RPG Data API

Use Case Description: Allows a user to create a weapon with attribute values that will be stored in the database for future use. This weapon's attributes represent its generic statistics that can be utilized in any role playing game application.

Use Case Name: Create Weapon

Actors:

- User

Triggers:

- The user wants to create a weapon for their role playing game application.

Preconditions:

- The user knows what attribute values should be included with the created role playing game weapon.

Post Conditions:

- A role playing game weapon is created in the database.

Normal Flow:

1. The user executes a HTTP POST call using an available HTTP client such as curl or Postman, and sends the attribute information for the weapon in a JSON request body.
2. The RPG Data API receives the POST request sent by the user, and the WeaponsController object executes the “post” method.
3. The JSON request body sent by the user is transformed into a Weapon object, and the object's contents are validated.
4. If the Weapon object has invalid data, an error response is returned to the user.
5. If the data is valid, the “createWeapon” method is called on the WeaponsService interface which executes the method implemented on the WeaponsServiceImpl object.
6. The WeaponsServiceImpl object creates a new WeaponEntity object from the data stored in the Weapon object, and persists the entity object by making a “save” method call on the WeaponsRepository interface.
7. The WeaponsRepository interface executes the “save” method via the Spring Data JPA framework, and returns the newly created weapon's UUID on success. On failure, an exception is thrown and returned to the user as an error response.
8. The new weapon's UUID is returned to the WeaponsController object, and the WeaponsController object creates and returns a ResponseEntity object storing the UUID in a Location header.
9. On success, a Location header with the new weapon's UUID is returned to the end user along with a HTTP response code of 201 Created.