```
AtomicDB Add-On in Wolfram Language
AtomicDB API Primitive Operations Package
DEMO VERSION
(c) April 2015, By Athanassios I. Hatzis, PhD
All Rights Reserved

Package for Microsoft Windows (64-bit), x86-64, Windows
 Mathematica ver.10.0 for Microsoft Windows (64-bit) (June 29, 2014)
.NET Framework ver. : 4.0.30319.34209
```

Notice : This version of AtomicDB AddOn is for demonstration purposes only, not for commercial or other business use !

AtomicDB Add - On in *Mathematica*

AtomicDB API Primitive Operations Package Test (Demo Version)

By Athanassios I. Hatzis - (C) Thu 9 Apr 2015 × 01 : 23 : 50

This output has been generated automagically. ☺

```
In this demo we build first  a simple relational data model using the
  Wolfram List structure.Our relational model example includes two
  main tables STOCK and ORDER that are joined with a third junction
  table STOCK-ORDER. Then we convert this to AtomicDB data model
  by adding a new Model, then Concepts (columns) and Records (rows).
```

■ Relational Model

Relations

STOCK Table

StockID	StockNameEN	StockPrice	StockNameGR
991	Pinto Beans	11.1	Φασόλια Πίντο
992	Kidney Beans	9.85	Φασόλια Κόκκινα
993	White Beans	13.45	Φασόλια Άσπρα
994	Wax Beans	18.72	Φασόλια Καναρίνια

ORDER Table

OrderID	OrderKey
441	1111-BZ
442	1117-CM
443	1118-SA
444	1119-TT

STOCK - ORDER Table

```
SOID     SOOrderID     SOStockID     SOQuantity
224      441           991           1
225      442           992           3
226      443           994           2
227      444           993           1
228      441           993           3
```

AtomicDB Model

Login To Server

Existing Models

```
{}
```

Concept Map System

Add A New Model

```
{0, 3, 13, 256} → Beans Stock-Order Model Example
```

Get All Models

```
« NETObject[System.Collections.Generic.List`1[System.Collections.
     Generic.List`1[IAMCore_SharpClient.Core_KeyValuePair]]] »

{« NETObject[
   System.Collections.Generic.List`1[IAMCore_SharpClient.Core_KeyValuePair]] »}

« NETObject[IAMCore_SharpClient.Core_KeyValuePair] »
```

Print Key - Value Pair of the first model

```
{0, 3, 13, 256} → Beans Stock-Order Model Example
```

Add Concepts to the Model

Add STOCK Group Concepts

```
{2, 1025, 256, 1} → StockNEXUS
{2, 1025, 256, 2} → StockID
{2, 1025, 256, 3} → StockNameEN
{2, 1025, 256, 4} → StockPrice
{2, 1025, 256, 5} → StockNameGR
```

Add ORDER Group Concepts

```
{2, 1025, 256, 6} → OrderNEXUS
{2, 1025, 256, 7} → OrderID
{2, 1025, 256, 8} → OrderKey
```

Add STOCK - ORDER Group Concepts

```
{2, 1025, 256, 9} → SONEXUS
{2, 1025, 256, 10} → SOID
{2, 1025, 256, 7} → OrderID
{2, 1025, 256, 2} → StockID
{2, 1025, 256, 11} → SOQuantity
```

Data Holder System

Add Collections

Add STOCK Group Collections

```
{0, 3, 15, 257} → StockNEXUS
{0, 3, 15, 258} → StockID
{0, 3, 15, 259} → StockNameEN
{0, 3, 15, 260} → StockPrice
{0, 3, 15, 261} → StockNameGR
```

Add ORDER Group Collections

```
{0, 3, 15, 263} → OrderNEXUS
{0, 3, 15, 264} → OrderID
{0, 3, 15, 265} → OrderKey
```

Add STOCK - ORDER Group Collections

```
{0, 3, 15, 266} → SONEXUS
{0, 3, 15, 267} → SOID
{0, 3, 15, 264} → OrderID
{0, 3, 15, 258} → StockID
{0, 3, 15, 268} → SOQuantity
```

Add Records

Add STOCK Group Records

```
{2, 7, 257, 1} → 1
{2, 7, 257, 2} → 2
{2, 7, 257, 3} → 3
{2, 7, 257, 4} → 4
```

Add ORDER Group Records

```
{2, 8, 263, 1} → 1
{2, 8, 263, 2} → 2
{2, 8, 263, 3} → 3
{2, 8, 263, 4} → 4
```

Add STOCK - ORDER Group Records

```
{2, 8, 263, 1} → 1
{2, 8, 263, 2} → 2
{2, 8, 263, 3} → 3
{2, 8, 263, 4} → 4
```