

Product Data Broker

1 Project Structure

```
.
docs
  dependancies.drawio
  dependancies.drawio.pdf
  object_graph.drawio
  object_graph.drawio.pdf
README.md
src
  data
    Mongo
      generated_products.json
      init.js
    MySQL
      generated_products.csv
      init.sql
  docker-compose.yml
  pdb
    go.mod
    go.sum
    pdb.go
```

2 Description

This project consists of two databases (MySQL and MongoDB) managed via Docker, with a Go application (pdb.go) that connects to both and executes queries.

3 Prerequisites

- Docker & Docker Compose
- Golang (version 1.21 or above)

4 Setup Instructions

1. Start Docker Containers:

```
docker compose up -d
```

2. Check containers are running:

```
docker ps
```

5 Database Initialization

5.1 MySQL

- Database: `products`
- Schema and initial data are set up with `init.sql` and `generated_products.csv`.

5.2 MongoDB

- Database: `products_datas`
- Collections initialized with `init.js` and `generated_products.json`.

6 Usage

1. Run Go Application:

```
cd src/pdb
go run pdb.go
```

- ### 2. Follow the prompts to query products. Examples of product codes are available in the file `src/pdb/products_codes.txt`

7 Diagrams

Diagrams for dependencies and object structure are available in the `docs` folder:

- `dependancies.drawio.pdf`
- `object_graph.drawio.pdf`

8 Cleanup

To stop and remove containers:

```
docker compose down
```