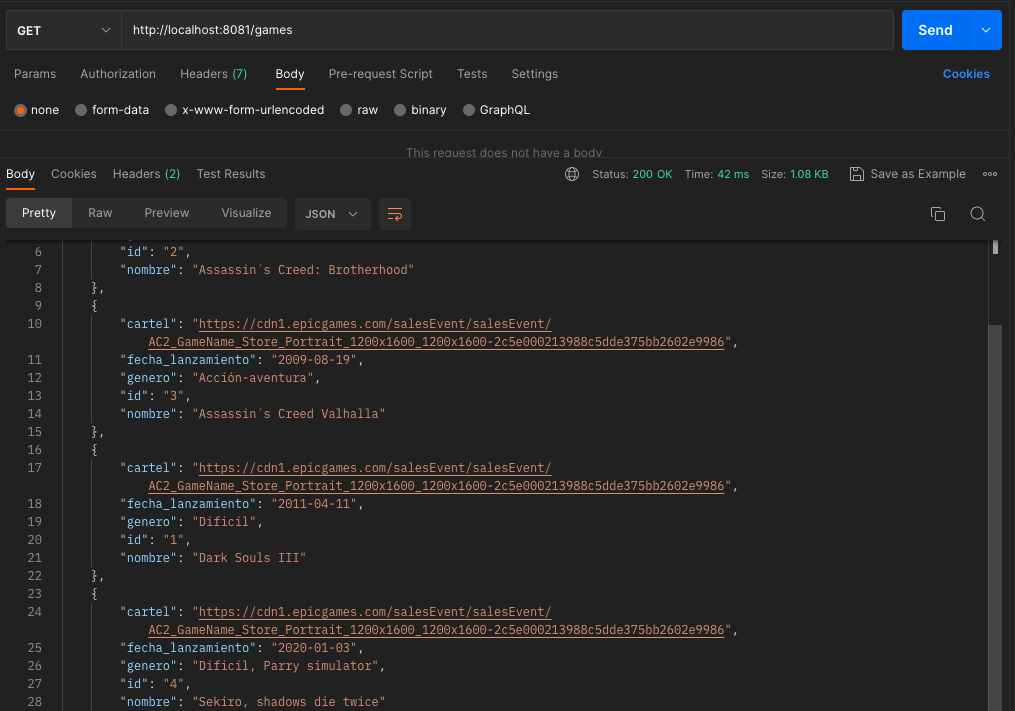
* Método http: PUT
* Resource path: “/games/{id}”
* Obtiene del listado de videojuegos una a través de su id y actualiza sus valores.
* Incluye un pathparam id para identificar el videojuego y un UnidentifiedVideojuego.
* Status code 204 No content
* Media Type “application/json”
* Ejemplo:



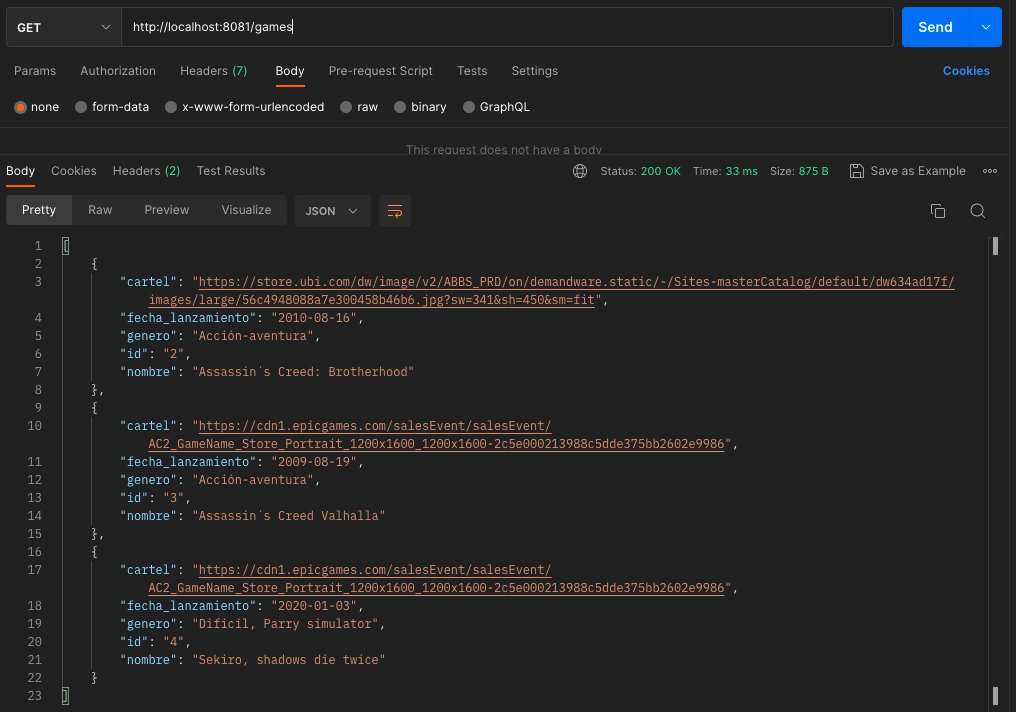


1. Función deleteVideojuegoById():

* Método http: PUT
* Resource path: “/games/{id}”
* Obtiene del listado de videojuegos a través de su id y lo elimina.
* Incluye un pathparam id para identificar el videojuego.
* Status code 204 No content
* Media Type “application/json”
* Ejemplo:

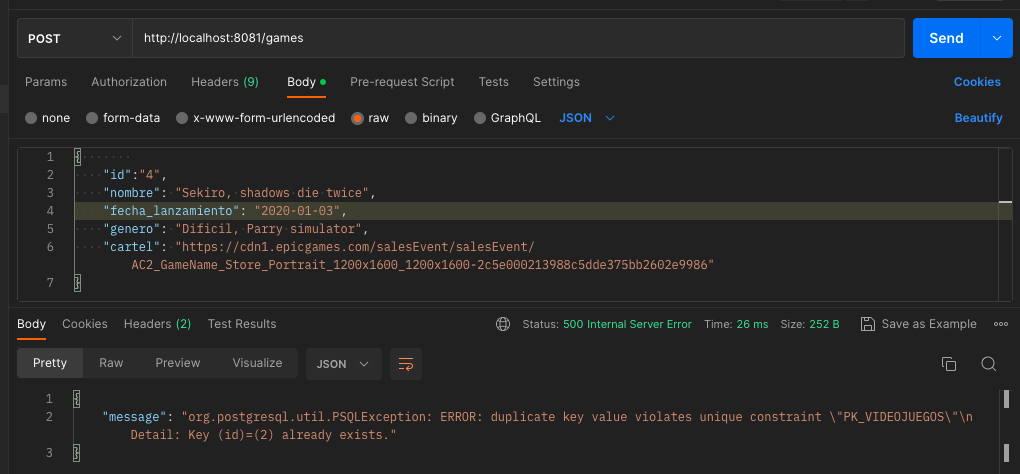






1. Función insertVideojuego():

* Método http: POST
* Resource path: “/games”
* Inserta un nuevo videojuego en el path indicado dentro de la función y la mete en la base de datos.
* Se le pasa como parámetro un UnidentifiedVideojuego.
* Status code 201 Created
* Media Type “application/json”
* Ejemplo:



Si la secuencia coincide con un id ya existente emite un error por la key.

