**Nimbus** is a toolkit that, once installed on a [cluster](https://en.wikipedia.org/wiki/Cluster_(computing)), provides an [infrastructure as a service](https://en.wikipedia.org/wiki/Infrastructure_as_a_service) [cloud](https://en.wikipedia.org/wiki/Cloud_computing) to its client via [(web service resource framework WSRF)](https://en.wikipedia.org/wiki/Web_Services_Resource_Framework)-based or [Amazon EC2](https://en.wikipedia.org/wiki/Amazon_Elastic_Compute_Cloud) [WSDL](https://en.wikipedia.org/wiki/Web_Services_Description_Language) [web service](https://en.wikipedia.org/wiki/Web_service) [APIs](https://en.wikipedia.org/wiki/Application_programming_interface). Nimbus is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software), subject to the requirements of the [Apache License](https://en.wikipedia.org/wiki/Apache_License), version 2.

Multiple technologies were used in development and building this project.

I would like to discuss about the mongodb in particular as it forms a core technology to handle data. There are multiple locations where mongo db manipulations were found but I picked up two of the files DbSequence and MongoIdSequenceRepository from the sub directories of path given below.

[**nimbus-core**](https://github.com/openanthem/nimbus-core)/[nimbus-core](https://github.com/openanthem/nimbus-core/tree/develop/nimbus-core)/[src](https://github.com/openanthem/nimbus-core/tree/develop/nimbus-core/src)/[main](https://github.com/openanthem/nimbus-core/tree/develop/nimbus-core/src/main)/[java](https://github.com/openanthem/nimbus-core/tree/develop/nimbus-core/src/main/java)/[com](https://github.com/openanthem/nimbus-core/tree/develop/nimbus-core/src/main/java/com)/[antheminc](https://github.com/openanthem/nimbus-core/tree/develop/nimbus-core/src/main/java/com/antheminc)/[oss](https://github.com/openanthem/nimbus-core/tree/develop/nimbus-core/src/main/java/com/antheminc/oss)/[nimbus](https://github.com/openanthem/nimbus-core/tree/develop/nimbus-core/src/main/java/com/antheminc/oss/nimbus)/[domain](https://github.com/openanthem/nimbus-core/tree/develop/nimbus-core/src/main/java/com/antheminc/oss/nimbus/domain)/[model](https://github.com/openanthem/nimbus-core/tree/develop/nimbus-core/src/main/java/com/antheminc/oss/nimbus/domain/model)/[state](https://github.com/openanthem/nimbus-core/tree/develop/nimbus-core/src/main/java/com/antheminc/oss/nimbus/domain/model/state)/[repo](https://github.com/openanthem/nimbus-core/tree/develop/nimbus-core/src/main/java/com/antheminc/oss/nimbus/domain/model/state/repo)

Apart from the above locations there are other locations where mongodb files can be found. The other locations are as follows:

n[imbus-core/src/main/java/com/antheminc/oss/nimbus/app/extension/config/DefaultMongoConfig.java](https://github.com/openanthem/nimbus-core/blob/4dbcc561544b1e7b3152fc23a880bfc20ac9094b/nimbus-core/src/main/java/com/antheminc/oss/nimbus/app/extension/config/DefaultMongoConfig.java" \o "nimbus-core/src/main/java/com/antheminc/oss/nimbus/app/extension/config/DefaultMongoConfig.java)

[nimbus-core/src/main/java/com/antheminc/oss/nimbus/domain/cmd/exec/internal/process/ParamUpdateEventListener.java](https://github.com/openanthem/nimbus-core/blob/308c8dee165eb629cb2740bb2e477839533ed240/nimbus-core/src/main/java/com/antheminc/oss/nimbus/domain/cmd/exec/internal/process/ParamUpdateEventListener.java)

DbSequence.java contains an annotation @Repo stating the database as repository with repository name as **Database.rep\_mongodb.**

MongoIdSequenceRepository is used to generate a sequence Id from mongodb. The class holds a constructor which carries MongoOperations instance. To generate a sequence id a String key or Id from mongodb is passed on to a method **getNextSequenceId**. This method contains query object, update object and FindAndModifyOptions object to get the sequence, update the sequence. Sequence id is generated finally passing the said objects as parameters to the mongooperations method findAndModify.

DBSequence seqId = mongoOperations.findAndModify(query, update, options, DBSequence.class, "sequence");

Finally sequence is returned. If a sequence is not returned it throws exception unable to get sequenceid for the key.