Customer: Rentschler & Holder Rotebühlplatz 41 70178 Stuttgart

Supplier: Team 2 (Paul Brenner, Jonas Alexander Graubner, Mohaddeseh Tibashi, Selvana Dwi Ayunda, Luka Dominik Pavic) Rotebühlplatz 41 70178 Stuttgart

Version History

Version	Date	Author	Comment
0.1	28.04.2023	Jonas Graubner	Created 1-6
1.0	12.05.2023	Jonas Graubner	Finalization

Table of content

1. Scope

The MongoDB Database is used to persistently store all Asset data. This has the advantage of persistent Storge of all Asset and faster Asset search times.

2. Glossary

GUI - graphical user interface AAS - asset administration shell AASX - file format to store an asset AASX server - server, that can store AAS assets and has a standardized API specified in the GitHub repository

3. Module Requirements

3.1. User View

Since MongoDB is an Database the user doesn't have a view of this Module. Nonetheless can a developer use the programm MongoDBCompass to get an easy overview of the Data currently stored in the MongoDB

3.2. Requirements

This Module is the MongoDB integration thus it has the function of reliable Data Storage.

- AASM-REQ7 MongoDB Integration
- AASM-NF20 Reliability

3.3. Module Context

This Module is used in the Module MOD02. The MongoDB is used within the AASX Server with the help of the C# MongoDB Driver. The Job of the MongoDB Driver is not only to expose an interface within C# for the MongoDB but also to Convert C# objects into an BSON File which can be stored in the MongoDB.

4. Analysis

This Module needs to work so that the AASX Server isn't dependant on in Memory storage of the Assets. If this Module fails the AASX Server will become unresponsive and the Server Rest-API won't work anymore.

For easier use within the AASX Server a seperate high Level class MongoDBInterface was created for the mongoDB which exposes the following functions:

- function to initialize the Database Connection
- functions to read from the DB
- functions to update in the DB
- function to delete in the DB
- functions to save delete from the DB
- function to write to DB
- function to import AAS Environment into DB

5. Design

For more effizient use the MongoDB is divided in the Collections ConceptDescription, dataSpecifications, Filenames, Shells and Submodels. For each collection exists an own read, update and save delete function. The functions to initialize the Database, delete in Database, write to Database and import AAS Environment into Database can be used for all different collections.

6. Implementation

The MongoDB .NET/c# Driver is used to creacte an Connection in C# to the MongoDB. The Driver also handles the conversion from C# Object to BSON/JSON documents and vice versa.

Link to Code: https://github.com/JoTec2002/TINF21C_AAS_Management/blob/main/SOURCE/Server/aasx-server/src/AasxServerStandardBib/MongoDBInterface.cs

7. Module Tests