# GameModeler SDK

## SaveLoadObject module documentation

## Introduction

With this module, you can save an object in differents languages on file. And you can load an object from file.

Differents languages/method :

- Json
- Yaml
- Xml
- Binary
- Csv
- Character

## Configuration

### Differents languages

- WRITE
  - Communal parameter list : path , fileName, T ObjectToWrite, append (optional)
  - Example (Json)

```
Class1 class1 = new Class1();
class1.Id = 1;
class1.Name = "name";
class1.Status = true;

JsonManager.WriteToJsonFile<Class1>(@"A:\", "test.json", class1);
```

- READ
  - Communal parameter list : path, fileName
  - Example (Json)

```
Class1 objJson = JsonManager.ReadFromJsonFile<Class1>(@"A:\", "test.json");
```

# MySQL

The MySQL submodules contains six methods :

- 3 methods for write/execute script from object
  - ConvertObjectInScript
    - Parameters :
      - T objectToWrite
      - boolean append
      - string dbName
      - bool execute
      - string user (mysql user)
      - string pwd (mysql password)
    - Return : SQL script (string) for creating database and table and execute or not
    - Example :

```
Class1 class1 = new Class1();
class1.Id = 1;
class1.Name = "name";
class1.Status = true;

string script = SqlManager.ConvertObjectInScript<Class1>(class1, false, "Test", true, true);
```

  - ExecuteStringSql
    - Parameters :
      - string script
      - string user (mysql user)
      - string pwd (mysql password)
    - Return : Execute SQL script (string)
  - WriteToSqlFile
    - Parameters :
      - string path
      - string fileName
      - T objectsToWrite
      - boolean append
      - string dbName
      - boolean createDb
      - boolean execute
      - string user (mysql user)
      - string pwd (mysql password)
    - Return : Write SQL script into file and execute or not

- 3 methods for read database and convert into object (IDictionnary)
  - CreateObjectByDatabase
    - Parameters :
      - string user (mysql user)
      - string pwd (mysql password)
      - string database
      - string table
    - Return : IDictionnary object from table of database
    - Example :

```
dynamic dynamicObject = new ExpandoObject();
var dic = dynamicObject as IDictionary<string, object>;
dic = DynamicManager.CreateObjectByDatabase("root", "", "test3", "class1");
```

  - CreateMySqlCommandArray
    - Parameters :
      - string query
      - MysqlConnection connection
      - Boolean openclose (for open and close connection in function)
    - Return mysql values (array) of query
  - CreateMySqlCommandDict
    - Parameters :
      - string query
      - MysqlConnection connection
      - Boolean openclose (for open and close connection in function)
    - Return mysql values (dictionnary) of query