RPG Data API

Use Case Description: Allows a user to retrieve a weapon stored in the database based on its unique identification. After locating the weapon, the API will return the weapon's attributes/statistics.

Use Case Name: Read Weapon

Actors:

- User

Triggers:

- The user wants to obtain information about a pre-existing weapon stored in the database of their role playing game application.
- The user wants to check to see if a weapon exists in the database

Preconditions:

- The user knows the UUID of the weapon being retrieved.

Post Conditions:

- A weapon whose UUID corresponds to the one provided will be returned..

Normal Flow:

1. The user makes an HTTP GET request using an available HTTP client such as curl or Postman, passing a URI with the UUID at the end of the path.
2. The RPG Data API receives the GET request sent by the user, and the WeaponsController object executes the "get" method. The UUID gets passed in as an argument.
3. The API then tries to make a new ResponseEntity whose body contains the attributes of the weapon corresponding to the UUID by calling the "findWeapon" method on the WeaponsService interface, which executes the method implemented on the WeaponsServiceImpl object.
4. The "findWeapon" method takes the UUID and calls "findByID" in the WeaponsRepository interface.
5. If the UUID matches an existing WeaponEntity in the database, WeaponsRepository will return the WeaponEntity in a JSON representation. If this process fails, an error will be thrown.
6. This WeaponEntity is returned to the WeaponsController object, and the WeaponsController object creates and returns a ResponseEntity object. Again, if this fails, an error will be thrown.
7. On success, the weapon's information is returned to the end user along with a HTTP response code of 200 Created.