(Demo prep)

Using the MongoDB docker image

1. To start using docker in my box I'll start by opening a terminal

2. Collapse navigation pane

3. Start MongoDb container:

   ```
   docker run -d --rm --name mongo -p 27017:27017 -v mongodbdata:/data/db mongo
   ```

4. Run the service and try a POST:

   ```
   {
    "name": "Potion",
    "description": "Restores a small a mount of HP",
    "price": 5
   }
   ```

5. Notice the **location header**

6. Install the MongoDB extension and show the database

7. To fix the json representation of our items we just need to register a couple of MongoDB serializers. We can do this in the ConfigureServices method of the Startup class. We will talk more about the role of Startup in ASP.NET Core apps in the next lesson, so for now let's just think of ConfigureServices as the place where we register services that can be used across the entire application.

8. Register the MongoDB serializers in Startup.ConfigureServices:

   ```
   BsonSerializer.RegisterSerializer(new GuidSerializer(BsonType.String));
   BsonSerializer.RegisterSerializer(new DateTimeOffsetSerializer(BsonType.String));
   ```

9. Delete the Items DB

10. Post the potion again:

    ```
    {
     "name": "Bronze sword",
    ```

```
  "description": "Deals a small amount of damage",
  "price": 20
}
```

11. Show the new item in the MongoDB extension

12. Try all operations

In the next lesson we will learn about the dependency injection technique and the .NET configuration system.