**MÓDULO PLATAFORMAS-QUARKUS:**

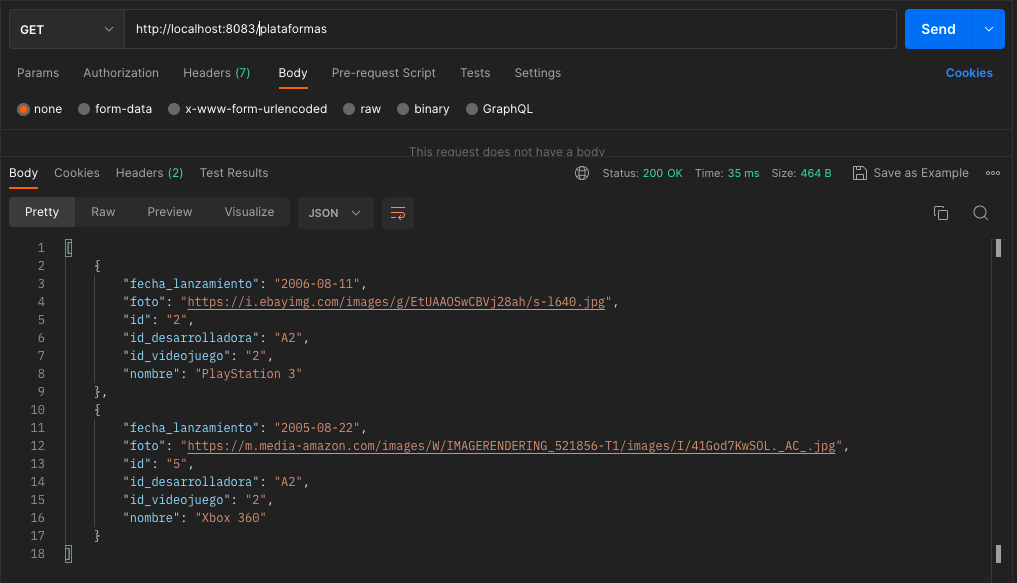
Este módulo es el encargado de obtener los datos de las plataformas de la base de datos. Contiene los siguientes paquetes:

* Paquete daos: En este paquete se incluyen la interfaz con las funciones necesarias para hacer las peticiones a la base de datos de las plataformas 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 las plataformas
* 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 getPlataformas():

Dentro de plataformaResource(/plataformas)

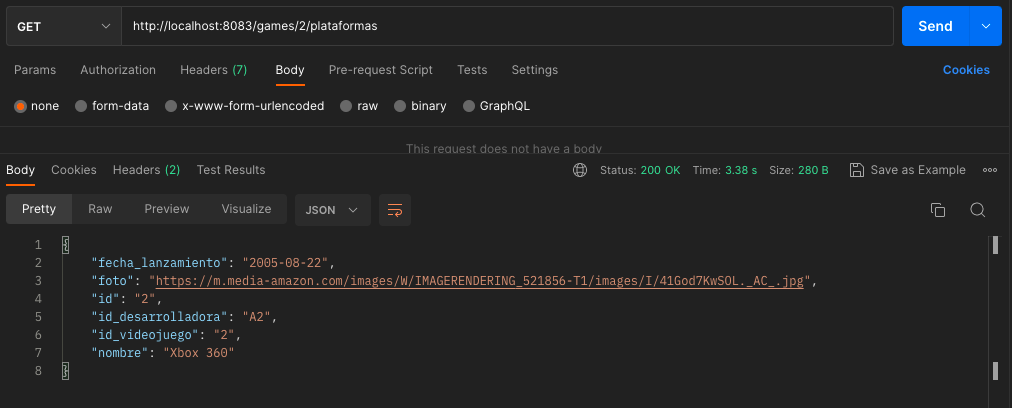
* Método http: GET
* Resource path: “/plataformas” (el propio de la clase, no de la funcion)
* Obtiene el listado de plataformas
* Status code 200 OK
* Media Type “application/json”
* Ejemplo:



1. Función getPlataformaByVideojuego():

Dentro de VideojuegosResource (/games)

* Método http: GET
* Resource path: “/{id}/plataformas” ( seleccionamos el videojuego por su id, y de ese la plataforma a la que pertenece)
* Incluye un pathparam id para identificar la plataforma.
* Status code 200 OK
* Media Type “application/json”
* Ejemplo:



Esta función solo realiza estas dos funciones, podríamos agregarle update, put y delete como a videojuegos y a desarrolladoras…  
  
Continuaremos con esto según necesidades para el trabajo 2.