```
/register
(POST)
                    JWT
/login (POST)
```

- User Service ↔ User Repo ↔ User DB
- /register (POST) → User Service
- /login (POST) ← JWT
- Validate JWT (User Service ↔ Toy Service)
- CRUD Ops ↔ Toy Service ↔ Toy Repo ↔ Toy DB
- Toy Service → /inventory (POST)
- On Create Toy, Set Initial Inventory of 50 units
- Reduces Inventory — /inventory/{id} (PUT) ↔ Inventory Service ↔ Inventory Repo ↔ Inventory DB
- Retrieves Current Inventory — /inventory/{productNumber} (GET) → Inventory Service

High-Level Requirements:

1. Create a separate inventory microservice that supports a set initial inventory operation, an update inventory operation, and a get inventry operation
2. When executing the create operation for a toy, call out to the inventory service to initialize quantity for that toy to 50
3. Use a separate database for inventory information
4. Inventory data should include:
   - ID (internal, autoincrementing ID)
   - Product number (string)
   - Quantity (whole number)
5. If you want to reset your database container, remove the container from Docker and run `sudo -rf rm ~/mysql` to remove the MySQL state stored locally