



«interface»

IDataAccess

«Property»

- PoolAccess :IPoolAccess

+ CreateDataEntry(string, string, double, double, double, double) :bool

+ DeleteAllData() :void

+ GetChlorineValues(string, string, int) :List<Tuple<SensorTypes, double>>

+ GetTemperatureValues(string, string, int) :List<Tuple<SensorTypes, double>>

+ GetPhValues(string, string, int) :List<Tuple<SensorTypes, double>>

+ GetHumidityValues(string, string, int) :List<Tuple<SensorTypes, double>>

Exception

UserNotFoundException

Exception

PoolNotFoundException

Exception

MultipleOccourencesOfEmailWasFoundException

Humidity

«Property»

+ Id :int

+ Value :double

+ DataId :int

«property»

+ Data() :Data

«interface»

ISmartpoolDB

«Property»

- PoolAccess :IPoolAccess

- UserAccess :IUserAccess

- DataAccess :IDataAccess

+ ClearEntireDatabase() :void

DataAccess

«Property»

+ PoolAccess :IPoolAccess

+ DataAccess(IPoolAccess)

+ CreateDataEntry(string, string, double, double, double, double) :bool

+ DeleteAllData() :void

+ GetChlorineValues(string, string, int) :List<Tuple<SensorTypes, double>>

+ GetTemperatureValues(string, string, int) :List<Tuple<SensorTypes, double>>

+ GetPhValues(string, string, int) :List<Tuple<SensorTypes, double>>

+ GetHumidityValues(string, string, int) :List<Tuple<SensorTypes, double>>

pH

«Property»

+ Id :int

+ Value :double

+ DataId :int

«property»

+ Data() :Data

User

«Property»

+ Id :int

+ Firstname :string

+ Middelname :string

+ Lastname :string

+ Email :string

+ Password :string

+ User()

«property»

+ Pool() :ICollection<Pool>

SmartpoolDB

«Property»

+ PoolAccess :IPoolAccess

+ UserAccess :IUserAccess

+ DataAccess :IDataAccess

+ SmartpoolDB(IDataAccess)

+ ClearEntireDatabase() :void

DbContext

DatabaseContext

+ DatabaseContext()

# OnModelCreating(DbModelBuilder) :void

«property»

+ UserSet() :DbSet<User>

+ PoolSet() :DbSet<Pool>

+ pHSet() :DbSet<pH>

+ ChlorineSet() :DbSet<Chlorine>

+ TemperatureSet() :DbSet<Temperature>

+ HumiditySet() :DbSet<Humidity>

+ DataSet() :DbSet<Data>

Temperature

«Property»

+ Id :int

+ Value :double

+ DataId :int

«property»

+ Data() :Data

Pool

«Property»

+ Id :int

+ Name :string

+ Volume :double

+ UserId :int

+ Pool()

«property»

+ User() :User

+ Data() :ICollection<Data>