

Design Pattern- Builder Pattern

The RPGD-API implements the Builder pattern, primarily as a part of its weapons creation process. The top layer of the execution comes from the WeaponsController class, which enacts a POST request as means of accomplishing the build. The user passes a Weapon object, which contains attributes for the new weapon, into the post() method in WeaponsController. The WeaponsController contains a WeaponsService object, and the Weapon object is passed down to it (under the assumption that it is a valid Weapon). The WeaponsService interface is implemented by the WeaponsServiceImpl class, and upon receiving the Weapon, it passes the Weapon onto the WeaponEntity class, which contains the nested WeaponBuilder class. WeaponBuilder has a set method for each of the Weapon's attributes, and a build() method which returns the complete WeaponEntity object once each of the properties have been assigned. The WeaponEntity is passed back up to WeaponsServiceImpl, which saves it into the Weapon Repository. Then, if the process threw no exceptions, the WeaponEntity is passed back up again to the WeaponsController. WeaponsController will create a URI for the WeaponEntity and will finish the post request by producing an affirmative ResponseEntity.

To summarize, the creation of the weapon occurs in the WeaponBuilder class, which is then handed over to the implementation of the WeaponsService class. The weapon is then saved in the repository, and handed up once more to the WeaponsController, which creates a URI for the weapon. This completes the build process.